

IBM Cognos Software Development Kit
Version 11.1.0

Developer Guide



Note

Before using this information and the product it supports, read the information in [“Notices” on page 1923.](#)

Product Information

This document applies to IBM Cognos Software Development Kit Version 11.1.0 and may also apply to subsequent releases.

Licensed Materials - Property of IBM

© **Copyright International Business Machines Corporation 2005, 2021.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

- Chapter 1. Introduction..... 1**
- Chapter 2. Understanding the IBM Cognos Software Development Kit.....3**
 - IBM Cognos Architecture..... 3
 - Service-Oriented Architecture..... 4
 - Access and Authentication.....5
 - Using the BI Bus API..... 5
 - IBM Cognos Analytics Services..... 5
 - IBM Cognos Software Development Kit Components..... 11
 - Cognos Software Development Kit..... 11
 - IBM Cognos Framework Manager Development.....12
 - IBM Cognos Mashup Service..... 12
 - Custom Authentication Provider API.....13
 - Script Player..... 13
 - Other Customization Capabilities..... 13
 - Extending IBM Cognos Into Your Business.....13
 - Integration, Customization, and Extensibility..... 13
 - Developer Resources.....15
 - Product Documentation..... 15
- Chapter 3. Getting started..... 17**
 - BI Bus API.....18
 - URL interface.....18
 - Script Player.....19
 - IBM Cognos toolkits.....19
 - The Java toolkit..... 20
 - The .NET Framework toolkit..... 23
 - Reporting examples..... 24
 - Example - connect to an IBM Cognos Analytics server.....25
 - Example - log on to an IBM Cognos Analytics server.....26
 - Example - run a report..... 28
 - Example - delete a report..... 33
 - Example - schedule a report.....36
 - Example - modify a report.....40
 - Example - run the script player.....44
- Chapter 4. Managing security.....45**
 - Authenticating Users..... 45
 - Authentication Providers.....45
 - Custom Authentication Provider.....45
 - Connect to IBM Cognos Analytics.....45
 - Log On.....46
 - Log Off.....47
 - Managing Authorization.....47
 - Before You Begin.....48
 - Managing Users, Groups, and Roles.....48
 - Managing Policies.....49
 - Managing Capabilities..... 51
- Chapter 5. Managing content..... 55**

Data Integrity.....	55
Internal Consistency.....	55
Concurrency Control.....	55
Classes and Properties.....	56
Property Values and Characteristics.....	56
Multilingual Properties.....	56
General Properties.....	56
Changing the Properties of an Object.....	57
Retentions Property.....	57
Relationships Between Objects.....	57
Parents and Children.....	58
References.....	59
Querying the Content Store.....	60
Example: Querying the Content Store in Java.....	60
Steps.....	60
Organizing Objects.....	60
Creating a Folder Object.....	61
Creating a URL Object.....	61
Creating a Shortcut Object.....	61
Copying an Object.....	61
Moving an Object.....	61
Controlling the Visibility of Objects.....	61
Disabling an Object.....	63
Deleting an Object.....	63
Setting the Order of Objects.....	63
Managing User Accounts.....	63
Deleting a User Account.....	63
Copying a User Account.....	64
Maintain the Content Store.....	64

Chapter 6. Running tasks..... 65

Using run().....	65
How IBM Cognos Builds a Request.....	66
Specifying Options and Parameters.....	66
Managing Event History.....	70
Scheduling Tasks.....	70
Retrying Tasks.....	72
Running Service-based Tasks.....	72
Running Reports.....	72
Running Jobs.....	76
Running Agents.....	76
Running Content Manager Tasks.....	77
Running Metrics Tasks.....	77
Running Notification Tasks.....	77
Understanding the Asynchronous Conversation.....	78
Primary and Secondary Requests.....	78
Using wait().....	78
Optimizing the Asynchronous Conversation.....	79
The Conversation Context.....	80
Processing Constraints.....	83

Chapter 7. Dispatchers and services.....85

Standby Dispatchers.....	85
Before Using the IBM Cognos Software Development Kit for Server Administration.....	85
Read the Availability of Services and Studios.....	85
Read the Status of Dispatchers and Services.....	86
Stop and Start Dispatchers and Services.....	86

Remove Dispatchers from the Environment.....	87
Steps.....	87
Test Dispatchers.....	87
Step.....	87
Activate a Content Manager Service.....	87
Set the Logging Level.....	88
Logging Properties.....	88
Steps.....	89
Tune Server Performance.....	89
Models.....	89
Operating Systems.....	89
Tuning Properties.....	89
Steps.....	90
Group Dispatchers in Configuration Folders.....	90
Steps.....	90
Define Server Groups.....	90
Steps.....	90
Configure Advanced Routing.....	90
Steps.....	91
Configure Advanced Settings.....	91
Steps.....	92
Use Advanced Routing.....	92
Steps to Use Advanced Routing for Content Store Objects.....	92
Steps to Use Advanced Routing for Inline Specifications.....	93
Hot Swapping PowerCubes.....	93
Steps.....	93
Chapter 8. Deploying content.....	95
Creating a Deployment Specification.....	95
Exporting Data From the Content Store.....	95
Moving the Deployment Archive.....	95
Importing Data Into the Content Store.....	95
Accessing Deployment History.....	96
Listing the Deployment Archives.....	96
Retrieving a Set of Deployment Options.....	96
Chapter 9. Coding practices and troubleshooting.....	97
Coding Practices.....	97
Managing Service Headers in Java Applications.....	97
Handling Exceptions.....	99
Language-specific Naming Conventions.....	108
Recommendations for URLs.....	108
Specifying Parameter Values of Type xsdDateTime or xsdDate.....	109
Replacing Illegal Characters in MIME Headers.....	110
Interpreting Messages that Use Response.Write Format.....	110
Localizing API Elements for Non-English Locales.....	110
Test Connections to Data Sources.....	110
Troubleshooting.....	110
End of line characters removed for MHT and XLWA output.....	110
Long-running Processes Do Not Complete.....	111
Data Source Connections are Ambiguous.....	111
UpdateMetadata Method Fails Unless the IBM Cognos Software Development Kit is Installed	
On The Server.....	111
Using updateMetadata With Multiple Servers.....	111
Problems Encountered with Object Retention in Content Manager.....	111
SAP BW Issues Error Messages.....	112
Problems Using the Java Print Sample on Windows.....	112

Report Output in CSV Format Contains an Extra Character.....	112
Base64 Decoding XLS or singleXLS Formats Creates Invalid Files.....	112
IBM Cognos Analytics Reports Run More Slowly than Expected.....	113
Unable to View Metadata in Query Studio After Adding a Query.....	113
Client Applications Cannot Connect to IBM Cognos.....	113
IBM Cognos Portal Services Extended Applications Portlet Not Working.....	114
A Namespace with a defaultName Property Cannot be Updated.....	114
IBM Cognos Access Manager will Attempt to Authenticate a User who Runs a Report Associated with an Unauthenticated Data Source.....	115
Multithreaded Java Client Causes Deserializer Not Found Error.....	115
Enabling Apache Axis Logging.....	115
Using Axis Attachment Support in Java Applications.....	116
Unable to View Running Reports or Jobs.....	116
Value keyTransformation of Enumeration Set validateSeverityEnum Only Partially Works.....	116
triggerName Issues.....	116
The Caption of an Unrecoverable Error is not Displayed in the Browser.....	117
Using the selectRoles Method has no Effect.....	117
The Logged Information for Multiple Report Runs are Collapsed under the Same Request Id.....	117
Deprecated Methods may Timeout before Returning a Response.....	117
Performance Problems in Java Applications due to Serialization of many Items with xs:nil value of true.....	117
Use of SystemMetricThresholds in the Object Model Differs from the Documentation.....	118
ui.drillThroughTargetParameterValues Not For External Use.....	118
Application Development Guidelines.....	118
Recommendation - Managing Long-Running or Resource-Intensive Tasks.....	119
Recommendation - Managing biBusHeader Objects in Multithreaded Applications.....	120
Chapter 10. Upgrading SDK applications.....	121
General Guidelines for Upgrading your Applications.....	121
Using What's New Appendices to Review Changes.....	121
Upgrading to IBM Cognos Analytics Version 10.2.0.....	121
Changes to .NET Framework support.....	121
Upgrading to IBM Cognos Analytics Version 10.1.0.....	122
Upgrading Java Applications for Axis 1.4.....	122
Changes to the cam_passport cookie.....	123
Changes to the Microsoft Excel Output Formats.....	124
Multi-Instance IBM Cognos Connection.....	124
Upgrading to IBM Cognos Analytics Version 8.4.....	124
Changes to Report Specification Validation.....	125
Upgrading to IBM Cognos Analytics Version 8.3.....	125
Changes to Deployment Options.....	125
Chapter 11. Schema reference.....	127
bibus.....	127
Chapter 12. Services.....	129
Services in Java.....	129
Services in C# .NET.....	129
List of Services.....	130
agentService.....	131
batchReportService.....	132
contentManagerService.....	134
dataIntegrationService.....	136
dataMovementService.....	137
deliveryService.....	138
dimensionManagementService.....	139
dispatcher.....	140

eventManagementService.....	141
idVizService.....	142
indexSearchService.....	142
indexUpdateService.....	143
jobService.....	145
metadataService.....	146
migrationService.....	147
mobileService.....	148
monitorService.....	149
planningAdministrationConsoleService.....	150
planningRuntimeService.....	150
planningTaskService.....	151
powerPlayService.....	151
queryService.....	153
relationalMetadataService.....	153
reportService.....	154
repositoryService.....	156
saCAMService.....	156
systemService.....	157

Chapter 13. Method sets..... 159

agent.....	159
asynch.....	159
authentication.....	160
content.....	160
dataSource.....	161
delivery.....	162
deployment.....	162
dispatcher.....	162
drillThrough.....	162
event.....	163
indexTerm.....	164
indexUpdate.....	164
metadata.....	165
monitor.....	165
paging.....	165
parameter.....	166
promptPaging.....	166
report.....	167
rolapCubeAdministration.....	168
standaloneCAM.....	168
system.....	169
validate.....	169

Chapter 14. Methods.....171

Methods in Java.....	171
activate(searchPath).....	171
activateURI(uri).....	172
add(parentPath, objects, options).....	173
add(objectPath, parameterValues, options).....	176
add(parentPath, object, options).....	177
addAnnotations(containerPath, objects, options).....	181
addDrillPath(parentPath, object, options).....	183
addNotification(objectPath).....	185
addTermAssociation(term, parameterValues, options).....	187
back(conversation, parameterValues, options).....	188
background(conversation).....	190

cancel(conversation).....	191
cancelEvent(eventID).....	194
cancelEvents(eventIDs).....	195
clearCubeWorkloadLog(cubeNames, parameterValues, options).....	196
clearNotifications(objectPath).....	198
collectParameterValues(objectPath, parameterValues, options).....	200
collectParameterValues(objectPath, parameterValues, options) – obsolete.....	202
collectParameterValuesSpecification(specification, parameterValues, options).....	203
collectParameterValuesSpecification(specification, parameterValues, options) – obsolete.....	207
convertDrillThroughContext(inputContext, parameterValues, options).....	208
copy(objects, targetPath, options).....	209
copyAccount(sourceAccountPath, targetAccountPath, options).....	212
copyRename(objects, targetPath, newNames, options).....	213
currentPage(conversation, parameterValues, options).....	215
delayEventsFor(eventIDs, for).....	217
delayEventsUntil(eventIDs, until).....	219
delete(objects, options).....	220
delete(objectPath, parameterValues, options).....	222
deleteAccount(objectPath, options).....	224
deleteAllNotifications().....	225
deleteHotList(objectPath).....	226
deleteNotification(objectPath).....	227
deleteTenants(tenantIDs).....	229
deleteTermAssociation(term, parameterValues, options).....	230
deliver(conversation, parameterValues, options).....	231
determineRouting(objectPaths).....	235
drill(conversation, parameterValues, options).....	236
findDrillThroughPaths(objectPath, parameterValues, options).....	238
firstPage(conversation, parameterValues, options).....	239
forward(conversation, parameterValues, options).....	242
get(objectPath, parameterValues, options).....	244
getActiveContentManager().....	245
getConfiguration(properties).....	246
getContext(conversation, parameterValues, options).....	247
getCubeMessages(cubeNames, parameterValues, options).....	249
getCubeMetrics(cubeNames, parameterValues, options).....	251
getCubeState(cubeNames, parameterValues, options).....	252
getDeploymentOptions(archive, options).....	254
getFormatSamples(name).....	256
getIdentity().....	257
getObjectContext(objectPath, parameterValues, options).....	258
getOutput(conversation, parameterValues, options).....	259
getParameters(objectPath, parameterValues, options).....	262
getParametersSpecification(specification, parameterValues, options).....	265
getPromptValues(conversation, parameterValues, options).....	268
getPromptValues(conversation, parameterValues, options) – obsolete.....	271
holdEvent(eventID).....	272
holdEvents(eventIDs).....	273
incrementallyLoadCubes(cubeNames, parameterValues, options).....	274
lastPage(conversation, parameterValues, options).....	276
lineage(conversation, parameterValues, options).....	279
listArchives().....	280
listTenants(options).....	282
logoff().....	282
logon(credentials, roles).....	283
mapContentLocale(locale, normalize).....	286
mapProductLocale(locale, normalize).....	287
move(objects, targetPath, options).....	288

moveRename(objects, targetPath, newNames, options).....	290
nextPage(conversation, parameterValues, options).....	292
normalizeLocale(locale).....	295
pauseCubes(cubeNames, parameterValues, options).....	296
ping(dispatcherPath).....	298
previousPage(conversation, parameterValues, options).....	299
query(searchPath, properties, sortBy, options).....	302
query(objectPath, parameterValues, options).....	305
queryDrillPath(objectPath, parameterValues, options).....	308
queryMetadata(request).....	310
queryMultiple(requests).....	311
queryMultipleCache(requests, options).....	312
queryNotification(objectPath).....	314
queryTenantMembership(tenantIDs).....	315
refreshCubeDataCache(cubeNames, parameterValues, options).....	316
refreshCubeMemberCache(cubeNames, parameterValues, options).....	318
refreshCubeSecurity(cubeNames, parameterValues, options).....	320
release(conversation).....	322
releaseEvent(eventID).....	325
releaseEvents(eventIDs).....	326
render(conversation, parameterValues, options).....	328
restartCubes(cubeNames, parameterValues, options).....	330
retrieveCredential(namespace).....	331
run(objectPath, parameterValues, options).....	333
runAt(startTime, objectPath, parameterValues, options).....	340
runSpecification(specification, parameterValues, options).....	342
scheduleEvent(eventID).....	353
scheduleEvents(eventIDs).....	354
selectRoles(roles).....	355
startCubes(cubeNames, parameterValues, options).....	356
startService(servicePath).....	358
stopCubes(cubeNames, parameterValues, options).....	359
stopService(servicePath, immediately).....	361
terminateSessions(search).....	363
testDataSourceConnection(connectionString, credentials).....	364
testDataSourceConnection(connectionString, credentials) – obsolete.....	366
testDataSourceConnectionWithInfo(connectionString, credentials).....	367
trigger(triggerName).....	368
update(objects, options).....	370
update(object, options).....	373
updateDrillPath(object, options).....	376
updateEvents(events).....	377
updateMetadata(request).....	379
validate(objectPath, parameterValues, options) – obsolete.....	380
validate(objectPath, parameterValues, options).....	381
validateContentLocale(locale).....	384
validateProductLocale(locale).....	385
validateSpecification(specification, parameterValues, options) – obsolete.....	386
validateSpecification(specification, parameterValues, options).....	387
wait(conversation, parameterValues, options).....	391

Chapter 15. Classes..... 397

Classes in Java.....	397
account.....	397
adaptiveAnalyticsService.....	406
addOptions.....	410
addressSMTP.....	411

addressSMTPArrayProp.....	412
addressSMTPProp.....	412
adminFolder.....	412
agentDefinition.....	414
agentDefinitionView.....	415
agentOption.....	416
agentOptionBoolean.....	416
agentOutputEnumProp.....	416
agentOutputHotList.....	417
agentService.....	418
agentState.....	420
agentTaskDefinition.....	421
agentTaskState.....	424
aliasLocation – obsolete.....	425
aliasRoot.....	425
analysis.....	426
ancestorInfo.....	427
ancestorInfoArrayProp.....	428
ancestorInfoProp.....	428
annotation.....	429
annotationFolder.....	430
annotationService.....	431
anyTypeMIMEProp.....	433
anyTypeProp.....	433
anyURIArrayProp.....	435
anyURIProp.....	435
archiveDescriptor.....	436
archiveLocation.....	438
archiveOption.....	439
archiveOptionConflictResolution.....	440
archiveOptionDescriptor.....	440
archiveOptionSearchPathSingleObject.....	441
archiveOptionString.....	441
asynchDetail.....	442
asynchDetailAgentNotificationStatus.....	442
asynchDetailAsynchSpecification.....	443
asynchDetailContext.....	443
asynchDetailCount.....	444
asynchDetailDrillPathObject.....	444
asynchDetailDrillThroughRequest.....	445
asynchDetailDrillThroughTarget.....	446
asynchDetailDrillThroughTargetURI.....	448
asynchDetailEventID.....	449
asynchDetailEventRecord.....	449
asynchDetailIndexData.....	453
asynchDetailIndexTerms.....	455
asynchDetailMessages.....	455
asynchDetailMIMEAttachment.....	456
asynchDetailParameters.....	457
asynchDetailParameterValue.....	457
asynchDetailPromptPage.....	458
asynchDetailReportMetadata.....	458
asynchDetailReportObject.....	458
asynchDetailReportOutput.....	459
asynchDetailReportStatus.....	460
asynchDetailReportValidation.....	461
asynchDetailROLAPDataSourceState.....	461
asynchDetailROLAPMessages.....	462

asynchDetailROLAPMetrics.....	462
asynchDetailSelectionContext.....	463
asynchDetailUnstructuredData.....	463
asynchOption.....	464
asynchOptionBoolean.....	464
asynchOptionEncoding.....	465
asynchOptionInt.....	465
asynchOptionSearchPathSingleObject.....	465
asynchOptionSearchPathSingleObjectArray.....	466
asynchOptionStringArray.....	466
asynchReply.....	466
asynchRequest.....	468
asynchSecondaryRequest.....	470
asynchSpecification.....	471
auditLevelEnumProp.....	472
authoredAgentDefinition.....	476
authoredPowerPlay8Report.....	477
authoredReport.....	479
base64BinaryMIMEProp.....	482
baseAgentDefinition.....	483
baseAgentDefinitionActionEnumProp.....	488
baseClass.....	489
baseClassArrayProp.....	499
baseClassProp.....	502
baseDataIntegrationTask.....	502
baseDataMovementTask.....	507
baseHistoryDetail.....	510
baseParameter.....	512
baseParameterAssignment.....	513
baseParameterAssignmentArrayProp.....	514
baseParameterAssignmentProp.....	514
basePowerPlay8Report.....	514
basePowerPlay8ReportActionEnumProp.....	518
basePowerPlayClass.....	518
baseProp.....	520
baseReport.....	523
baseReportActionEnumProp.....	530
baseROLAPCubeConfiguration.....	530
baseROLAPCubeConfigurationArrayProp.....	532
baseROLAPCubeConfigurationProp.....	533
baseROLAPDataSource.....	533
baseRoutingRule.....	535
baseRSSTask.....	535
batchReportService.....	537
biBusHeader.....	544
biBusHeaderExtension1.....	546
booleanProp.....	546
boundRangeParmValueItem.....	548
cacheOutput.....	549
CAF.....	550
CAM.....	551
CAMException.....	554
CAMPassport.....	555
CAMProtect.....	557
CAMSettings.....	558
capability.....	558
catalog.....	559
catalogFolder.....	559

classEnumArrayProp.....	560
classEnumProp.....	560
configuration.....	561
configurationData.....	625
configurationFolder.....	628
contact.....	685
content.....	690
contentManagerCacheService.....	692
contentManagerQueryOption.....	694
contentManagerQueryOptionBoolean.....	695
contentManagerQueryOptionInt.....	695
contentManagerQueryOptionPropEnumArray.....	695
contentManagerQueryOptionRefPropArray.....	696
contentManagerQueryOptionString.....	696
contentManagerService.....	696
contentStoreUtilizationConfiguration.....	699
contentTask.....	700
contentTaskOption.....	703
contentTaskOptionAuditLevelEnum.....	703
contentTaskOptionBoolean.....	704
contentTaskOptionClassEnumArray.....	704
contentTaskOptionCSUtilizationConfiguration.....	705
contentTaskOptionRetentionRuleArray.....	705
contentTaskOptionSearchPathSingleObjectArray.....	705
contextOption.....	706
contextOptionFormat.....	706
contextOptionStringArray.....	707
contextOptionType.....	707
conversationContext.....	707
cookieVar.....	709
copyAccountOption.....	709
copyAccountOptionBoolean.....	710
copyOptions.....	710
credential.....	711
currency.....	713
currencyArrayProp.....	714
currencyProp.....	714
dashboard.....	715
dataAdvisorService.....	718
dataIntegrationService.....	719
dataIntegrationServiceSpecification.....	721
dataIntegrationServiceSpecificationArrayProp.....	722
dataIntegrationServiceSpecificationProp.....	722
dataIntegrationTaskOption.....	722
dataIntegrationTaskOptionBoolean.....	723
dataMovementService.....	723
dataMovementServiceSpecification.....	728
dataMovementServiceSpecificationProp.....	728
dataMovementTask.....	728
dataMovementTaskAlias.....	730
dataSet.....	731
dataSetFolder.....	732
dataSource.....	732
dataSourceCommandBlock.....	734
dataSourceCommandBlockProp.....	734
dataSourceConnection.....	735
dataSourceCredential.....	737
dataSourceNameBinding.....	738

dataSourceSignon.....	739
dateTimeProp.....	740
decimalProp.....	741
deleteOptions.....	741
deliveryOption.....	742
deliveryOptionAddressSMTPArray.....	743
deliveryOptionChannel.....	743
deliveryOptionMemoPart.....	744
deliveryOptionSearchPathMultipleObjectArray.....	744
deliveryOptionSearchPathSingleObject.....	744
deliveryOptionString.....	745
deliveryService.....	745
deliveryServiceSpecification.....	747
deploymentDetail.....	748
deploymentImportRule.....	750
deploymentObjectInformation.....	751
deploymentOption.....	752
deploymentOptionAnyType.....	753
deploymentOptionArrayProp.....	753
deploymentOptionAuditLevel.....	753
deploymentOptionBoolean.....	754
deploymentOptionClassEnumArray.....	754
deploymentOptionImportRuleArray.....	754
deploymentOptionMultilingualString.....	755
deploymentOptionObjectInformationArray.....	755
deploymentOptionProp.....	755
deploymentOptionResolution.....	756
deploymentOptionSearchPathSingleObjectArray.....	756
deploymentOptionString.....	757
deploymentOptionStringArray.....	757
deploymentReference.....	757
deploymentReferenceArrayProp.....	758
deploymentReferenceProp.....	759
deploymentStatusEnumProp.....	759
dimensionManagementService.....	759
dimensionManagementServiceSpecification.....	764
directory.....	764
dispatcher.....	765
dispatcherTransportVar.....	829
displayObject.....	830
distributionList.....	831
document.....	833
documentContent.....	835
documentVersion.....	839
drillOption.....	840
drillOptionParameterValues.....	840
drillPath.....	840
drillThroughAction.....	844
drillThroughOption.....	844
drillThroughOptionAnyURI.....	845
drillThroughOptionBoolean.....	845
drillThroughOptionXMLEncodedXML.....	845
drillThroughPath.....	846
durationProp.....	846
environmentVar.....	847
eventManagementService.....	847
eventManagementServiceSpecification.....	849
eventRecord.....	849

eventTypeEnumArrayProp.....	851
eventTypeEnumProp.....	851
EVService.....	852
exportDeployment.....	853
faultDetail.....	856
faultDetailArrayProp.....	856
faultDetailMessage.....	857
faultDetailProp.....	857
favoritesFolder.....	858
floatProp.....	858
folder.....	859
formFieldVar.....	863
gatewayMapping.....	864
gatewayMappingArrayProp.....	864
gatewayMappingProp.....	865
genericOption.....	865
genericOptionAnyURI.....	866
genericOptionBoolean.....	866
genericOptionStringArray.....	867
genericOptionXMLEncodedXML.....	867
governor.....	867
governorArrayProp.....	868
governorBoolean.....	868
governorInt.....	869
governorProp.....	869
ggraphic.....	869
graphicsService.....	871
ggroup.....	874
guid.....	877
guidArrayProp.....	877
guidProp.....	878
hdrSession.....	878
hierarchicalParmValueItem.....	879
history.....	880
historyDetail.....	883
historyDetailAgentService.....	884
historyDetailDataMovementService.....	884
historyDetailDeploymentSummary.....	885
historyDetailMigrationService.....	886
historyDetailRelatedHistory.....	887
historyDetailRelatedReports.....	888
historyDetailReportService.....	888
historyDetailRequestArguments.....	889
humanTask.....	889
humanTaskService.....	892
identity.....	894
idVizService.....	895
importDeployment.....	896
indexDataService.....	898
indexOption.....	900
indexOptionBoolean.....	900
indexOptionInt.....	901
indexOptionSearchPathMultipleObjectArray.....	901
indexSearchService.....	902
indexSearchServiceSpecification.....	904
indexTerm.....	904
indexTermOption.....	905
indexTermOptionBoolean.....	905

indexTermOptionIndexTermArray.....	906
indexUpdateService.....	906
indexUpdateServiceSpecification.....	908
indexUpdateServiceSpecificationProp.....	909
indexUpdateTask.....	909
installedComponent.....	912
installedComponentEnumProp.....	912
interactiveReport.....	913
intProp.....	914
jobDefinition.....	925
jobOption – deprecated.....	929
jobOptionBoolean – deprecated.....	929
jobService.....	930
jobStepDefinition.....	931
languageProp.....	933
languageArrayProp.....	934
launchable.....	934
listTenantsOptions.....	936
loadBalancingModeEnumProp.....	936
locale.....	937
localeArrayProp.....	938
localeMapEntry.....	938
localeMapEntryArrayProp.....	938
localeMapEntryProp.....	939
localeProp.....	939
logService.....	939
mapOption.....	941
memo.....	941
memoPart.....	943
memoPartAgentObject.....	944
memoPartAgentObjectLinks.....	945
memoPartComposite.....	946
memoPartMIMEAttachment.....	946
memoPartObject.....	947
memoPartObjectLinks.....	948
memoPartString.....	948
message.....	948
metadataModelExpression.....	949
metadataModelItemName.....	949
metadataModelItemNameArrayProp.....	949
metadataModelItemNameProp.....	950
metadataService.....	950
metadataServiceLineageSpecification.....	954
metadataServiceModelInformationSpecification.....	955
metadataServiceSpecification.....	955
metricsDataSourceETLTask.....	956
metricsExportTask.....	956
metricsFileImportTask.....	956
metricsMaintenanceTask.....	957
metricsManagerService.....	957
migrationCubeMapping.....	959
migrationMapping.....	959
migrationService.....	960
migrationServiceSpecification.....	962
migrationServiceSpecificationProp.....	963
migrationTask.....	963
migrationTaskOption.....	966
migrationTaskOptionMappingArray.....	967

migrationTaskOptionResolution.....	967
migrationTaskOptionSearchPathSingleObject.....	967
migrationTaskOptionSearchPathSingleObjectArray.....	968
mobileOption.....	968
mobileOptionSearchPathMultipleObjectArray.....	969
mobileService.....	969
model.....	971
modelView.....	974
monitorOption.....	974
monitorOptionBoolean.....	974
monitorService.....	975
monitorServiceSpecification.....	976
moveOptions.....	976
mruFolder.....	977
multilingualString.....	979
multilingualStringProp.....	979
multilingualToken.....	980
multilingualTokenProp.....	980
namespace.....	981
namespaceFolder.....	983
nameValue.....	984
navigationPath.....	985
navigationPathArrayProp.....	986
navigationPathProp.....	986
ncnameProp.....	986
nil.....	987
nmtokenArrayProp.....	987
nmtokenProp.....	987
nonNegativeIntegerProp.....	988
objectLink.....	989
option.....	990
optionArrayProp.....	993
optionProp.....	994
output.....	995
package.....	999
packageActionEnumProp.....	1005
packageConfiguration.....	1005
page.....	1007
pageDefinition.....	1008
pagelet.....	1009
pageletFolder.....	1010
pageletInstance.....	1011
pagingOption.....	1012
pagingOptionInt.....	1012
parameter.....	1013
parameterAssignmentDataItem.....	1014
parameterDataSource.....	1014
parameterValue.....	1014
parameterValueArrayProp.....	1020
parameterValueProp.....	1021
parmValueItem.....	1021
pdfCharacterEncodingEnumProp.....	1022
pdfCompressionTypeEnumProp.....	1023
pdfFontEmbeddingEnumProp.....	1023
pdfOption.....	1024
pdfOptionBoolean.....	1025
pdfOptionPrintQuality.....	1025
pdfOptionXMLEncodedXML.....	1025

periodical.....	1026
permission.....	1027
personalization.....	1027
personalizationFolder.....	1028
pingReply.....	1029
planningAdministrationConsoleService.....	1030
planningAdministrationConsoleServiceSpecification.....	1032
planningAdministrationConsoleServiceSpecificationProp.....	1032
planningApplication.....	1033
planningDataService.....	1034
planningMacroTask.....	1037
planningRuntimeService.....	1040
planningRuntimeServiceSpecification.....	1041
planningStateEnumProp.....	1042
planningTask.....	1042
planningTaskService.....	1045
planningTaskServiceSpecification.....	1047
planningTaskServiceSpecificationProp.....	1047
policy.....	1048
policyArrayProp.....	1048
policyProp.....	1049
portal.....	1049
portalOption.....	1050
portalOptionBoolean.....	1050
portalOptionDisplayMode.....	1050
portalOptionInt.....	1051
portalOptionListSeparator.....	1051
portalOptionSearchPathSingleObject.....	1051
portalOptionSearchPathSingleObjectArray.....	1052
portalOptionString.....	1052
portalOptionXMLEncodedXML.....	1052
portalPackage.....	1053
portalSkin.....	1054
portalSkinFolder.....	1056
portlet.....	1056
portletFolder.....	1058
portletInstance.....	1059
portletProducer.....	1059
positiveIntegerProp.....	1062
powerPlay8Option.....	1062
powerPlay8OptionAnyURI.....	1063
powerPlay8OptionBoolean.....	1064
powerPlay8OptionData.....	1064
powerPlay8OptionLanguageArray.....	1064
powerPlay8OptionOutputFormat.....	1065
powerPlay8OptionSaveAs.....	1065
powerPlay8OptionSearchPathSingleObject.....	1066
powerPlay8OptionString.....	1066
powerPlay8OutputFormatEnumArrayProp.....	1067
powerPlay8OutputFormatEnumProp.....	1067
powerPlay8Report.....	1068
powerPlay8ReportView.....	1068
powerPlayCube.....	1069
powerPlayDataBlock.....	1069
powerPlayDataBlockArrayProp.....	1070
powerPlayDataBlockProp.....	1070
powerPlayOption.....	1070
powerPlayOptionArrayProp.....	1071

powerPlayOptionBoolean.....	1071
powerPlayOptionData.....	1072
powerPlayOptionOutputFormat.....	1072
powerPlayOptionProp.....	1072
powerPlayReport.....	1073
powerPlayReportActionEnumProp.....	1073
powerPlayService.....	1074
powerPlayServiceReportSpecification.....	1078
powerPlayServiceSpecification.....	1078
powerPlayServiceSpecificationProp.....	1079
presentationService.....	1079
printer.....	1080
promptInfo.....	1081
promptOption.....	1082
provider.....	1083
productCapability.....	1083
query.....	1084
queryCountOptions.....	1085
queryCountResult.....	1085
queryMultipleOptions.....	1085
queryMultipleResult.....	1086
queryOptions.....	1086
queryReply.....	1088
queryRequest.....	1088
queryService.....	1090
queryServiceTask.....	1097
refProp.....	1099
relationalMetadataService.....	1100
relationalMetadataServiceSpecification.....	1104
report.....	1104
reportCache.....	1105
reportDataService.....	1107
reportDataServiceAgentDefinition.....	1108
reportDataServiceSpecification.....	1109
reportDataServiceSpecificationProp.....	1110
reportService.....	1110
reportServiceAnalysisSpecification.....	1117
reportServiceDrillThroughSpecification.....	1117
reportServiceDrillThroughSpecificationProp.....	1118
reportServiceInteractiveReportSpecification.....	1118
reportServiceMetadataSpecification.....	1119
reportServiceQueryDrillPathOption.....	1119
reportServiceQueryDrillPathOptionBoolean.....	1120
reportServiceQueryOption.....	1120
reportServiceQueryOptionBoolean.....	1121
reportServiceQueryOptionSpecificationFormat.....	1121
reportServiceQuerySpecification.....	1121
reportServiceReportSpecification.....	1122
reportServiceSpecification.....	1122
reportStudioOption - deprecated.....	1122
reportStudioOptionSearchPathSingleObject - deprecated.....	1123
reportTemplate.....	1123
reportVersion.....	1124
reportView.....	1127
repositoryRule.....	1128
repositoryRuleArrayProp.....	1129
repositoryRuleProp.....	1130
repositoryService.....	1130

resource.....	1132
retentionRule.....	1133
retentionRuleArrayProp.....	1135
retentionRuleProp.....	1136
rolapCubeConfiguration.....	1136
rolapCubeMessages.....	1139
rolapCubeMetrics.....	1140
rolapDataSource.....	1140
rolapMessage.....	1141
rolapMetric.....	1141
rolapMetricDateTime.....	1142
rolapMetricNumber.....	1142
rolapMetricString.....	1144
rolapVirtualCubeConfiguration.....	1144
rolapVirtualDataSource.....	1144
role.....	1145
root.....	1148
routingHintObject.....	1149
routingInfo.....	1150
routingRuleObject.....	1150
routingTableEntry.....	1151
routingTableEntryArrayProp.....	1151
routingTableEntryProp.....	1152
rssOption.....	1152
rssOptionAnyURI.....	1152
rssOptionSearchPathSingleObject.....	1153
rssOptionString.....	1153
runConditionEnumProp.....	1153
runningStateEnumProp.....	1154
runOption.....	1155
runOptionAnyURI.....	1156
runOptionArrayProp.....	1156
runOptionBoolean.....	1157
runOptionData.....	1157
runOptionDateTime.....	1157
runOptionInt.....	1158
runOptionLanguageArray.....	1158
runOptionMultilingualString.....	1158
runOptionNameValueArray.....	1159
runOptionOutputEncapsulation.....	1159
runOptionPromptCacheMode.....	1159
runOptionProp.....	1160
runOptionSaveAs.....	1160
runOptionString.....	1161
runOptionStringArray.....	1161
runTimeState.....	1161
saCAMService.....	1163
schedule.....	1164
scheduledEvent.....	1171
schemaInfo.....	1172
searchPathMultipleObject.....	1173
searchPathSingleObject.....	1174
searchPathSingleObjectArrayProp.....	1176
searchPathSingleObjectProp.....	1176
securedFeature.....	1176
securedFunction.....	1177
session.....	1177
setCookieVar.....	1179

severityEnumProp.....	1180
shortcut.....	1181
shortcutAgentRSSTask.....	1182
shortcutRSSTask.....	1182
simpleParmValueItem.....	1183
softwareEdition.....	1184
softwareEditionArrayProp.....	1185
softwareEditionProp.....	1185
sort.....	1186
specification.....	1186
specificationOption.....	1186
specificationOptionString.....	1187
specificationOptionXMLEncodedXML.....	1187
SQL.....	1188
storedProcedureTask.....	1188
stringArrayProp.....	1191
stringMapEntry.....	1192
stringMapEntryArrayProp.....	1192
stringMapEntryProp.....	1193
stringProp.....	1193
subscriptionFolder.....	1198
subscriptionOption.....	1199
subscriptionOptionSearchPathSingleObject.....	1199
systemMetricThresholds.....	1200
systemService.....	1202
tenancy.....	1203
tenant.....	1204
tenantInfo.....	1205
tenants.....	1205
timeProp.....	1206
timeZone.....	1206
tokenArrayProp.....	1207
tokenProp.....	1207
tracking.....	1208
transientStateFolder.....	1209
uiClass.....	1210
uiComponentEnumArrayProp.....	1216
uiComponentEnumProp.....	1216
uiProfile - deprecated.....	1217
uiProfileFolder - deprecated.....	1218
unboundedEndRangeParmValueItem.....	1219
unboundedStartRangeParmValueItem.....	1219
updateOptions.....	1220
uriValue.....	1221
uriValueArrayProp.....	1222
uriValueProp.....	1222
URL.....	1222
urlRSSTask.....	1224
userCapabilityCache.....	1225
userCapabilityEnumArrayProp.....	1225
userCapabilityEnumProp.....	1226
userCapabilityPermission.....	1226
userCapabilityPolicy.....	1227
userCapabilityPolicyArrayProp.....	1228
userCapabilityPolicyProp.....	1228
userInterfaceProfile.....	1228
userPreferenceVar.....	1229
validateOption.....	1230

validateOptionBoolean.....	1231
validateOptionHint.....	1231
validateOptionValidateSeverity.....	1231
visualization.....	1232
webServiceTask.....	1233
xmlEncodedXML.....	1236
xmlEncodedXMLArrayProp.....	1238
xmlEncodedXMLMIMEProp.....	1238
xmlEncodedXMLProp.....	1239

Chapter 16. Enumeration sets..... 1241

Enumeration Sets in Java.....	1241
accessEnum.....	1242
agentNotificationStatusEnum.....	1242
agentOptionEnum.....	1242
agentOutputEnum.....	1243
archiveConflictResolutionEnum.....	1243
archiveOptionEnum.....	1244
asynchDetailReportStatusEnum.....	1245
asynchOptionEnum.....	1245
asynchReplyStatusEnum.....	1247
asynchStatusEnum.....	1247
auditLevelEnum.....	1248
automaticAggregateOptimizationEnum.....	1252
baseAgentDefinitionActionEnum.....	1252
basePowerPlay8ReportActionEnum.....	1253
baseReportActionEnum.....	1253
baseTextDirectionEnum.....	1254
biDirectionalOptionEnum.....	1254
bindingQualifierEnum.....	1255
classEnum.....	1255
configurationDataEnum.....	1266
conflictResolutionEnum.....	1267
contentManagerQueryOptionEnum.....	1267
contentTaskOptionEnum.....	1268
contextFormatEnum.....	1271
contextOptionEnum.....	1271
contextTypeEnum.....	1272
copyAccountOptionEnum.....	1272
dashboardOptionEnum.....	1273
dataEnum.....	1273
dataIntegrationTaskOptionEnum.....	1273
dataSourceCapabilityEnum.....	1275
daysEnum.....	1276
deliveryChannelEnum.....	1276
deliveryOptionEnum.....	1277
deploymentOptionEnum.....	1278
deploymentStatusEnum.....	1284
drillOptionEnum.....	1285
drillThroughActionEnum.....	1285
drillThroughOptionEnum.....	1286
drillThroughRecipientEnum.....	1287
encodingEnum.....	1288
eventTypeEnum.....	1288
formatEnum.....	1289
gcPolicyEnum.....	1289
governorEnum.....	1290

indexOptionEnum.....	1291
indexTermOptionEnum.....	1293
installedComponentEnum.....	1293
isolationLevelEnum.....	1298
jobOptionEnum – obsolete.....	1298
linearUnitEnum.....	1299
loadBalancingModeEnum.....	1299
mapLayerTypeEnum.....	1300
mapOptionEnum.....	1300
migrationTaskOptionEnum.....	1301
mobileOptionEnum.....	1302
monitorOptionEnum.....	1302
monthsEnum.....	1303
namespaceCapabilityEnum.....	1304
orderEnum.....	1305
outputEncapsulationEnum.....	1305
outputFormatEnum.....	1306
packageActionEnum.....	1308
pageOrientationEnum.....	1308
pagingOptionEnum.....	1308
parameterCapabilityEnum.....	1309
parameterDataTypeEnum.....	1310
pdfCharacterEncodingEnum.....	1311
pdfCompressionTypeEnum.....	1312
pdfFontEmbeddingEnum.....	1313
pdfOptionEnum.....	1314
pdfPrintQualityEnum.....	1315
planningStateEnum.....	1316
portalDisplayModeEnum.....	1316
portalListSeparatorEnum.....	1316
portalOptionEnum.....	1317
powerPlay8DataEnum.....	1318
powerPlay8OptionEnum.....	1318
powerPlay8OutputFormatEnum.....	1320
powerPlay8SaveAsEnum.....	1321
powerPlayDataBlockEnum.....	1321
powerPlayDataEnum.....	1321
powerPlayOptionEnum.....	1322
powerPlayOutputFormatEnum.....	1322
powerPlayReportActionEnum.....	1323
promptCacheModeEnum.....	1323
promptTypeEnum.....	1324
propEnum.....	1324
queryModeEnum.....	1356
queryOptionEnum.....	1356
queryProcessingEnum.....	1357
queryTaskOptionEnum.....	1357
reportSaveAsEnum.....	1362
reportServiceQueryDrillPathOptionEnum.....	1362
reportServiceQueryOptionEnum.....	1362
reportStudioOptionEnum - deprecated.....	1363
rolapDataSourceStateEnum.....	1363
rolapOptionEnum.....	1364
rssOptionEnum.....	1364
runConditionEnum.....	1365
runningStateEnum.....	1366
runOptionEnum.....	1367
runStatusEnum.....	1373

scheduleDailyPeriodEnum.....	1375
scheduleEndTypeEnum.....	1375
scheduleTypeEnum.....	1376
selectionContextFormatEnum.....	1379
sequencingEnum.....	1379
severityEnum.....	1380
smtpContentDispositionEnum.....	1381
smtpContentTypeEnum.....	1381
specificationFormatEnum.....	1381
specificationOptionEnum.....	1382
subscriptionOptionEnum.....	1383
systemMetricEnum.....	1383
systemMetricThresholdsPropertyEnum.....	1386
systemOptionEnum.....	1387
temporaryObjectLocationEnum.....	1388
uiComponentEnum.....	1388
updateActionEnum.....	1389
usageEnum.....	1390
userCapabilityEnum.....	1391
validateHintEnum.....	1414
validateOptionEnum.....	1415
validateSeverityEnum.....	1415
weeksEnum.....	1416
Chapter 17. Secondary requests.....	1417
Example: Checking Valid Secondary Requests in Java.....	1418
Chapter 18. Code samples and language-specific coding practices.....	1419
Active Server Page Samples.....	1419
ASP Sample File Locations.....	1419
Java Samples.....	1419
Java Sample File Locations.....	1420
Set Up Your System to Use the Java Samples.....	1422
C# .NET Samples.....	1424
C# .NET Sample File Locations.....	1426
Set Up Your System to Use the C# .NET Samples.....	1426
Chapter 19. Language-specific data type mappings.....	1427
C# Equivalents.....	1427
Java Equivalents.....	1427
XML-SQL Data Type Mappings.....	1428
Chapter 20. Performing tasks using URLs.....	1431
Recommendations for URLs.....	1431
CGI Program and Alternative Gateways.....	1431
URL Methods.....	1432
Parameterized URL Method.....	1432
cognosLaunch Method.....	1433
Common Optional Parameters.....	1434
URL Validation.....	1434
Starting IBM Cognos Analytics Components.....	1435
Start Parameters.....	1435
Starting Query Studio.....	1435
Starting Analysis Studio.....	1437
Starting Metric Studio.....	1438
Starting Event Studio.....	1439
Starting IBM Cognos Viewer.....	1440

Starting PowerPlay Studio.....	1442
Starting IBM Cognos Analytics Components in a Specified Browser Window.....	1443

Chapter 21. URL API reference.....1447

IBM Cognos Viewer.....	1447
asynch.<option>.....	1447
run.<option>.....	1447
cv.contextinfo.....	1449
cv.contextmenu.....	1449
cv.debug.....	1449
cv.drill.....	1449
cv.header.....	1449
cv.navlinks.....	1450
cv.selection.....	1450
cv.toolbar.....	1450
p_<parameterName>.....	1450
parameterValues.....	1450
ui.action.....	1450
ui.backURL.....	1450
ui.conversation.....	1450
ui.errURL.....	1451
ui.folder.....	1451
ui.format.....	1451
ui.gateway.....	1451
ui.name.....	1451
ui.object.....	1451
ui.primaryAction.....	1451
ui.productLocale.....	1452
ui.routingServerGroup.....	1452
ui.spec.....	1452
ui.tool.....	1452

Chapter 22. Passing environment variables to external applications..... 1453

Types of Environment Variables.....	1453
Example - Passing Environment Variables.....	1454
Specify the Names of the Environment Variables to be Passed.....	1454
Sample Environment Variables File.....	1455

Chapter 23. Using report specifications..... 1457

Use the SDK to Modify Report Specifications.....	1457
Retrieve a Report Specification.....	1457
Modify a Report Specification.....	1458
Save the Report Specification to the Content Store.....	1458
Report Specification Validation.....	1459
Use Reporting to View Report Specification XML.....	1460
Basic Structure of a Report Specification.....	1461
Queries Element.....	1462
Layouts Element.....	1463
Types of Data Containers.....	1465
List Reports.....	1465
Sample of a List Report.....	1466
Group Data in a Column.....	1469
Set the Group Span for a Column.....	1471
Add List Headers.....	1472
Sort Data.....	1473
Sample of a Crosstab Report.....	1474
Query Portion of the Report.....	1474

Layout Portion of the Crosstab Report.....	1475
Report Formatting.....	1478
Default Style Sheet.....	1478
Use Local Class Styles to Define and Reuse Styles.....	1478
Conditional Formatting.....	1479
Formatting Data.....	1480
Report Variables.....	1480
Support Multiple Locales.....	1481
Optimize Query Performance.....	1482
Use Query Hints.....	1483
Reuse Queries.....	1485
Summarize Data.....	1486
Set the Type of Aggregate.....	1486
Show Detail Rows and Rollup Summaries.....	1488
Create a Calculated Column.....	1489
Ways of Filtering Data	1490
Add a Filter and a Prompt Sample.....	1491
Filter Measure Values in a Crosstab Report Sample.....	1492
Filter Data With a Subquery.....	1493
Define a Slicer.....	1495
Chapter 24. Creating custom report functions and function sets.....	1497
Creating custom report functions.....	1497
Building a custom report functions library.....	1497
Registering custom report functions.....	1500
Installing a custom report functions library.....	1502
Example of a custom report functions implementation.....	1503
Creating custom report functions.....	1505
Custom function sets.....	1505
Creating a custom function set.....	1505
Example of creating a custom function set.....	1506
Chapter 25. Using lineage specifications.....	1509
Creating lineage requests for reports.....	1509
Sample request XML for a saved report.....	1509
Creating lineage requests for packages or models.....	1513
Sample request XML for package data.....	1513
Understanding lineage responses from metadataService.....	1514
Integrating a custom lineage solution	1517
Example: Creating a Web interface to customize the lineage request parameters.....	1518
Security considerations.....	1520
Source mappings.....	1520
Chapter 26. Lineage specification reference.....	1529
childRef.....	1529
connection.....	1529
lineageRequest.....	1529
lineageResponse.....	1530
name.....	1530
object.....	1530
objectQuery.....	1530
objectQueryRef.....	1531
objectRef.....	1531
param.....	1531
parentRef.....	1532
property.....	1532
queryResult.....	1532

service.....	1533
transformation.....	1533
type.....	1533
type.....	1533
V5QuerySet.....	1534
Chapter 27. Understanding dynamic drill-through.....	1535
Chapter 28. Drill-through specification examples.....	1537
Example - Drill-through to an IBM Cognos Analytics report.....	1537
Sample XML.....	1537
Example - Drill-through to a package.....	1538
Sample XML.....	1538
Chapter 29. Drill-through specification reference.....	1541
action.....	1541
alternateItems.....	1541
bookmark.....	1542
characterEncoding.....	1542
drillThroughSpecification.....	1543
extension.....	1543
item.....	1544
iwrTarget.....	1544
modelSearchPaths.....	1545
packageTarget.....	1545
parameterAssignment.....	1545
parameterAssignments.....	1546
parameterName.....	1546
powerPlay7ReportTarget.....	1547
powerPlay8ReportTarget.....	1547
reportTarget.....	1547
scope.....	1548
searchPath.....	1548
studios.....	1548
targetSearchPath.....	1549
targetURI.....	1549
text.....	1550
uri.....	1550
uriTarget.....	1551
Chapter 30. Edit specification examples.....	1553
Example - Edit specification for entire selection context.....	1553
Sample XML.....	1553
Example - Edit specification demonstrating various data value alternatives.....	1554
Sample XML.....	1554
Sample XML - Edit specification demonstrating various data item operations.....	1555
Example - Edit specification demonstrating various data item operations with xml:base.....	1556
Sample XML.....	1556
Chapter 31. Edit specification reference.....	1557
actions.....	1557
argument.....	1557
arguments.....	1558
dataItem.....	1558
editSpecification.....	1558
end.....	1559
expression.....	1559

extension.....	1559
filter.....	1560
name.....	1560
null.....	1560
nullOrBlank.....	1560
operator.....	1561
operator.....	1561
range.....	1561
simple.....	1562
start.....	1562
value.....	1562
value.....	1563
Chapter 32. Search path syntax.....	1565
Find the Search Path for an Object.....	1565
Steps.....	1565
Search Path Examples.....	1565
Search Path Syntax.....	1567
Location Steps.....	1567
Axes and Node Tests.....	1568
Predicates.....	1569
Functions.....	1569
Chapter 33. Advanced settings configuration.....	1577
Advanced settings reference.....	1577
Agent service advanced settings.....	1577
Content Manager service advanced settings.....	1578
Common configuration settings.....	1583
Presentation service advanced settings.....	1584
Delivery service advanced settings.....	1587
Dispatcher service advanced settings.....	1588
Event management service advanced settings.....	1589
Job service advanced settings.....	1591
Metrics manager service advanced settings.....	1591
Monitor service advanced settings.....	1592
Report service and batch report service advanced settings.....	1597
Repository service advanced settings.....	1606
Chapter 34. Initial content store settings.....	1607
Initial Objects.....	1607
Predefined Objects.....	1607
Description.....	1607
Object Hierarchy.....	1608
/.....	1614
Administration.....	1615
Capability.....	1616
Adaptive Analytics.....	1616
Administration.....	1617
Adaptive Analytics Administration.....	1618
Administration tasks.....	1618
Configure and manage the system.....	1619
Controller Administration.....	1620
Data Source Connections.....	1620
Directory - obsolete.....	1621
Distribution Lists and Contacts.....	1621
Metric Studio Administration.....	1622
Mobile Administration.....	1623

My Data Sets Administration.....	1623
Planning Administration.....	1624
PowerPlay Servers.....	1625
Printers.....	1625
Query Service Administration.....	1626
Run activities and schedules.....	1626
Set capabilities and manage UI profiles.....	1627
Styles and portlets.....	1628
Users, Groups, and Roles.....	1629
Analysis Studio.....	1629
Open PowerPlay Reports with Analysis Studio.....	1630
Cognos Insight.....	1631
Cognos Viewer.....	1631
Context Menu.....	1632
Run With Options.....	1633
Selection.....	1634
Toolbar.....	1634
Collaborate.....	1635
Allow collaboration features.....	1635
Launch collaboration tools.....	1636
Controller Studio.....	1636
Data Manager.....	1637
Detailed Errors.....	1637
Drill Through Assistant.....	1638
Event Studio.....	1639
EVStudio.....	1639
Execute Indexed Search.....	1640
Executive Dashboard.....	1640
Use Advanced Dashboard Features.....	1641
Use Interactive Dashboard Features.....	1642
Use the Edit Features.....	1643
External Repositories.....	1644
Manage repository connections.....	1644
View external documents.....	1645
Generate CSV Output.....	1645
Generate PDF Output.....	1646
Generate XLS Output.....	1647
Generate XML Output.....	1647
Glossary.....	1648
Hide Entries.....	1648
Import relational metadata.....	1649
Lineage.....	1650
Manage own data source signons.....	1650
Metric Studio.....	1651
Edit View.....	1651
Mobile.....	1652
My Data Sets.....	1653
Package Data Sources.....	1653
Planning Contributor.....	1654
PowerPlay Studio.....	1655
Query Studio.....	1655
Advanced.....	1656
Create.....	1656
Reporting.....	1657
Allow External Data.....	1657
Bursting.....	1658
Create/Delete.....	1659
HTML Items in Report.....	1659

Open PowerPlay Reports with Reporting.....	1660
User Defined SQL.....	1661
Scheduling.....	1661
Schedule by day.....	1662
Schedule by hour.....	1663
Schedule by minute.....	1663
Schedule by month.....	1664
Schedule by trigger.....	1664
Schedule by week.....	1665
Schedule by year.....	1665
Scheduling Priority.....	1665
SDK.....	1666
Self Service Package Wizard.....	1667
Set Entry-Specific Capabilities.....	1667
Specification Execution.....	1668
Statistics.....	1668
Catalog - removed.....	1669
Watch Rules.....	1669
Library.....	1670
Visualizations.....	1671
User Interface Profiles.....	1672
Cognos Workspace Do More Profiles.....	1672
Reporting Profiles.....	1673
Configuration.....	1673
11x17.....	1674
A3.....	1674
A4.....	1675
B4 JIS.....	1676
B5 JIS.....	1676
Legal.....	1677
Letter.....	1677
User Interface Profiles - deprecated.....	1678
Reporting Profiles - deprecated.....	1678
Express - deprecated.....	1679
Professional - deprecated.....	1680
User Profile.....	1681
Most Recently Used list.....	1681
My Folders.....	1682
My Watch Items.....	1682
Directory.....	1683
Tenants.....	1683
<Indeterminate>.....	1684
<Known User>.....	1684
<Indeterminate>.....	1685
Most Recently Used List.....	1686
My Folders.....	1686
My Watch Items.....	1687
Cognos.....	1687
Adaptive Analytics Administrators.....	1688
Adaptive Analytics Users.....	1688
All Authenticated Users.....	1689
Analysis Users.....	1689
Anonymous.....	1690
<Indeterminate>.....	1691
Most Recently Used list.....	1691
My Folders.....	1692
My Watch Items.....	1692
Authors.....	1693

Catalog Administrators - removed.....	1693
Cognos Insight Users.....	1694
Consumers.....	1694
Controller Administrators.....	1695
Controller Users.....	1696
Data Manager Authors.....	1696
Directory Administrators.....	1697
Everyone.....	1698
Express Authors.....	1698
Library Administrators.....	1699
Metrics Administrators.....	1700
Metrics Authors.....	1700
Metrics Users.....	1701
Mobile Administrators.....	1702
Mobile Users.....	1702
Planning Contributor Users.....	1703
Planning Rights Administrators.....	1703
Portal Administrators.....	1704
PowerPlay Administrators.....	1705
PowerPlay Users.....	1705
Query Users.....	1706
Readers.....	1707
Report Administrators.....	1707
Server Administrators.....	1708
System Administrators.....	1708
Tenant Administrators.....	1709
Export.....	1710
Import.....	1710
Portal.....	1711
Administration.....	1711
Pagelets.....	1712
Console.....	1712
Configuration.....	1713
Content Administration.....	1713
Data Source Connections.....	1714
Dispatchers and Services.....	1714
Distribution Lists and Contacts.....	1715
Portlets.....	1715
Printers.....	1716
Styles.....	1717
Index Search.....	1717
Index.....	1718
Search.....	1718
Storage.....	1719
Multitenancy.....	1719
Tenants.....	1720
Library.....	1720
User Interface Profiles.....	1721
Visualizations.....	1721
PowerPlay.....	1722
Security.....	1722
Capabilities.....	1723
User Interface Profiles.....	1723
Users, Groups, and Roles.....	1724
Status.....	1725
Current Activities.....	1725
Data Sets.....	1726
Past Activities.....	1726

Schedules.....	1727
System.....	1727
Upcoming Activities.....	1728
Portlets.....	1728
Console.....	1729
cogadmin.....	1729
Capabilities.....	1730
Content Administration.....	1730
Current Activities.....	1731
Data Sets.....	1732
Dynamic Cubes.....	1732
Data Source Connections.....	1732
Dispatchers and Services.....	1733
Distribution Lists and Contacts.....	1734
Dynamic Cubes.....	1734
Past Activities.....	1735
Portlets.....	1735
PowerPlay.....	1736
Printers.....	1736
Profiles Administration.....	1737
Schedules.....	1737
Styles.....	1738
System.....	1738
Upcoming Activities.....	1739
Users Groups and Roles.....	1739
IBM Cognos Go! Search.....	1740
IBM Cognos Enhanced Search - command panel.....	1740
IBM Cognos Enhanced Search - External Search Results.....	1741
IBM Cognos Enhanced Search - main UI.....	1741
IBM Cognos Enhanced Search - Refinement Viewer.....	1742
IBM Cognos Enhanced Search - Results Viewer.....	1742
IBM Cognos Go! Search Admin.....	1743
Index.....	1743
Search.....	1744
Storage.....	1744
Multitenancy.....	1745
Tenants.....	1745
Connection.....	1746
Pages.....	1746
Portlets.....	1747
Dashboard.....	1747
Multi-page.....	1748
IBM Cognos Content.....	1748
IBM Cognos Navigator.....	1749
IBM Cognos Search.....	1749
IBM Cognos Viewer.....	1750
IBM Cognos Extended Applications.....	1750
IBM Cognos Extended Applications Portlet.....	1751
IBM Cognos Metric Studio.....	1751
IBM Cognos History Chart.....	1752
IBM Cognos Metric List.....	1753
IBM Cognos Utility.....	1753
Bookmarks Viewer.....	1754
HTML Source.....	1754
HTML Viewer.....	1755
Image Viewer.....	1755
RSS Viewer.....	1756
Styles.....	1756

Business.....	1757
Classic.....	1757
Contemporary.....	1758
Corporate.....	1758
Modern.....	1758
Presentation.....	1759
Public Folders.....	1759
Transient.....	1762
Job.....	1762
Notebook.....	1763

Chapter 35. Metadata schema reference..... 1765

MAQuery Elements.....	1765
Actions.....	1765
BlockConstraint.....	1766
Constraint.....	1766
Constraint.....	1767
Constraints.....	1767
Constraints.....	1767
Functions.....	1768
Metadata.....	1768
Path.....	1769
Properties.....	1769
Properties.....	1770
Property.....	1770
Property.....	1771
request.....	1772
Start_at.....	1772
MAResponse Elements.....	1772
calculation.....	1773
decimalSeparator.....	1773
defaultLocale.....	1773
dimension.....	1774
effectiveLocale.....	1774
filter.....	1775
folder.....	1775
function.....	1776
functionsRoot.....	1776
group.....	1777
hierarchy.....	1777
level.....	1777
listSeparator.....	1778
locale.....	1778
locales.....	1779
measure.....	1779
member.....	1780
modelSearchPath.....	1781
package.....	1782
queryItem.....	1782
queryItemFolder.....	1784
querySubject.....	1784
ResponseRoot.....	1785
Data Type Mapping for Response.....	1785

Chapter 36. Event specification reference..... 1787

detailSeverity.....	1787
dispatcherID.....	1788

display.....	1788
end.....	1788
endTime.....	1788
eventID.....	1789
filters.....	1789
groupBy.....	1789
objectClass.....	1789
order.....	1790
owner.....	1790
parentEventID.....	1790
priority.....	1791
queryEventSpecification.....	1791
restartEventID.....	1791
restartParentEventID.....	1792
scheduleTrigger.....	1792
scheduleType.....	1792
scope.....	1793
searchPath.....	1793
searchPaths.....	1793
sort.....	1793
sortItem.....	1794
sortItem.....	1794
start.....	1794
startTime.....	1795
status.....	1795
subSort.....	1796
summary.....	1796
user.....	1796
Chapter 37. Using selection context.....	1799
Viewing a selection context in Reporting.....	1799
Selection context example.....	1799
Selection context for a multidimensional model.....	1801
Chapter 38. Selection context reference.....	1803
axis.....	1803
cell.....	1804
cells.....	1804
definingCells.....	1805
dependentCells.....	1805
extension.....	1805
metadataCell.....	1806
metadataCells.....	1807
s.....	1807
selection.....	1808
sf.....	1808
strings.....	1809
value.....	1809
values.....	1810
Chapter 39. Repository Service API reference.....	1811
List report repositories.....	1811
List report versions.....	1812
List report version outputs.....	1813
Retrieve output content.....	1815
External references in HTML output.....	1815
Retrieve default output.....	1816

Appendix A. New in lineage specification.....	1817
Schema version 10.1.0.....	1817
Appendix B. New in Version 11.1.2.....	1819
Support for Jobs and Notebooks.....	1819
New Capability Cookie.....	1819
Appendix C. New in Version 10.2.2.....	1821
Connecting to a datasource using namespace credentials.....	1821
Data Stores renamed to Dynamic Cubes in Cognos Administration.....	1821
My data sets.....	1821
Report output format restriction.....	1821
Data source connection information.....	1822
Dynamic cube messages.....	1822
Dynamic cube metrics.....	1822
New options to manage dynamic cubes.....	1823
Full tenant impersonation capability for system administrators.....	1823
Changes to IBM Cognos Analytics SOAP action for services.....	1823
WebSphere Liberty Profile replaces Apache Tomcat.....	1824
New dynamic cube configuration properties.....	1824
Reporting user interface profiles.....	1825
Deprecation of Reporting profiles.....	1825
Dynamically rename excel sheet in a multi-tab excel output.....	1826
Support for delegated tenant administration.....	1826
Support for new advanced settings.....	1826
Documentation Updates.....	1827
Appendix D. New in Version 10.2.1.....	1829
External object store for report archiving.....	1829
Support for archiving a namespace or namespaceFolder.....	1829
Enhanced tenant administration functionality.....	1829
Performance enhancements.....	1830
Removal of statistics service.....	1830
Administrative changes for Cognos Analytics Mobile Reports.....	1830
Changes to IBM Cognos Analytics SOAP action for services.....	1831
Visualization support.....	1831
New configuration options for burst distribution.....	1832
New standalone IBM Cognos Access Manager (CAM) service.....	1832
Developer description.....	1833
Repository Service API supports retrieval of default report output.....	1833
End of line characters removed for MHT and XLWA output.....	1833
Appendix E. New in Version 10.2.0.....	1835
Relational metadata service.....	1835
Support for bi-directional languages.....	1835
Graduated dashboard capabilities.....	1835
IBM Cognos Dynamic Cubes Administration.....	1836
IBM Cognos Dynamic Cubes Aggregate Advisor configuration.....	1836
New queryService configuration options.....	1837
New standalone IBM Cognos Access Manager (CAM) service.....	1838
Support for multi-tenancy.....	1838
Deprecation of IBM Cognos PowerPlay capabilities and objects.....	1839
Support for interactive discovery and visualization.....	1839
Support for SAP BW hierarchy variables.....	1839
Changes to IBM Cognos Analytics SOAP action for services.....	1840
Repository service.....	1840

Excel 2007 Data output format.....	1841
IBM Business Process Server integration.....	1841
Restricting access to Cognos Insight in IBM Cognos Analytics.....	1841
Restricting access to Cognos Analytics Mobile Reports Service in IBM Cognos Analytics.....	1842
Changes to .NET Framework support.....	1842
Updated default settings for Report Service and Batch Report Service.....	1842
New Repository Service (REST) API.....	1843
ReportNET API is obsolete.....	1843

Appendix F. New in Version 10.1.1..... 1845

Support for IBM Cognos Content Archival.....	1845
JMX Proxy Server Scalability.....	1846
Changes to IBM Cognos Analytics SOAP action for services.....	1846
Run with Owner Capabilities.....	1846
HttpOnly Attribute Support for the CAM_Passport Cookie.....	1847
IBM Cognos ROLAP Administration.....	1847
Deprecation of qsDisableQueryPlanCache.....	1849
Personal Data Source Signons.....	1849
Maximum process configuration values for the statistics service are not used.....	1849
Changes to monitor service advanced settings.....	1850
Improving the utilization of system resources by using the asynch » release(conversation) method.....	1850
Changes to runSpecification() method for eventManagementService and monitorService.....	1850

Appendix G. New in Version 10.1.0..... 1851

External Data.....	1851
Personal Data Source Credentials.....	1851
Flexible Scheduling.....	1851
Scheduling — Better Administrative Oversight.....	1852
Delaying Scheduled Tasks.....	1852
New for Report Specification Schema Version 7.0.....	1853
Support for Axis 1.4.....	1853
New Data Integration Service Task Option.....	1853
Human Task Service.....	1854
Variable Support for Data Movement Tasks.....	1854
Support for Lineage Requests.....	1855
Query Service.....	1855
Adaptive Analytics Service.....	1856
Storing Temporary Output Objects Outside the Content Store.....	1857
IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace.....	1858
IBM Cognos TM1 Packages.....	1858
Relational Metadata Service.....	1858
Accessibility.....	1859
Graphics Service.....	1860
New PowerPlay Service Methods.....	1860
Supporting Fail-over for Tree Prompts.....	1861
Content Cleanup Content Manager Task.....	1861
Content Pane Improvements.....	1862
Content Manager Cache Service.....	1862
Human Task.....	1862
Portal Skin Management.....	1863
Multi-Instance IBM Cognos Connection.....	1863
Personal Packages.....	1863
Launchable.....	1864
Annotation Service.....	1864
Visual Basic 6.0 Support.....	1864
CAM Passport Changes.....	1864
Statistics Service.....	1865

Use Report Name for Output File Name.....	1865
Query Modes.....	1865
Production of Excel 2000 Format Report Outputs is Obsolete.....	1866
Support for .NET 1.1 Removed.....	1866
IBM Cognos Content Manager/Enterprise Content Management Integration.....	1866
Object Updates.....	1867
Package Capabilities Overwritten on Default Import/Export.....	1867
IBM Cognos TM1 and IBM Cognos Workspace Integration.....	1867
Collaboration Tool Integration.....	1867
New Email Configuration Parameters.....	1868
IBM Cognos Active Report.....	1869
Query Service Administration Task.....	1869
Batch Report Service/Report Service Optimizations.....	1870
Data Collection Options for Indexing Tasks.....	1870
JMX Proxy Server Scalability.....	1871
Updated Support for IBM Cognos Express.....	1871
Run with Owner Capabilities.....	1871
Old Drill-through API Removed.....	1871
Integration of Adaptive Analytics and Dashboards.....	1872
Object Documentation Updates.....	1872
Object Updates.....	1876
Previously Reserved Index Options Now Available for Client Use.....	1877
Object Capabilities Properties.....	1877
New Limitations for Setting Object Ownership.....	1877
Administration Capabilities for Background Job Queue Maintenance.....	1877
Changes to JSP Tags in IBM Cognos Portal Services.....	1878
Documentation Updates.....	1878

Appendix H. New in Version 9.0.0.....1879

Support for IBM Cognos Express.....	1879
-------------------------------------	------

Appendix I. New in Version 8.4 GA..... 1881

Statistics.....	1881
Capability Cookie Format Changed.....	1881
Lineage - Filter Support.....	1881
Documentation Updates.....	1882
Lineage Specification Changes.....	1882
Bug Fixes.....	1882

Appendix J. New in Version 8.4..... 1883

Hiding Objects in the Portal.....	1883
Report Server deliver() method.....	1883
Migration Service.....	1883
On Demand Refresh of Prompt Cache.....	1884
Object Capabilities.....	1885
PowerPlay 8 Integration.....	1886
pagedLayoutDataXML Report Output Format.....	1889
Drill-Through Improvements.....	1889
Dynamic Filtering of Report Data.....	1889
Writing Task History Subsets.....	1890
Supporting New Drill-through Targets.....	1890
Mobile Service.....	1891
Migration Capabilities.....	1891
Lineage Metadata.....	1891
URL Validation Rules.....	1893
Index Options.....	1893
Adaptive Analytics Integration.....	1893

Software Editions.....	1893
Dashboards.....	1894
Updating the Credential of Imported Schedules.....	1895
Package Data Sources.....	1895
Report Output Annotations.....	1895
Dimension Management Service.....	1896
Support for IBM® WebSphere® Business Glossary.....	1897
userInterface property of the uiProfile class.....	1897
Bug Fixes.....	1897

Appendix K. New in Version 8.3.....1899

Conditional Subscriptions.....	1899
Report Email Alerts.....	1900
Agent Default Portal Action.....	1901
Agent Task Event Filters.....	1901
Saving Report Output to File System.....	1901
Improve Batch Processing.....	1902
Administration Console.....	1906
Schedule Management.....	1907
System Metrics.....	1908
Task Retry.....	1909
Chart Hotspot Limit.....	1909
PDF Options - Password Protection.....	1910
spreadsheetML Output Format.....	1910
Reporting Profiles.....	1910
Disabling Selection-based Interactive Report Output Features.....	1911
Improved Context Metadata for Selection.....	1911
E-Mail Activity and Administration.....	1911
Schedule Priority.....	1912
Run Agent Tasks on Failure.....	1912
Advanced Features for Report Output.....	1912
Drill Through Assistant Capability.....	1913
Package Hierarchies.....	1913
Object Icons.....	1914
Dashboard Fragments.....	1914
Deferring Creation of History Objects.....	1915
Deployment History.....	1915
Parameter Method Set.....	1915
PDF Configuration Parameters.....	1915
Email Delivery Enhancements.....	1916
Data Movement Service Configuration Parameters.....	1917
Capabilities Refinements.....	1918
Client Token.....	1918
New Security Roles.....	1918
Default Value Change.....	1919
New Properties for Group and Role Classes.....	1919
Notification List.....	1920
mruFolder Retention Rules.....	1920
Search Capability.....	1920
Search — For Internal Use Only.....	1920

Notices.....1923

Glossary.....1925

A.....	1925
B.....	1925
C.....	1926
D.....	1926

F.....	1927
G.....	1927
H.....	1927
I.....	1927
J.....	1927
L.....	1927
M.....	1928
N.....	1928
P.....	1928
Q.....	1929
R.....	1929
S.....	1929
U.....	1929

Index..... 1931

Chapter 1. Introduction

This document is intended for use with IBM® Cognos® Software Development Kit.

The IBM Cognos Software Development Kit allows you to manage IBM Cognos processes and implement custom reporting solutions by using the cross-platform Web services, libraries, and programming interfaces provided with the IBM Cognos Software Development Kit. As a developer, you can use the Software Development Kit to create and modify reports and queries, schedule and deploy reports and other objects, and administer IBM Cognos Analytics servers and security.

This document describes in detail how you can develop applications using the IBM Cognos Software Development Kit. It also contains detailed reference information about the Software Development Kit.

This document uses notational conventions based on conventions used by the world wide web consortium (<http://www.w3.org/standards/about.html>). For example, the "this namespace" (tns) prefix is used as a convention to refer to the types of properties and arguments in the BI Bus API.

Audience

To use the IBM Cognos Software Development Kit Developer Guide effectively, you should know how to authenticate users and be familiar with the following:

- Reporting, Query Studio, and Framework Manager
- XML, HTML, WSDL, and SOAP
- active server page (.asp) or Java™ server page (.jsp) basics
- XSL style sheets and your standard Web page layout
- Java or Microsoft Visual Studio .NET

Finding information

To find product documentation on the web, including all translated documentation, access [IBM Knowledge Center](http://www.ibm.com/support/knowledgecenter) (<http://www.ibm.com/support/knowledgecenter>).

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

The Sample Outdoors Company, Great Outdoors Company, GO Sales, any variation of the Sample Outdoors or Great Outdoors names, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.

Accessibility features

Consult the documentation for the tools that you use to develop applications to determine their accessibility level. These tools are not a part of this product.

IBM Cognos HTML documentation has accessibility features. PDF documents are supplemental and, as such, include no added accessibility features.

Chapter 2. Understanding the IBM Cognos Software Development Kit

As a developer, you can use the IBM Cognos Software Development Kit to integrate IBM Cognos into your existing software or portals and to provide several methods for developing custom solutions. The IBM Cognos Software Development Kit provides the following tools for integrating with IBM Cognos services and components:

- IBM Cognos toolkits
 - to provide programmers with access to the BI Bus API in Java and the Microsoft® .NET Framework programming environments
- A URL interface
 - to automate actions by passing commands and parameters over HTTP to integrate with other applications
- Published IBM Cognos DTD or Schema and documentation
 - to support the development of customized solutions
- Custom report function libraries
 - to provide customized report functions to report authors
- Published IBM Cognos Framework Manager schemas and documentation
 - to support the development and publishing of Framework Manager models.

IBM Cognos Architecture

IBM Cognos is a solution designed to address the challenges of enterprise-scale reporting, analysis, scorecarding, and event notification.

The web-based IBM Cognos architecture was designed for scalability, availability, and openness. It uses platform independent, industry proven technology, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Definition Language (WSDL).

As shown in the following diagram, the IBM Cognos architecture is based on a typical three-tiered web architecture that consists of the following components:

a web server

The IBM Cognos web server tier contains one or more IBM Cognos gateways.

applications

The IBM Cognos applications tier contains one or more IBM Cognos servers. An IBM Cognos server runs requests, such as reports, analyses, and queries, that are forwarded by a gateway.

data

The IBM Cognos data tier consists of the content store, data sources, and the metric store.

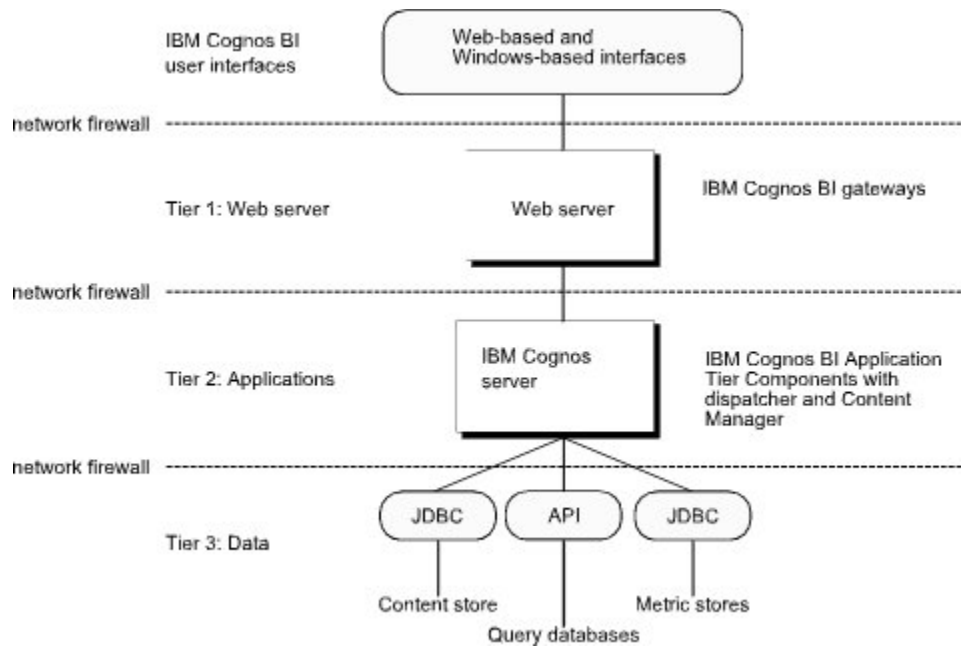


Figure 1. IBM Cognos server between a Web server and a Data tier

IBM Cognos components communicate with each other, and with any additional applications integrated with IBM Cognos, using the BI Bus API. The BI Bus API is an open, documented, SOAP API that supports WSDL, providing platform-independent access to all IBM Cognos functionality.

Service-Oriented Architecture

The IBM Cognos architecture includes a number of services for interacting with and managing runnable objects. A runnable object is an instance of a class that defines the actions performed by a service when a run request is received. The run request specifies the content store location of an object in IBM Cognos. For example, a report, job, or agent are examples of runnable objects that can be invoked interactively through a portal action or a client request, or by a scheduled event.

Services exist for each of the major components of the IBM Cognos product. Each of the services listed in this chapter communicates using the BI Bus API.

Each service shares a set of generic methods for running objects associated with the service. Users send requests directly to the appropriate service, based on the class of the object. For example, a request to run a report is sent to the report service while a request to run an agent is sent to the agent service. Individual services may provide additional methods to perform operations specific to the service. For example, the Content Manager service provides additional methods to move, copy, and rename objects in the content store.

To support this architecture, services execute tasks using a generalized asynchronous protocol. The protocol uses a consistent mechanism for passing parameters to standardize how tasks are executed. Classes and methods exist to support this consistent process of executing tasks. Asynchronous conversations are managed to optimize overall performance. For more information about asynchronous conversations and runnable objects, see [Chapter 6, “Running tasks,” on page 65](#).

To support a generic approach shared across services, the following mechanisms are also in place:

- A single, generic `bibus » history` class structure is shared by background tasks to record significant events during execution.
- The various options specified to run tasks are also unified under a single `bibus » option` class hierarchy.

Access and Authentication

Users send and receive SOAP messages using the BI Bus API as the route to the IBM Cognos service they require. To access these components, unless your users are configured for anonymous access, valid authentication information must be specified in the SOAP request header or through one of the toolkit's logon methods. Security providers check user credentials and send the information back to a client application using the BI Bus API.

For more information about access and authentication, see the *IBM Cognos Administration and Security Guide*.

Using the BI Bus API

When you use BI Bus API, you are working with the central communications channel between IBM Cognos services. The BI Bus API consists of a set of services and protocols that IBM Cognos services use to communicate with each other, and with external applications. It is not a software component like IBM Cognos Analytics - Reporting or the Content Manager, but a set of rules that govern communications between services, and that is published as a web service.

WSDL is an XML-based language used to define the services provided by a server. IBM Cognos is designed to take advantage of this Web service technology which, like SOAP and XML, makes IBM Cognos platform-independent and fully scalable.

The published WSDL file that represents the BI Bus API is part of the IBM Cognos Software Development Kit package. Access to services using the BI Bus API is achieved through a SOAP request transmitted over HTTP.

You use a SOAP message to send a request to a service. SOAP messages are encoded in XML and then transmitted using HTTP. Because these technologies work on all IBM Cognos platforms, components can be distributed among Microsoft Windows, UNIX, and Linux servers to optimize query performance.

The SOAP request consists of the following components:

- An HTTP header, required as part of transmitting a request using that protocol.
- A SOAPAction section that the IBM Cognos dispatcher uses to identify the target service for the request.
- A SOAP header and body that contains the information required to complete the request. The contents of the header and body must be compliant with the BI Bus API object model, as described in the *IBM Cognos Software Development Kit Developer Guide*.

The toolkits turn commands into SOAP messages that the BI Bus API then delivers to the appropriate service. These toolkits provide a Java and .NET toolkit for developing custom applications to automate IBM Cognos tasks and a rich library to access IBM Cognos functionality on the BI Bus API.

The BI Bus API wraps each XML-formatted request or response in a SOAP envelope and transmits it using HTTP.

IBM Cognos Analytics Services

After you install and configure a server, one dispatcher is available by default. Each dispatcher has a set of associated services, depending on the configuration, described in this section.

Adaptive Analytics service

Defines the service responsible for performing tasks for Analyst Add-in for IBM Cognos Analytic Applications client applications.

When a user invokes user interface elements within Analyst Add-in for IBM Cognos Analytic Applications client applications, a request is sent to this service to retrieve the metadata that is required to render the user interface elements, such as a **Table of Contents**. The adaptive analytics service also provides on-demand generation of query information based on user interaction with the client.

This service makes requests to

- the [contentManagerService](#) for access to objects in the content store, and for authorization information.
- the [metadataService](#) for metadata information that renders user interface elements in Analyst Add-in for IBM Cognos Analytic Applications client applications.
- the [reportService](#) to render reports for Analyst Add-in for IBM Cognos Analytic Applications client applications based on metadata changes in the user interface.

The adaptive analytics service has an associated set of configuration parameters defined by the properties of the [bibus » adaptiveAnalyticsService](#) class.

Agent service

Defines the service responsible for the execution of agents ([baseAgentDefinition](#) class), stored procedures ([storedProcedureTask](#) class), and web service methods ([webServiceTask](#) class).

The agent service uses the [monitorService](#) to execute each task in an [agentDefinition](#).

The agent service has a set of configuration parameters defined by the properties of the [agentService](#) class.

The agent service creates an [agentState](#) object to store data that is required during the execution of the agent. This data includes the [output](#) objects containing the agent events and the [agentOutputHotList](#).

If the [agentState](#) object is deleted the agent service will re-create it, but any events reported by the agent during re-creation are considered to be new.

Related information:

- [bibus » humanTaskService](#) class

Annotation service

The annotation service stores and manages comments against the context of a dashboard, report, report widget, or report element that is made accessible through IBM Cognos Workspace.

These comments persist across multiple versions of a report.

The annotation service also supports comments associated with tasks in **My Inbox** in IBM Cognos Connection.

This service makes requests to the [contentManagerService](#) for authorization information.

The annotation service receives requests from both the [bibus » reportDataService](#) class to support IBM Cognos Workspace features and from the [bibus » reportDataService](#) class to support features related to human tasks in IBM Cognos Connection.

Related information:

- [bibus » humanTaskService](#) class

Batch report service

Defines the service responsible for the execution of batch reports [baseReport](#) class.

The batch report service is dedicated to running non-interactive batch reports. For example, when a report is run as part of an agent or job or is run in the background, the request is sent to the batch report service. A report is run in the background when it is scheduled, emailed, printed, saved, or run in response to an external occurrence (see [event » trigger\(triggerName\)](#) method).

The batch report service has an associated set of configuration parameters that are defined by the properties of the [batchReportService](#) class. These parameters can be configured to optimize the batch report service for non-interactive use. This is the only difference between the batch report service and the report service ([reportService](#)).

This service may also make additional requests to the following IBM Cognos Analytics services when processing a request:

- [contentManagerService](#) service
- [deliveryService](#) service
- [eventManagementService](#) service
- [monitorService](#) service

Related information:

- [bibus](#) » [reportService](#) class
- [reportService](#) service

Content Manager service

Defines the service responsible for managing the content store.

In addition to managing the content store, the Content Manager service also performs authentication-related activities.

The Content Manager service has an associated set of configuration parameters defined by the properties of the [contentManagerService](#) class.

For more information about Content Manager, see [Chapter 5, “Managing content,” on page 55](#).

Content Manager cache service

Defines the service responsible for caching query results from dispatchers.

This enhances the overall system performance and Content Manager scalability by caching frequent query results in each dispatcher.

Data advisor service

Defines the service responsible for supporting IBM Cognos Express Data Advisor clients.

This service processes requests from Data Advisor clients to build OLAP cubes and register them on the IBM Cognos Express server.

Data integration service

Defines the service responsible for the execution of metrics-related tasks.

You can define metrics-related tasks using the [bibus](#) » [baseDataIntegrationTask](#) class.

The data integration service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [dataIntegrationService](#) class.

Data movement service

Manages the execution of data movement tasks in IBM Cognos Analytics.

Data movement tasks, such as Builds and JobStreams, are created in Data Manager Designer and published to IBM Cognos Analytics.

Delivery service

Defines the service responsible for guaranteed delivery of email to an external email service.

The delivery service is used by the [batchReportService](#) and the [reportService](#) to deliver report outputs.

The [agentService](#) uses the delivery service to execute [memo](#) objects that are part of an agent ([baseAgentDefinition](#) class).

The delivery service has an associated set of configuration parameters defined by the properties of the [deliveryService](#) class.

Dimension management service

Defines the service responsible for dimension management.

The dimension management service acts as a gateway to the IBM Cognos Business Viewpoint application.

The dimension management service has a set of configuration parameters defined by the properties of the [dimensionManagementService](#) class.

Dispatcher service

Defines the entry point for IBM Cognos Analytics service requests sent by a web server gateway or other software.

The dispatcher manages a set of associated services, performing activities such as routing requests to these services, as well as managing their configuration at run time.

You can also manage and configure dispatchers and their associated services using the server administration tool.

EV service

The service responsible for supporting IBM Cognos Express.

Event management service

Defines the service responsible for the management of events.

An event is any task that has run, is currently running, or will be run at a future time.

The event management service manages the event execution queue. Events in the queue can be held for execution at a later time or removed from the queue. Events that are held can later be released for execution.

Events to be run on an irregular basis can also be manually added to the event execution queue. These events are usually on-demand tasks or tasks that are executed in response to an external trigger condition.

The [contentManagerService](#) works in conjunction with the event management service to update the event execution queue whenever a [schedule](#) is added to, updated in, or deleted from the content store.

The event management service has an associated set of configuration parameters defined by the properties of the [eventManagementService](#) class.

Graphics service

Defines the service responsible for producing graphics on behalf of the [reportService](#) service.

Graphics (charts) that are used in reports can be generated in Raster, Vector, Microsoft[®] Excel XML, or PDF format.

The graphics service has an associated set of configuration parameters defined by the properties of the [graphicsService](#) class.

Human task service

The human task service creates and manages human tasks used in IBM Cognos Event Studio and IBM Cognos Workspace.

A human task is either a notification of information or an action to be performed by a person, such as a report approval request. Human tasks can be assigned to users manually by another user, or automatically by another service.

This service makes requests to the [contentManagerService](#) and [deliveryService](#) for managing notifications. It receives requests from [agentService](#) to support automated human task generation and [reportDataService](#) to support dashboard features.

Index services

The [indexSearchService](#) service and [indexUpdateService](#) service support search functionality in IBM Cognos products.

Reserved.

Job service

Defines the service responsible for the execution of jobs ([jobDefinition](#) class).

The job service uses the [monitorService](#) to execute each step in a job.

The job service has an associated set of configuration parameters defined by the properties of the [jobService](#) class.

Log service

Records log messages generated by the dispatcher and other services.

The log service can be configured to record log information in a file, a database, a remote log server, Windows Event Viewer, or a UNIX[®] system log.

Metadata service

Defines the service responsible for querying and updating unpublished IBM Cognos Framework Manager models.

The metadata service has an associated set of configuration parameters defined by the properties of the [metadataService](#) class.

Metrics Manager service

Provides support for IBM Cognos Metrics Manager for managing performance information.

Migration service

Defines the service that provides for the migration of content to IBM Cognos.

This service can be used to perform the following migrations:

- [powerPlayReport](#) (IBM Cognos Series 7 PowerPlay) to [powerPlay8Report](#) (PowerPlay[®])
- [powerPlayReport](#) (IBM Cognos Series 7 PowerPlay) to [analysis](#) (IBM Cognos Analysis Studio) or [report](#) (IBM Cognos Analytics - Reporting)
- [powerPlay8Report](#) (IBM Cognos PowerPlay Studio) to [analysis](#) (IBM Cognos Analysis Studio) or [report](#) (IBM Cognos Analytics - Reporting)

The migration service has an associated set of configuration parameters defined by the properties of the [migrationService](#).

Mobile service

Defines the service responsible for the delivery of content to mobile devices.

This service is invoked by the [reportService](#) service to deliver IBM Cognos content to wireless devices, such as mobile phones or other wireless devices.

The mobile service has an associated set of configuration parameters defined by the properties of the [mobileService](#) class.

Monitor service

Defines the service responsible for monitoring non-interactive tasks and managing [history](#) objects in the content store.

The monitoring service updates histories in the event of a catastrophic server failure. The monitoring service is also used to monitor tasks that are run in the background by the user for batch completion.

Unlike many of the other IBM Cognos Analytics services, the monitoring service is not responsible for running a task. Requests to run tasks that are sent to the monitoring service are forwarded to the service that can run the specified task. The monitoring service updates the appropriate [history](#) object based on the response from the service that is running the task.

The monitoring service is used by the [agentService](#) to run the steps that form an agent ([baseAgentDefinition](#)). The [jobService](#) uses the monitoring service to run the steps that form a job ([jobDefinition](#)).

The monitoring service has an associated set of configuration parameters defined by the properties of the [monitorService](#) class.

Planning services

Provides support for IBM Cognos Planning.

Reserved.

PowerPlay service

Defines the service responsible for the execution of PowerPlay reports ([basePowerPlay8Report](#) class).

The PowerPlay service has an associated set of configuration parameters defined by the properties of the [powerPlayService](#) class.

Presentation service

Transforms generic XML responses from another service into output format, such as HTML or PDF and provides display, navigation, and administration capabilities in Connection.

Query service

Defines the service responsible for executing dynamic query mode queries.

The query service has an associated set of configuration parameters defined by the properties of the [queryService](#) class.

Relational metadata service

Defines the service that extracts relational metadata from data sources.

Report data service

Manages the transfer of report data between IBM Cognos Analytics and applications that consume the data, such as IBM Cognos for Microsoft Office, IBM Cognos Analytics Mobile Reports, and IBM Cognos Mashup Service.

See the *IBM Cognos Mashup Service Developer Guide* for more information.

Report service

Defines the service responsible for the execution of reports ([baseReport](#) class).

The report service is dedicated to running interactive reports. For example, when a user runs a report from IBM Cognos Connection, the request is sent to the report service.

The report service has an associated set of configuration parameters defined by the properties of the `reportService` class. These parameters can be configured to optimize the report service for interactive use. This is the only difference between the report service and the batch report service (`batchReportService`).

This service may also make additional requests to the following IBM Cognos Analytics services when processing a request:

- `contentManagerService` service
- `deliveryService` service
- `eventManagementService` service
- `monitorService` service

Repository service

Supports connections to an external repository, such as IBM FileNet Content Manager, to archive versioned report output in the content store.

Standalone Cognos Access Manager

Defines the service responsible for providing authentication and authorization functionality to internal IBM Cognos components.

The standalone CAM service has an associated set of configuration parameters defined by the properties of the `saCAMService` class.

Note: Do not use the IBM Cognos Software Development Kit to start, stop, or update this service.

System service

Defines the service that can be used to obtain application-wide, IBM Cognos Analytics configuration parameters.

This service also provides methods that can be used to normalize and validate locale strings, as well as to map locale strings to locales supported by your application.

The system service has an associated set of configuration parameters defined by the properties of the `systemService` class.

IBM Cognos Software Development Kit Components

The IBM Cognos Software Development Kit consists of a number of individual software components and supporting documentation. This section describes the individual pieces that make up the package.

Cognos Software Development Kit

The IBM Cognos Software Development Kit consists of the following components:

Toolkits

Native libraries are provided, based on the published WSDL file that describes the BI Bus API, in the following programming environments:

- Java
- .NET Framework

Code Samples and Working Applications

The IBM Cognos Software Development Kit includes code samples and working applications that demonstrate various common operations developers might perform in customizing IBM Cognos functionality.

Documentation and Examples

The IBM Cognos Software Development Kit includes documentation and examples that provide information about

- using the URL API to run and view IBM Cognos objects (for example, reports, agents, and jobs), customize the launching of IBM Cognos applications such as IBM Cognos Connection or IBM Cognos Analytics - Reporting, and pass environment variables to external applications
- configuring advanced settings for IBM Cognos services
- creating applications with IBM Cognos Portal Services

Published Report Specification Schema

A report specification is an XML document that defines the structure and layout of a report. The IBM Cognos Software Development Kit includes methods to retrieve, validate, and save report specifications to the content store. You can use these methods as well as the XML schema files (*.xsd) that define the structure of a report specification and supporting documentation to

- programmatically apply the same changes to a number of reports
- automate report creation using a client application

IBM Cognos Framework Manager Development

IBM Cognos Framework Manager is a data modeling product that allows users to import metadata from one or more data sources and transform the metadata into a business-oriented model for creating reports.

Developers can use a collection of cross-platform web services, libraries, and programming interfaces provided with the IBM Cognos Software Development Kit to access the full functionality of Framework Manager. You can use the Framework Manager API to model metadata and publish packages without the use of the Framework Manager application.

For more information about developing modeling solutions, see the *IBM Cognos Framework Manager Developer Guide*.

IBM Cognos Mashup Service

The IBM Cognos Mashup Service gives you a simplified programmatic access to IBM Cognos content. This service exposes the application content built with IBM Cognos products (such as reports, analyses, and PowerPlay reports) as Web services, both SOAP and REST. This allows you to integrate IBM Cognos content into new client environments like mashups, BPM/BPEL workflow processes, desktop widgets, alternate visualizations like third party charting engines and rich Internet applications.

The IBM Cognos Mashup Service consists of the following components:

Code Samples and Working Applications

The Mashup Service includes code samples and working applications that demonstrate various common operations developers might perform in creating Mashup Service applications

Documentation and Examples

The Mashup Service includes documentation and examples that provide information about

- creating applications using the SOAP interface with the Java and C# programming languages

- creating applications using the REST interface in HTML and JavaScript
- the schemas used by the Mashup Service

Custom Authentication Provider API

Authentication providers give access to authentication sources. By configuring an authentication provider in IBM Cognos, you define an authentication namespace.

The authentication namespace configuration is part of the IBM Cognos installation. The namespace can be configured to use full authentication provider, or a trusted signon provider in conjunction with a full authentication provider.

IBM Cognos does not use authentication providers for authorization purposes. Policies and permissions are stored in Content Manager and are managed by IBM Cognos.

If your authentication provider of choice is not supported, or the basic functionality of any of the supported providers does not meet your requirements, you can use the IBM Cognos Software Development Kit to implement a custom authentication provider for IBM Cognos.

For more information about the IBM Cognos custom authentication solutions, see the *IBM Cognos Custom Authentication Provider Developer Guide*.

Script Player

Script Player is a command line utility that runs previously created action logs to create or modify models, or to publish packages for report authors. Script Player can run an entire action log or only the part that is required for a specific task.

An action log is an XML file that contains a series of action elements, which are grouped into transactions. When you use IBM Cognos Framework Manager to create a model, each time you click **OK** in the wizard, a new action is recorded in an action log.

You can run action logs using Script Player, or you can modify them and save them as new action logs. You can also create action logs manually, using the reference material in the *Framework Manager Developer Guide*.

Other Customization Capabilities

You can customize the look and feel of IBM Cognos Connection and other IBM Cognos applications, in addition to altering the layout and presentation of many of the IBM Cognos interfaces.

Extending IBM Cognos Into Your Business

The IBM Cognos Software Development Kit provides a single application programming interface (API) that delivers complete openness and the ability to access broad analytics functionality. The IBM Cognos Software Development Kit, along with other components and features of IBM Cognos, supports a comprehensive set of tools to integrate, customize, and extend IBM Cognos functionality in your organization.

Integration, Customization, and Extensibility

Virtually everything you can do with the IBM Cognos graphical user interface, you can do using the appropriate API, XML file, or command line utility.

IBM Cognos components are built on the same BI Bus API that the IBM Cognos Software Development Kit is based on. This foundation provides developers with comprehensive access to

- a broad range of IBM Cognos platform functionality including reporting, analysis, administration, and security features.
- all of the quality, performance, and scalability benefits that are part of the service-oriented architecture of IBM Cognos.

- multi-language support.

You can choose to automate only a specific task or you can program the entire process, from modeling through to reporting, scheduling, and distribution.

You can use the IBM Cognos Software Development Kit to create custom reports, manage deployment, and integrate security and portal functionality to suit your individual needs, locale, and existing software infrastructure. For example, you can:

Manage Security

You can use the IBM Cognos Software Development Kit to manage your security. This functionality includes:

- authenticating users using supported or custom authentication providers
- managing authorization information in the content store
- managing capabilities for users, groups, and roles by restricting access to predefined secured functions and features

Manage and Querying Objects and Properties

You can use the IBM Cognos Software Development Kit to query objects and properties in the content store. You can also manage and manipulate content in the content store.

Execute Runnable Objects

Execute runnable objects such as reports, agents, queries, or jobs using one of the supported programming languages or through a URL interface.

Perform Administration Functions

You can use the IBM Cognos Software Development Kit to perform all your administration functions such as

- managing dispatchers and services
- deploying content and performing content maintenance activities
- monitoring system information for performance or tuning purposes
- performing or automating administrative operations

Perform Customized Operations

You can use the IBM Cognos Software Development Kit to perform customized operations on reports by programmatically manipulating report specifications. You can also use the IBM Cognos Software Development Kit to automate report specification creation or batch editing.

Create and Publish Framework Manager Models

You can use the `updateMetadata()` method in the IBM Cognos Software Development Kit to modify or publish a model programmatically. The Script Player can also be used to modify or publish models to the content store.

For more information about the Script Player, see the *IBM Cognos Framework Manager Developer Guide*.

Integrate with Other Applications or Web Portals

Integrate IBM Cognos content into other applications or Web portals in order to interact directly with IBM Cognos information.

Use the URL API

You can use the URL API to launch IBM Cognos applications without the use of IBM Cognos Connection. The URL API can also be used to customize how objects are launched in IBM Cognos Viewer.

Developer Resources

This section describes IBM Cognos resources available to developers.

Product Documentation

The IBM Cognos Software Development Kit installation contains a full suite of developer documentation and sample applications.

IBM Cognos Software Development Kit Developer Guide

The *IBM Cognos Software Development Kit Developer Guide* provides essential conceptual, reference, and procedural information and examples for developers working in a broad variety of customization tasks, including:

- Provides introductory information about the IBM Cognos Software Development Kit and how SOAP and WSDL are used with it. It explains how to use the toolkits in Java and the .NET Framework environment. Task-oriented examples provide assistance to developers new to the IBM Cognos Software Development Kit to help them build applications using the toolkits.
- Performing various tasks with the IBM Cognos Software Development Kit using the BI Bus API. This includes information on managing authentication and authorization, understanding how the content store and search paths work, and general information about customizing IBM Cognos.
- Using of the URL API to run and view IBM Cognos objects, such as reports, agents, and jobs.
- Customizing the launching of IBM Cognos applications such as IBM Cognos Connection or IBM Cognos Analytics - Reporting, passing environment variables to external applications, and creating custom applications with IBM Cognos Portal Services.
- Customizing the creation, maintenance, and automated tasks that act on report objects.
- Working with lineage and selection context information to determine how objects in a report or package are derived through transformations from other objects, back to source data.
- General reference information applicable to all types of development or customization activities, including:
 - Search path syntax
 - Advanced configuration settings
 - Initial security settings
 - Metadata schema reference
- What's New topics that provide feature-specific information about the changes to the IBM Cognos Software Development Kit for the current version of the product, in addition to a history of changes from previous releases.

IBM Cognos Framework Manager Developer Guide

The *IBM Cognos Framework Manager Developer Guide* is for developers interested in using the collection of cross-platform Web services, libraries, and programming interfaces provided with the IBM Cognos Software Development Kit, to access the full functionality of Framework Manager. This guide includes both task-oriented and reference information, to help you implement custom solutions for metadata modeling.

IBM Cognos Mashup Service Developer Guide

This document describes in detail how you can develop applications using the IBM Cognos Mashup Service, using the C# and Java programming languages for the SOAP interface, and using HTML and JavaScript for the REST interface. It also contains detailed reference information about the SOAP and REST interfaces to the Mashup Service.

Custom Authentication Provider Developer Guide

This document provides information to assist developers in creating their own custom authentication provider, including a trusted signon provider, if what IBM Cognos supports does not satisfy your business needs.

Chapter 3. Getting started

Virtually everything you can do with the IBM Cognos graphical user interface, you can do using the appropriate API, XML file, or command line utility. The IBM Cognos Software Development Kit provides a platform-independent automation interface for working with IBM Cognos services and components. You can use the Software Development Kit to create custom reports, manage deployment, and integrate security and portal functionality to suit your individual needs, locale, and existing software infrastructure. The Software Development Kit uses a collection of cross-platform Web services, libraries, and programming interfaces.

IBM Cognos integrates easily into existing software or portals by providing the BI Bus API, which accommodates many programming languages. You can choose to automate only a specific task or you can program the entire process, from modeling through to reporting, scheduling, and distribution.

The IBM Cognos Web-based architecture is built around the following industry standards:

- Web Services Description Language (WSDL)
- Simple Object Access Protocol (SOAP), with XML and HTTP

WSDL is an XML-based language used to define the services provided by a server. IBM Cognos software is designed to take advantage of this Web service technology which, like SOAP and XML, is platform-independent and fully scalable.

Although .NET Framework and Java™ interfaces are provided as a convenience for developers, the main value-added component of the Software Development Kit is the IBM Cognos WSDL file, Cognos.wsdl. This file defines the services provided and the methods supported by each service.

Think of the WSDL file as the service contract between a server and its clients. Client applications send XML-based SOAP messages to IBM Cognos Analytics servers. These servers interpret the requests, which are formatted using SOAP, HTTP, and MIME standards. The servers then retrieve, save, and deliver the specified reports and queries to the clients.

SOAP messages are encoded in XML and then transmitted using HTTP. Because these technologies work on all IBM Cognos platforms, components can be distributed among Windows, UNIX, and Linux® servers to optimize query performance.

The SOAP standard and the IBM Cognos WSDL file specify how to encode an XML-structured message. If a program on one computer is set up to call a program on another computer, SOAP specifies the format and the content of the communication. For example, when a Software Development Kit user requests a report from the content store, the XML data is retrieved, enclosed in a SOAP envelope, and then returned.

The interfaces provided with the Software Development Kit eliminate the need to construct actual SOAP messages.

When you work with the BI Bus API, you are working with the central communications channel between IBM Cognos services. The Software Development Kit turns commands into SOAP messages that the BI Bus API then delivers to the appropriate services. The BI Bus API wraps each XML-formatted request or response in a SOAP envelope and transmits it using HTTP.

The BI Bus API is based on SOAP 1.1. For more information about setting up the Software Development Kit for the .NET Framework and the Java programming language, see [“The .NET Framework toolkit”](#) on page 23 and [“The Java toolkit”](#) on page 20.

By following the reporting examples (see [“Reporting examples”](#) on page 24) in this document, you will learn to address the BI Bus API and the command line utilities available to automate IBM Cognos tasks. These examples cover just a few of the methods that can be used to automate any functionality of IBM Cognos software. For more information, see the IBM Cognos Software Development Kit *Developer Guide* and the Framework Manager *Developer Guide*.

Before you begin programming using the Software Development Kit, we recommend that you familiarize yourself with the IBM Cognos user interface.

BI Bus API

You can use the BI Bus API to write programs in the Java and C# programming languages to automate IBM Cognos actions. The following table illustrates some of the uses of the BI Bus API.

Table 1. BI Bus API uses

Use	Reference	User interface equivalent
To modify an unpublished model	Information about the <code>updateMetadata</code> method found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Framework Manager
To retrieve the query items available in the published package	Information about the <code>runSpecification</code> method found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Connection
To grant capabilities	Information about applying security found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Connection, Administration
To administer and implement security	Information about logging on and logging off and applying security found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Connection, Administration
To manipulate objects in the content store	Information about managing Software Development Kit content found in the Software Development Kit <i>Developer Guide</i>	IBM Cognos Connection, Query Studio, Reporting, and Framework Manager
To administer servers	Information about dispatchers and services found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Connection, Administration
To modify a report	Information about using report specifications found in the IBM Cognos Software Development Kit <i>Developer Guide</i>	IBM Cognos Analytics - Reporting

URL interface

You can use the URL interface to automate IBM Cognos actions by passing commands and parameters in a URL. The following table illustrates some of the uses of the URL interface.

Use	Reference	User interface equivalent
To run, view, and edit reports through a hyperlink in a Web page	Information about viewing, editing, and running reports using URLs found in the Software Development Kit <i>Developer Guide</i>	Report Viewer, Query Studio, and Reporting
To modify the functionality of IBM Cognos	Information about hiding user interface elements and restricting content browsing found in the <i>Administration and Security Guide</i>	IBM Cognos Connection, Reporting, Query Studio

Script Player

Script Player is a command line utility that runs previously created action logs to create or modify models, or to publish packages for report authors. Script Player can run an entire action log or only the part that is required for a specific task.

An action log is an XML file that contains a series of action elements, which are grouped into transactions. When you use Framework Manager to create a model, each time you click **OK** in the wizard, a new action is recorded in an action log.

You can run action logs using Script Player, or you can modify them and save them as new action logs. You can also create action logs manually, using the reference material in the Framework Manager *Developer Guide*.

The following table illustrates a use of Script Player.

Use	Reference	User interface equivalent
To modify a model or republish a model to UNIX or Windows	Information about Script Player in the Framework Manager <i>Developer Guide</i>	Framework Manager

IBM Cognos toolkits

Web services applications access the IBM Cognos WSDL file through the Web server gateway, using the URL `webservername/ibmcognos/bi/v1/disp?wsdl`. You can also open the WSDL file in an XML editor to see how its contents compare to what you see in the Software Development Kit reference section in the Software Development Kit *Developer Guide*. The Cognos.wsdl file can be found in `installation_location/webapps/p2pd/WEB-INF/classes`.

Unless your IDE can use the WSDL file directly, you must install the appropriate toolkit before you can work with the BI Bus API. The Software Development Kit installation includes additional samples that are discussed in the Software Development Kit *Developer Guide*. For information about code samples and language-specific coding practices, see the Software Development Kit *Developer Guide*.

There are some differences in the way toolkits interact with BI Bus API objects and integrate with IBM Cognos software. The Software Development Kit *Developer Guide* describes all syntax related to the Software Development Kit. The Software Development Kit provides different libraries for the .NET Framework and the Java programming language.

The Java toolkit

The IBM Cognos Software Development Kit includes a Java toolkit that is based on Apache Axis. You can use the Java toolkit to interact with the Software Development Kit. For more information about Apache Axis, see the Apache Web site.

The IBM Cognos Software Development Kit CD includes the Javadocs documentation, which can be found in *installation_location/sdk/java/doc/cognos/javadoc*. Javadocs provides language-specific examples of accessor methods so that you can retrieve and set properties for the BI Bus API objects.

To build and run Software Development Kit clients using the IBM Cognos Software Development Kit Java toolkit, you need a Java Development Kit.

Java naming conventions

In the Java programming language, by convention, the names of classes are modified so that the first character in the name is uppercase.

The name of the deprecated `import` method conflicts with a Java keyword. Therefore, the method name in Java is prefixed with an underscore (`_`) so that the name becomes `_import`.

Java Archive files (.jar)

The following files, already installed in *installation_location/sdk/java/lib*, are the minimum .jar files required when building your own application:

- activation.jar
- axis.jar
- axisCognosClient.jar
- commons-discovery-0.2.jar
- commons-logging-1.1.jar
- commons-logging-adapters-1.1.jar
- commons-logging-api-1.1.jar
- jaxen-1.1.1.jar
- jaxrpc.jar
- mail.jar
- saaj.jar
- wsdl4j-1.5.1.jar

You must also use the following .jar files if you want to run the IBM Cognos Software Development Kit sample applications:

- dom4j-1.6.1.jar

Note:

- axis.jar is a modified version of Apache axis 1.4. Customers who need to use a higher version of axis that may conflict with axis.jar can use cognos-axis.jar (instead of axis.jar) and cognosClient.jar (instead of axisCognosClient.jar) in the classpath for the Software Development Kit application.
- If you have an application that requires a different version of a .jar file that ships with the IBM Cognos Software Development Kit, store your .jar file in a different location. Do not overwrite the .jar files provided with the Software Development Kit.
- The source code for the axisCognosClient.jar classes can be found in *installation_location/sdk/java/src/axisCognosClientSrc.jar*.

Using the Java samples

The IBM Cognos Software Development Kit includes Java program samples that show you some of the types of applications you can design. These samples include source files so that you can test changes to the sample code, and batch files or shell scripts for compiling and running the samples.

Comments in the source files describe the main purpose of each sample, including a summary of which BI Bus API Software Development Kit methods are used. The batch files and shell scripts contain instructions that you must follow before you run them.

Each subdirectory in `installation_location/sdk/java` contains the following files.

File	Description
<code>build.bat</code>	Builds the individual sample on Windows operating systems
<code>build.sh</code>	Builds the individual sample on UNIX or Linux operating systems
<code>run.bat</code>	Runs the individual sample on Windows operating systems
<code>run.sh</code>	Runs the individual sample on UNIX or Linux operating systems

In addition, the `installation_location/sdk/java` directory contains the following files.

File	Description
<code>build-samples.bat</code>	Builds all the Java samples on Windows operating systems
<code>build-samples.sh</code>	Builds all the Java samples on UNIX or Linux operating systems

Before you modify any of the Java samples, familiarize yourself with basic Java programming techniques.

Before you use these Java samples, check to see if you have anonymous access enabled. Although the samples will work with anonymous access, security features will not be demonstrated. To use the security features of the samples, ensure that you have a secured NTLM, LDAP, or other namespace, and that you disable anonymous access.

Because some of the Java samples issue output to the command console, ensure that the console is visible when you run the samples.

Java samples setup for Windows operating systems

Use this procedure to setup the Java samples on Windows operating systems.

Procedure

1. Install a Java Development Kit (JDK). To determine the supported Java. versions, see [IBM Cognos Analytics 11.0.0 Supported Software Environments](http://www-01.ibm.com/support/docview.wss?uid=swg27047186) (<http://www-01.ibm.com/support/docview.wss?uid=swg27047186>). Select your product and product version and create a report for related software. Choose Development Tools under Supported software.

The report that is generated lists the supported Java versions.

2. Ensure that your PATH environment variable includes the location where the JDK is installed.
3. Edit *installation_location/sdk/java/Common/CRNConnect.java* by locating the line

```
public static String CM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
```

and replacing localhost:9300 with the name and, if necessary, port number of your IBM Cognos dispatcher.

4. If you want to compile all the samples, edit *installation_location/sdk/java/build-samples.bat* and change the lines

```
set JAVA_HOME=c:/jdk1.5
set CRN_HOME=../..
```

so that they point to the locations where the JDK and the IBM Cognos Analytics server are installed, respectively.

5. If you want to compile individual samples, edit *installation_location/sdk/java/sample_name/build.bat* and change the lines

```
set JAVA_HOME=c:/jdk1.5
set CRN_HOME=../..
```

so that they point to the locations where the JDK and the IBM Cognos Analytics server are installed, respectively.

6. Compile the Java samples by running build-samples.bat (to compile all samples) or build.bat (to compile an individual sample).

Java setup for Linux and UNIX operating systems

Use this procedure to setup the Java samples on Linux and UNIX operating systems.

Procedure

1. Install a Java Development Kit (JDK). To determine the supported Java versions, see [IBM Cognos Analytics 11.0.0 Supported Software Environments](http://www-01.ibm.com/support/docview.wss?uid=swg27047186) (<http://www-01.ibm.com/support/docview.wss?uid=swg27047186>). Select your product and product version and create a report for related software. Choose Development Tools under Supported software.

The report that is generated lists the supported Java versions.

2. Set the JAVA_HOME environment variable to point to the location where the JDK is installed.
3. Edit *installation_location/sdk/java/Common/CRNConnect.java* by locating the line

```
public static String CM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
```

and replacing localhost:9300 with the name and, if necessary, port number of your IBM Cognos dispatcher.

4. If you want to compile all the samples, edit *installation_location/sdk/java/build-samples.sh* and change the lines

```
CRN_HOME=/usr/cognos/xxx
JAVA_HOME=/c/j2sdk1.5
```

so that they point to the locations where JDK and the IBM Cognos Analytics server and the JDK are installed, respectively.

5. If you want to compile individual samples, edit *installation_location/sdk/java/sample_name/build.sh* and change the lines

```
CRN_HOME=/usr/cognos/xxx
JAVA_HOME=/c/j2sdk1.5
```

so that they point to the locations where JDK and the IBM Cognos Analytics server and the JDK are installed, respectively.

6. Compile the Java samples by running `build-samples.sh` (to compile all samples) or `build.sh` (to compile an individual sample).

Distributing Java applications

You can run IBM Cognos Software Development Kit applications written in Java on a system that does not have the IBM Cognos server installed.

Procedure

1. Copy the class files and any other supporting files you have created for the application to the target system.
2. Copy the .jar files from `installationlocation/sdk/java/lib` to the target system.
3. Install a JRE on the target system, or copy the contents of `installationlocation\bin\jre` to the target system.
4. Ensure that the classpath of the application points to the .jar files you copied to the target system.

The .NET Framework toolkit

You can use any Microsoft .NET language and the Microsoft .NET Framework version 4.0 to interact with the Software Development Kit.

The `cognosdotnet_10_2.dll` and `cognosdotnetassembly_10_2.dll` assemblies are provided for use with the .NET Framework, and are located in `installation_location/sdk`

Using Microsoft Visual Studio, version 2010 or later, you can generate the BI Bus API objects by referencing these assemblies.

These assemblies have strong signed names, so they can be shared among applications and deployed in the Global Assembly Cache.

Note: Although Visual Studio directly supports WSDL through the use of Web references, limitations of the WSDL support in Visual Studio requires the use of these assemblies when developing applications using the IBM Cognos Software Development Kit.

Creating a Visual Studio project

Use this procedure to create a Microsoft Visual Studio project.

Procedure

1. Create a new project in Visual Studio.
2. From the **Project** menu, click **Add Reference**.
3. Add references to `cognosdotnet_10_2.dll` and `cognosdotnetassembly_10_2.dll`.
4. Using the Object browser, expand `cognosdotnet_10_2.dll` to view the classes in the BI Bus API.
5. Create an object to connect to an IBM Cognos Analytics server.
6. Use the BI Bus API methods that are available through the object to work with IBM Cognos.

C# naming conventions

When you use C#, class names that conflict with C# keywords such as `namespace` are generated with modified names such as, in this case, `@namespace`.

Using the C# samples

The C# sample files are installed in various subdirectories in the location `installation_location/sdk/csharp`. There is one subdirectory for each sample. All executable files are located in `installation_location/sdk/csharp/bin`.

The folder *installation_location/sdk/csharp/SamplesCommon* contains a solution that provides several classes used by other samples.

Each of the subdirectories contain the following files:

- AssemblyInfo.cs
- *sample_name.cs*
- *sample_name.csproj*
- *sample_name.sln*
- build.bat
- *csharp_sample_name_Explain.html*

Some samples may have additional files.

You must reference the *cognosdotnet_10_2.dll* and *cognosdotnetassembly_10_2.dll* files in your Visual Studio project. There are two versions of these files:

- .NET Framework 4.0 versions located in *installation_location/sdk*
- .NET Framework 3.5 versions located in *installation_location/sdk/NET3.5*

Note:

- The solution (.sln) and project (.csproj) files require Visual Studio, version 2010 or later.
- AllSamples.sln in *installation_location/sdk/csharp* includes all the sample projects. We recommend that you use this solution when you want to build all the C# samples at the same time.

In each sample, the build.bat script included with the sample code shows one way of building the application using the Visual Studio compiler. For more information about using build.bat, see the comments in the file.

Read the *csharp_sample_name_Explain.html* file before you run any of the C# samples. These files provide instructions about running the sample programs, and contain information about associated files. They are located in *installation_location/sdk/csharp/sample_name*.

In addition, the source files contain comments to help you follow the sequence, and understand what each sample does. We recommend that you become familiar with .NET programming techniques before you modify any of the samples.

The C# samples include executable programs, located in *installation_location/sdk/csharp/bin*. If you receive an automation error when you launch an .exe program, you must recompile the program for your environment.

Distributing .NET applications

You can run IBM Cognos Software Development Kit applications written in a .NET language on a system that does not have the IBM Cognos Analytics server installed.

Procedure

1. Copy the .exe or .dll files and any other supporting files you have created for the application to the target system.
2. Copy *cognosdotnetassembly_10_2.dll* and *cognosdotnet_10_2.dll* from *installation_location/sdk* to the target system.
3. Install the .NET framework on the target system.

Reporting examples

When you log on, run a report, or delete a report from the content store, you are using the BI Bus API to pass data along the IBM Cognos Analytics communications channel.

To learn more about the IBM Cognos Software Development Kit, we recommend that you explore the examples. These examples demonstrate some of the most common methods for reporting available in the BI Bus API. For more information, see the chapter on methods in the Software Development Kit *Developer Guide*. For information about BI Bus API classes, see the chapter on classes in the Software Development Kit *Developer Guide*.

After you understand the examples described in this chapter, you can move on to the more complex and detailed examples in the Software Development Kit *Developer Guide*. IBM Cognos contains many other examples, which show examples of business applications and functionality. Samples in the Software Development Kit *Developer Guide* also use JSP.

The code snippets provided in this chapter are available as code samples in *installation_location/sdk/java* for the Java programming language, and *installation_location/sdk/csharp* for the C# programming language. All sample reports are available in the package that is published when you set up the IBM Cognos Analytics samples. For more information, see the IBM Cognos Analytics *Installation and Configuration Guide*.

Example - connect to an IBM Cognos Analytics server

To connect to the IBM Cognos Analytics server, you connect to a service and provide URL information. If you are connecting through IBM Cognos Connection, you launch your Web browser and supply the gateway URL provided by your administrator, which is of the form

```
http[s]://host:port/ibmcognos
```

You cannot use this method of access with your Software Development Kit applications because the gateway is configured by default to use the following URI:

```
http://localhost:9300/p2pd/servlet/dispatch/ext
```

This causes it to block requests from Software Development Kit applications.

Depending on your network configuration, set up your Software Development Kit application to do one of the following:

- connect directly to the dispatcher using the internal Dispatcher URI (`http://localhost:9300/p2pd/servlet/dispatch`)
- connect to an additional, dedicated gateway that is configured to connect to the dispatcher using the internal Dispatcher URI. You must configure appropriate security for this gateway. This method is useful when the Software Development Kit application is outside a network firewall.

Note: Do not change your main gateway to use the internal Dispatcher URI. Doing so reduces the security of the IBM Cognos Analytics portal and studios.

The code used for connection varies with the service being connected to. The chapter on services in the Software Development Kit *Developer Guide* has code snippets illustrating how to connect to each service. The code snippets that follow show you how to connect to the agent service in Java and C#.

Java code

The following Java code snippet demonstrates how to connect to the agent service in the IBM Cognos Analytics Server.

```
private AgentService_ServiceLocator agentServiceLocator = null;
...
private AgentService_PortType agentService = null;
...
public static String CM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
...
    // Create the service locators
    agentServiceLocator = new AgentService_ServiceLocator();
...
    java.net.URL serverURL = new java.net.URL(CM_URL);
```

```
//acquire references to the services
agentService = agentServiceLocator.getagentService(serverURL);
```

C# code

The following C# code snippet demonstrates how to connect to the agent service in the IBM Cognos Analytics Server.

```
private agentService cBIAS = null;
...
    cBIUrl = "http://localhost:9300/p2pd/servlet/dispatch";
    cBICMS.Url = cBIUrl;
...
    if (cBIAS == null)
    {
        cBIAS = new agentService();
        cBIAS.Url = cBICMS.Url;
    }
    if (cBIAS.biBusHeaderValue == null)
    {
        cBIAS.biBusHeaderValue = cBICMS.biBusHeaderValue;
    }
}
```

Explanation

When you establish a connection, you create an object to handle communication in IBM Cognos Analytics Server Cognos. This object manages the set of requests and responses on behalf of the client application.

Steps when writing your own programs

1. In Java, create a service locator for the class.

In C#, create a variable and set it to an instance of a service class.

2. Set the URL for the connection.

The URL provided by the user is passed to the service class variable to establish a connection to the IBM Cognos Analytics gateway.

Example - log on to an IBM Cognos Analytics server

IBM Cognos Analytics supports authenticated and anonymous user access. To use IBM Cognos software, an application using the BI Bus API must successfully connect to the IBM Cognos Analytics Server and log on as a named user or an anonymous user. An application using the BI Bus API connects to IBM Cognos using an instance of a Web service class.

IBM Cognos Analytics Server has a flexible security architecture that supports both distributed and delegated security. It offers multiple levels of security from a group, role, or user perspective. For information about administering security, see the *IBM Cognos Analytics Administration and Security Guide*.

IBM Cognos Analytics Server also offers simultaneous authentication. You can use your existing security infrastructure to authenticate and authorize users against multiple providers, such as LDAP, Active Directory, and NTLM.

If you configured your IBM Cognos Analytics Server with multiple namespaces, you can log on to all available namespaces. However, you must log on only one user at a time to a namespace when using the BI Bus API methods. This limit is necessary because each logon is part of the same transaction and shares the same authentication passport. A program cannot log on to the same namespace multiple times.

Before using either the `logon` or `logoff` methods, you must connect to the IBM Cognos Analytics server (see [“Example - connect to an IBM Cognos Analytics server”](#) on page 25).

To log on to an IBM Cognos Analytics Server as a valid user, you must have the proper permissions and capabilities to use the server. For more information, see the *Administration and Security Guide*. Otherwise, you can log on anonymously, but your access permissions may be limited by your system administrator.

Method

Use the `logon(credentials, roles)` BI Bus API method to authenticate a user in the IBM Cognos Analytics Server

The `logon` method informs the BI server what you want to do (log on), who you are (credentials), and what you are allowed to do (roles). This method requires that you provide credentials although the roles parameter may be empty. If you do not specify a role, you are logged on with the combined access permissions of all the roles of which you are a member.

Tip: The `logon` method requires a string array for roles. Even if you do not pass IBM Cognos Analytics Server values, you must still include the roles parameter in your `logon` method call.

Use the `logout()` method to log off from the IBM Cognos Analytics Server.

Java code

To see this code in context, view the following sample:

`installation_location/sdk/java/Security/Logon.java`

The following Java code snippet demonstrates how to log on to IBM Cognos.

```
StringBuffer credentialXML = new StringBuffer();

credentialXML.append("<credential>");

credentialXML.append("<namespace>");
credentialXML.append(namespace);
credentialXML.append("</namespace>");

credentialXML.append("<username>");
credentialXML.append(uid);
credentialXML.append("</username>");

credentialXML.append("<password>");
credentialXML.append(pwd);
credentialXML.append("</password>");

credentialXML.append("</credential>");

String encodedCredentials = credentialXML.toString();
credentialString = encodedCredentials;

connection.getCMSservice().logon(new XmlEncodedXML(encodedCredentials),
    new SearchPathSingleObject[] { });
```

C# code

To see this code in context, view the following sample:

`installation_location/sdk/csharp/SamplesCommon/SamplesLogon.cs`

The following C# code snippet demonstrates how to log on to the IBM Cognos Analytics Server.

```
System.Text.StringBuilder credentialXML =
    new System.Text.StringBuilder("<credential>" );
credentialXML.AppendFormat( "<namespace>{0}</namespace>", userNamespace );
credentialXML.AppendFormat( "<username>{0}</username>", userName );
credentialXML.AppendFormat( "<password>{0}</password>", userPassword );
credentialXML.Append( "</credential>" );

//The csharp toolkit encodes the credentials
string encodedCredentials = credentialXML.ToString ();
xmlEncodedXML xmlEncodedCredentials = new xmlEncodedXML();
```

```
xmlEncodedCredentials.Value = encodedCredentials;
searchPathSingleObject[] emptyRoleSearchPathList = new searchPathSingleObject[0];
cBICMS.Logon(xmlEncodedCredentials, null);
```

Explanation

In this example, the `logon` method is called, and the encoded credentials are passed to the service. The credentials are the user ID, password, and namespace ID. The credential information is passed as an encoded XML string to the `logon` method.

Steps when writing your own programs

1. Build the credentials XML string.

Tip: In Java, XML reserved character escaping is done for you.

2. Pass the credentials XML string to the `logon` method to initialize the IBM Cognos session.

In the example code, no roles are specified. In the Java example, `null` is passed.

3. Use the `logout` method to terminate the user session and remove the passport from the CAM object. This, in turn, ends any namespace sessions a user opened.

This method works regardless of whether you have a secured namespace or have anonymous access enabled.

Example - run a report

To run a report, you must specify the location of the report in a search path. The search path tells the report server where to find the report in the content store. For information about managing content, see the Software Development Kit *Developer Guide*.

Before running a report, you must do the following:

- connect to an IBM Cognos Analytics server (see [“Example - connect to an IBM Cognos Analytics server” on page 25](#))
- if authentication is required, log on to an IBM Cognos Analytics server (see [“Example - log on to an IBM Cognos Analytics server” on page 26](#))

Methods

You can use the following BI Bus API methods to run reports or report specifications:

- `run(objectPath, parameterValues, options)`
- `runAt(startTime, objectPath, parameterValues, options)`
- `runSpecification(specification, parameterValues, options)`

These examples use the `run(objectPath, parameterValues, options)` method.

See the Methods chapter in the Software Development Kit *Developer Guide* for the permissions and capabilities required for each method.

Java code

To see this code in context, view the following samples:

`installation_location/sdk/java/ExecReports/RunReport.java`

`installation_location/sdk/java/ReportParams/ReportParameters.java` for a report with prompts

The following Java code snippets demonstrate how to run a report.

Steps for a Java program

1. Create an array of ParameterValue objects and get the parameters from the report.

- If the user is going to run a report without prompts, the array will be empty.

```
ParameterValue emptyParameterValues[] = new ParameterValue[] {};
ParameterValue reportParameterValues[] = null;

ReportParameters repParms = new ReportParameters();
BaseParameter[] parametersInReportSpec = new Parameter[] {};

try
{
    parametersInReportSpec =
        repParms.getReportParameters(report, connect);
}
}
```

- If the user is going to run a report with prompts, populate the array.

```
public static ParameterValue[] setReportParameters(BaseParameter[] prm)
{
    try
    {
        int numberOfParameters = 0;

        // Select the parameter values for the specified report.
        if (prm.length > 0)
        {
            numberOfParameters = prm.length;

            ParameterValue[] params =
                new ParameterValue[numberOfParameters];

            // Repeat for each parameter.
            for (int i = 0; i < prm.length; i++)
            {
                // Prompt the user to type a value for the parameter.
                // If the value is DateTime, the format must be in the ISO 8601
                // format. For example, a date and time of 2001-05-31T14:39:25.035Z
                // represents the thirty-first day of May in the year 2001. The time,
                // measured in Coordinated Universal Time (UTC) as indicated by the Z,
                // is 14 hours, 39 minutes, 25 seconds, and 35 milliseconds.
                String modelFilterItem = ((Parameter)prm[i]).getModelFilterItem();
                String item =
                    modelFilterItem.substring(
                        modelFilterItem.lastIndexOf("[",
                            modelFilterItem.lastIndexOf("]") + 1);
                String inputValue =
                    JOptionPane.showInputDialog(
                        "Please input a value for "
                            + item
                            + " of datatype ["
                            + prm[i].getType().getValue()
                            + "]");

                SimpleParmValueItem item1 = new SimpleParmValueItem();
                item1.setUse(inputValue);

                // Create a new array to contains the values for the parameter.
                ParmValueItem pvi[] = new ParmValueItem[1];
                pvi[0] = item1;

                // Assign the values to the parameter.
                params[i] = new ParameterValue();
                params[i].setName(prm[i].getName());
                params[i].setValue(pvi);
            }
        }
        return params;
    }
}
```

2. Set the options.

```
public Option[] getDefaultRunOptions(int reportType, Option[] options)
{
    ...
    Option execReportRunOptions[] = new Option[6];
}
```

```

RunOptionBoolean saveOutputRunOption = new RunOptionBoolean();
RunOptionStringArray outputFormat = new RunOptionStringArray();
RunOptionBoolean promptFlag = new RunOptionBoolean();
AsynchOptionBoolean includePrimaryRequest = new AsynchOptionBoolean();

//Set the option for saving the output to false
saveOutputRunOption.setName(RunOptionEnum.saveOutput);
saveOutputRunOption.setValue(false);

//What format do we want the report in: PDF, HTML, or XML?
outputFormat.setName(RunOptionEnum.outputFormat);
String[] reportFormat = null;
reportFormat = setFormatByType(reportType);
outputFormat.setValue(reportFormat);

//Set the report not to prompt as we pass the parameter (if any)
promptFlag.setName(RunOptionEnum.prompt);
promptFlag.setValue(false);

//Set the option to always have the primaryRequest in the response
includePrimaryRequest.setName(
    AsynchOptionEnum.alwaysIncludePrimaryRequest);
includePrimaryRequest.setValue(true);
...
execReportRunOptions[0] = saveOutputRunOption;
execReportRunOptions[1] = outputFormat;
execReportRunOptions[2] = promptFlag;
execReportRunOptions[3] = includePrimaryRequest;
...
return execReportRunOptions;

```

3. Run the report and get the output.

```

public String executeReport(
    BaseClassWrapper report,
    CRNConnect connect,
    int reportType,
    ParameterValue paramValueArray[],
    boolean doBurst,
    Option runOptions[])
    throws java.rmi.RemoteException
{
    Option execReportRunOptions[];
    AsynchReply rsr = null;

    //check for advanced routing server group
    String serverGroup = ((AuthoredReport)report.getBaseClassObject())
        .getRoutingServerGroup().getValue();
    if(serverGroup == null) {
        serverGroup = "";
    }

    String rSearchPath =
        report.getBaseClassObject().getSearchPath().getValue();
    String ERR_MESG =
        "run() failed to return a valid report in this format";

    //Prepare the runOptions
    if(doBurst)
    {
        execReportRunOptions = getBurstRunOptions(report, reportType, runOptions);
    }
    else
    {
        execReportRunOptions = getDefaultRunOptions(reportType, runOptions);
    }

    //Call run()
    rsr =
        connect.getReportService(true, serverGroup).run(
            new SearchPathSingleObject(rSearchPath),
            paramValueArray,
            execReportRunOptions);

    // If response is not immediately complete, call wait until complete
    if (!rsr.getStatus().equals(AsynchReplyStatusEnum.complete)&&!rsr.getStatus()
        .equals(AsynchReplyStatusEnum.conversationComplete))
    {
        while (!rsr.getStatus().equals(AsynchReplyStatusEnum.complete))

```



```

        &&!rsr.getStatus()
        .equals(AsynchReplyStatusEnum.conversationComplete))
    {
        //before calling wait, double check that it is okay
        if (!hasSecondaryRequest(rsr, "wait"))
        {
            return ERR_MESG;
        }
        rsr =
            connect.getReportService().wait(
                rsr.getPrimaryRequest(),
                new ParameterValue[] {},
                new Option[] {});
    }

    //After calling wait() it is necessary to check to make sure
    //the output is ready before retrieving it
    if (outputIsReady(rsr))
    {
        rsr =
            connect.getReportService().getOutput(
                rsr.getPrimaryRequest(),
                new ParameterValue[] {},
                new Option[] {});
    }
    else
    {
        return ERR_MESG;
    }
}
}

```

C# code

To see the code in context, view the following samples:

installation_location/sdk/csharp/ExecuteReport/ExecuteReport.cs

installation_location/sdk/csharp/ReportParameters/ReportParameters.cs for a report with prompts

The following C# code snippets demonstrate how to run a report.

Steps for a C# program

1. Create an array of ParameterValue objects, and get the report parameters array.

- If the user is going to run a report without prompts, the array will be empty.

```

parameterValue[] arrParm = new parameterValue[] {};
...
asynchReply gpReply = cBIRS.getParameters( cmReportPath, new parameterValue[]
    {}, new runOption[]{} );

```

- If the user is going to run a report with prompts, populate the array of ParameterValue objects.

```

foreach( parameter p in prm )
{
    // Prompt the user to type a value for the parameter.
    // If the value is DateTime, the format must be in the ISO 8601
    // format. For example, a date and time of 2001-05-31T14:39:25.035Z
    // represents the thirty-first day of May in the year 2001. The time,
    // measured in Coordinated Universal Time (UTC) as indicated by the Z,
    // is 14 hours, 39 minutes, 25 seconds, and 35 milliseconds.
    string desc = "Enter a value for prompt '" + p.name + "'
        (of type " + p.type.ToString() + "):";
    string val = inputter.getInput( null, desc, p.type.ToString() );

    simpleParmValueItem item = new simpleParmValueItem();
    item.use = val;
    parmValueItem[] pvi = new parmValueItem[1];
    pvi[0] = item;

    parms[pidx] = new parameterValue();
    parms[pidx].name = p.name;
    parms[pidx].value = pvi;
}

```

```

        pidx = pidx+1;
    }

```

2. Set the options.

```

runOptionBoolean blnPromptOption = new runOptionBoolean();
runOptionStringArray outputFormat = new runOptionStringArray();
asynchOptionInt primaryWait = new asynchOptionInt(); // primary wait threshold
string[] strarrFmt = new string[1];
runOptionBoolean blnSaveOption = new runOptionBoolean();

// Specify do not save the output
blnSaveOption.name = runOptionEnum.saveOutput;
blnSaveOption.value = false;

// execute the report in HTML format
strarrFmt[0] = "HTML";
outputFormat.name = runOptionEnum.outputFormat;
outputFormat.value = strarrFmt;

// Specify do not prompt for parameters (being passed)
blnPromptOption.name = runOptionEnum.prompt;
blnPromptOption.value = false;

// set the primary wait threshold to 0 seconds - wait indefinitely
primaryWait.name = asynchOptionEnum.primaryWaitThreshold;
primaryWait.value = 0;

// fill the array with the run options
arrRunOpts[0] = blnPromptOption;
arrRunOpts[1] = outputFormat;
arrRunOpts[2] = primaryWait;
arrRunOpts[3] = blnSaveOption;

```

3. Run the report and get the output.

```

executeReportResponse = cBIConnection.CBIRS.run(cmReportPath,
    arrParm, arrRunOpts);
// If response is not immediately complete, call wait until complete
if (!executeReportResponse.status.Equals(asynchReplyStatusEnum.complete))
{
    while (!executeReportResponse.status.Equals(asynchReplyStatusEnum.complete))
    {
        //before calling wait, double check that it is okay
        if (!hasSecondaryRequest(executeReportResponse, "wait"))
        {
            return ERR_MESG;
        }
        executeReportResponse =
            cBIConnection.CBIRS.wait(
                executeReportResponse.primaryRequest,
                new parameterValue[] {},
                new option[] {});
    }

    //After calling wait() it is necessary to check to make sure
    //the output is ready before retrieving it
    if (outputIsReady(executeReportResponse))
    {
        executeReportResponse =
            cBIConnection.CBIRS.getOutput(
                executeReportResponse.primaryRequest,
                new parameterValue[] {},
                new option[] {});
    }
}

```

Explanation

When you use the `run(objectPath, parameterValues, options)` method, you must first create an array of `ParameterValue` objects. These have to be populated if you are going to run a report with prompts. As the report is run asynchronously, the `wait(conversation, parameterValues, options)` method is used to determine when the report has completed. The report output can then be retrieved.

The section on understanding the asynchronous conversation in the chapter on running tasks in the Software Development Kit *Developer Guide* describes this process in greater detail, using the `run(objectPath, parameterValues, options)` method as an example.

Steps when writing your own programs

1. Specify the search path for the report that is associated with the request in a variable.

Tip: To find the search path for a report through the user interface, view the properties for the report in IBM Cognos Connection.

This variable will be passed to the `run` method as the `objectPath` parameter. The search path informs the application where the report is stored in the content store. For information about search paths, see “[Example - delete a report](#)” on page 33 or the appendix on search path syntax in the Software Development Kit *Developer Guide*.

2. Create a `parameterValue` array for any prompts in the report.
3. Set the values of the `parameterValue` array.
4. Create an `option` array and add any desired options to it.

For example, you can use options to specify the output format for the request.

You can also set options and parameters as properties of some objects. Options can be set in user preferences or system defaults. Options and parameter settings will be used in the following order:

- the method input parameters
- parameters in the `baseReport` object
- options defined in user preferences

For information about the options, see the chapter on enumeration sets in the Software Development Kit *Developer Guide*.

5. Call the `run(objectPath, parameterValues, options)` method, passing in the search path for the report, the parameters in the `parameterValue` array, and the options in the `option` array.

The `run` method returns an `asynchReply` object. See the chapter on running tasks in the Software Development Kit *Developer Guide* for information about understanding the asynchronous conversation.

6. The `outputPages` property of the class `asynchDetailReportOutput` contains the report output. Use the `getOutput(conversation, parameterValues, options)` method to access the output pages when the value of the `status` property of the `asynchDetailReportStatus` class is `responseReady`.
7. To navigate through output with multiple pages, use the `currentPage(conversation, parameterValues, options)`, `firstPage(conversation, parameterValues, options)`, `lastPage(conversation, parameterValues, options)`, `nextPage(conversation, parameterValues, options)`, and `previousPage(conversation, parameterValues, options)` methods.

View the samples mentioned in this section or the chapter on running tasks in the Software Development Kit *Developer Guide* for more information.

Example - delete a report

The content store is a database that stores the information about BI Bus API objects. You can use the BI Bus API methods to delete objects from the content store. For example, you may want to delete a report because it is no longer being used.

The code examples in this section show you how to do the following:

- retrieve a report from the content store using the `query(searchPath, properties, sortBy, options)` method
- delete a report from the content store using the `delete(objects, options)` method

Before using these methods, you must do the following:

- connect to the IBM Cognos Analytics server (see [“Example - connect to an IBM Cognos Analytics server”](#) on page 25)
- if authentication is required, log on to the BI server (see [“Example - log on to an IBM Cognos Analytics server”](#) on page 26)

In the code examples, the object is a report and its parent is a folder or package.

Methods

You can use the following methods to delete reports.

- `query(searchPath, properties, sortBy, options)`
- `delete(objects, options)`

See the Methods chapter in the Software Development Kit *Developer Guide* for the permissions and capabilities required for each method.

Java code

To see the code in context, view the following sample:

`installation_location/sdk/java/ReportDelete/DeleteReport.java`

The following Java code snippet demonstrates how you can delete a report from the content store.

```
delOptions.setForce(true);
delOptions.setFaultIfObjectReferenced(false);
delOptions.setRecursive(true);

try
{
    if (reportToBeDeleted != null)
    {
        System.out.println("Deleting report: " + reportToBeDeleted);

        BaseClass reportsForDeletion[] =
            new BaseClass[] { reportToBeDeleted.getBaseClassObject()};
        int delReturnCode =
            connection.getCMService().delete(reportsForDeletion, delOptions);
    }
}
```

C# code

To see the code in context, view the following sample:

`installation_location/sdk/csharp/DeleteReport/DeleteReport.cs`

The following C# code snippet demonstrates how you can delete a report from the content store.

```
deleteOptions del = new deleteOptions();
del.force = true;

// The recursive option guarantees that every report history
// attached to this report will also be deleted.
del.recursive = true;

// extract the baseClass from the report parameter
baseClass[] bc = new baseClass[1];
bc[0] = report.baseclassobject;

int nbItemsDeleted = cBICMS.delete( bc, del );

if (nbItemsDeleted>0)
{
    resultMessage = "...The item \"" + report.searchPath.value +
        "\" was successfully deleted.";
}
}
```

Explanation

When you use the `query(searchPath, properties, sortBy, options)` method, you must specify the search parameter to identify the single object you want to retrieve.

You may also specify the `properties` parameter to set the properties you want returned and the `options` parameter to set the options of the query operation, such as the maximum number of objects in the result set. In the code examples, the `options` parameter is not specified. The `query(searchPath, properties, sortBy, options)` method returns the report that you request.

In all the code examples, the search path parameter is represented by the variable `reportToBeDeleted`. This value is used to construct the reference required by the `delete(objects, options)` method. The search path specifies the objects you want to delete. Objects in the content store are organized in a single hierarchy that starts with a single root object. The root object, shown in a search path as a slash (`/`), contains all the objects available in the content store.

Objects are related to other objects through parent-child relationships (containment) and through references. Parent-child relationships are used to construct the search path property for objects persistent in the content store. These relationships describe the location of objects in the content store and are reflected in the elements of the search path for each object.

For example, the following search path shows the location in the content store of the report named `Order Product List` in the folder named `SDK Report Samples`:

```
/content/package[@name='GO Data Warehouse (query)']/folder[@name='SDK Report Samples']/report[@name='Order Product List']
```

For information about managing content, see the *Software Development Kit Developer Guide*.

In all the code examples, the `properties` parameter is represented by the variable `props`. The `properties` parameter specifies the set of properties you want to be returned. You use the type `propEnum` to define the `properties` parameter because this enumeration set contains all the possible properties for all the persisted objects in the content store. In the Java example, the `searchPath` and `defaultName` properties are returned.

In all the code examples, the results from the `query(searchPath, properties, sortBy, options)` method are used as the value for the `objects` parameter in the call to the `delete(objects, options)` method. Objects persisted in the content store are all instances of classes found in the enumeration set `classEnum`. These classes are all derived from the class `baseClass`. Therefore, the `objects` parameter is of type `baseClass` to allow for the specification of any class of persisted object.

When you use the `delete(objects, options)` method, you may also specify the following delete options:

- `faultIfObjectReferenced`
- `force`
- `recursive`

If you do not set the `recursive` option to `true`, objects that have descendants are not deleted and an exception is returned. The `force` option affects the permissions a user needs to delete an object. In both code examples, the `options` parameter is represented by the variable named `del`, which sets the `force` option to `true`.

In the code examples, an integer value is returned that indicates the number of objects deleted. If the value is greater than 0, the delete operation was successful.

Steps when writing your own programs

1. Create an array of `baseClass` objects.
2. Create a `deleteOptions` object and set its properties.

In the code examples, the `force` property is set to `true`.

3. Use the method `query(searchPath, properties, sortBy, options)` to retrieve the report from the content store. This action returns a `baseClass` object.

In the examples, the search path for the report is specified by the user.

4. Add the report to the `baseClass` array.
5. Call the `delete(objects, options)` method, using the `baseClass` array that contains the report and the `deleteOptions` object as parameters.

Tip: Prompt the user to confirm that the selected objects should be deleted.

You cannot undo a delete operation.

6. Check the integer returned for the number of objects deleted from the content store.

Example - schedule a report

Schedule a report to run at a specific time or at recurring times. For example, you may need to run a report daily or during off-peak hours.

To schedule a report using the Software Development Kit, create a `schedule` object and set the appropriate values for its properties. Then, use the `add(parentPath, objects, options)` method to add it to the content store.

These code examples show you how to schedule a report. For example, some groups in your organization may need the data in a report updated by the minute. To meet this need, you set up a report view, schedule the report to run each minute, and then verify that it is properly scheduled.

Methods

You can use the `add(parentPath, objects, options)` method to schedule a report.

See the `Methods` chapter in the Software Development Kit *Developer Guide* for the permissions and capabilities required for this method.

Java code

To see the code in context, view the following sample:

installation_location/sdk/vb/Scheduler/NewScheduler.java

The following Java code snippets demonstrate how you can schedule a report.

Steps for a Java program

1. Ensure that the currently logged-on account has a credential and obtain a reference to that credential.

```
if (! Credentials.hasCredential(connection))
{
    Credentials newCred = new Credentials();
    newCred.addCredential(connection);
}
Account logonInfo = Logon.getLogonAccount(connection);
Credential credential = new Credential();
StringProp credentialPath = new StringProp();
String credentialPathString = logonInfo.getSearchPath().getValue();
credentialPathString = credentialPathString + "/credential[@name='Credential']";
credentialPath.setValue(credentialPathString);
credential.setSearchPath(credentialPath);
BaseClassArrayProp credentials = new BaseClassArrayProp();
credentials.setValue(new BaseClass[] {credential} );
```

2. Set the schedule start date and time.

```
public DateTimeProp setScheduleStartDate(Calendar startOnDate)
{
    DateTimeProp startDate = new DateTimeProp();
    //If start date not passed set to today/now
```

```

    if (startOnDate != null)
        startDate.setValue(startOnDate);
    else
        startDate.setValue(Calendar.getInstance());

    return startDate;
}

```

3. Set the schedule end date and time.

```

public DateTimeProp setScheduleEndDate(
    Calendar endOnDate,
    String endOnTime)
{
    DateTimeProp endDate = new DateTimeProp();
    if (endOnTime.compareToIgnoreCase("onDate") == 0)
        if (endOnDate != null)
        {
            endDate.setValue(endOnDate);
        }
    else
        System.out.println(
            "Parameter endOnTime cannot be onDate if no enddate provided");
    return endDate;
}

public NmtokenProp setScheduleEndTime(String endOnTime)
{
    NmtokenProp endTime = new NmtokenProp();
    if (endOnTime != null)
    {
        endTime.setValue(endOnTime);
    }
    else
        System.out.println(
            "Parameter endOnTime cannot be null! Options: indefinite or onDate");
    return endTime;
}

```

4. Assign the current user as the owner.

```

BaseClass[] owners = new BaseClass[] { Logon.getLogonAccount(connection) };
BaseClassArrayProp ownersProp = new BaseClassArrayProp();
ownersProp.setValue(owners);

```

5. Set the options.

```

OptionArrayProp roap = new OptionArrayProp();
String reportPath = report.getBaseClassObject().getSearchPath().getValue();

//Set options
roap.setValue(
    this.setSchedulerOptions(
        outFormat,
        delivery,
        reportPath,
        printerName,
        saveasName,
        pageOrientation,
        paper,
        emails,
        connection));

...
RunOptionSaveAs saveAs = new RunOptionSaveAs();
MultilingualToken[] obj = new MultilingualToken[1];

//Set the name of the reportView
obj[0] = new MultilingualToken();
obj[0].setLocale("en-us");

//If no name provided use default
if (saveasName != null)
{
    obj[0].setValue(saveasName);
}
else

```

```

{
    obj[0].setValue("View of Report " + reportPath);
}

//Save the object as report view with name saveasName
saveAs.setName(RunOptionEnum.saveAs);
saveAs.setObjectClass(ReportSaveAsEnum.reportView);
saveAs.setObjectName(obj);
saveAs.setParentSearchPath(csh.getParentPath(connection, reportPath));

```

6. Set the properties of the schedule object and add it to the report.

```

newSchedule.setActive(isActive);
newSchedule.setCredential(credentials);
newSchedule.setEndDate(this.setScheduleEndDate(endOnDate, endOnTime));
newSchedule.setEndType(this.setScheduleEndTime(endOnTime));
newSchedule.setOwner(ownersProp);
newSchedule.setParameters(pv);
newSchedule.setOptions(roap);
newSchedule.setStartDate(this.setScheduleStartDate(startOnDate));
newSchedule.setType(howOften);

// add the schedule to the report
AddOptions ao = new AddOptions();
ao.setUpdateAction(UpdateActionEnum.replace);
BaseClass newBc = connection.getCMService().add(
    new SearchPathSingleObject(reportPath),
    new BaseClass[] { newSchedule },
    ao)[0];

```

C# code

To see the code in context, view the following sample:

installation_location/sdk/csharp/Schedule/Schedule.cs

Before you can use this sample, you must create a credential object for the account that you will use to run the sample. For more information about creating credentials, see the *Administration and Security Guide*.

The following C# code snippets demonstrate how you can schedule a report.

Steps for a C# program

1. Ensure that the currently logged-on account has a credential and obtain a reference to that credential.

```

//get the credential for the schedule
credential schedCred = connection.getCredential();
baseClassArrayProp credentials = new baseClassArrayProp();
credentials.value = new baseClass[] {schedCred};
newSched.credential = credentials;

```

2. Set the schedule start and end dates and times.

```

// set schedule time to now + 2 minutes
System.DateTime startTime = new DateTime();
startTime = DateTime.Now;
startTime = startTime.AddMinutes(2);
dateTimeProp schedStartTime = new dateTimeProp();
schedStartTime.value = startTime;
newSched.startDate = schedStartTime;

// set schedule end time to now + 5 minutes
System.DateTime endTime = new DateTime();
endTime = DateTime.Now;
endTime = endTime.AddMinutes(5);
dateTimeProp schedEndTime = new dateTimeProp();
schedEndTime.value = endTime;
newSched.endDate = schedEndTime;

// set the schedule end type
nmtokenProp endType = new nmtokenProp();

```



```
endType.value = "onDate";
newSched.endType = endType;
```

3. Assign the current user as the owner.

```
//set the owner
baseClassArrayProp ownersProp = new baseClassArrayProp();
baseClass[] owners = new baseClass[]
    { SamplesConnect.getLogonAccount(connection) };
ownersProp.value = owners;
newSched.owner = ownersProp;
```

4. Set the options.

```
//Set the name of the reportView
multilingualToken[] reportViewName = new multilingualToken[1];
reportViewName[0] = new multilingualToken();
reportViewName[0].locale = "en-us";
reportViewName[0].value = "View of Report " +
    reportToSchedule.baseclassobject.defaultName.value;

//Save the output as report view with name saveasName
runOptionSaveAs saveAs = new runOptionSaveAs();
saveAs.name = runOptionEnum.saveAs;
saveAs.objectClass = reportSaveAsEnum.reportView;
saveAs.objectName = reportViewName;
saveAs.parentSearchPath = reportToSchedule.parentPath.value;

//Turn off prompting
runOptionBoolean prompt = new runOptionBoolean();
prompt.name = runOptionEnum.prompt;
prompt.value = false;

//set the output format
runOptionStringArray format = new runOptionStringArray();
format.name = runOptionEnum.outputFormat;
format.value = new string[] { "HTML" };

//put the run options where they need to go
runOption[] schedRunOptArr = new runOption[3];
schedRunOptArr[0] = saveAs;
schedRunOptArr[1] = prompt;
schedRunOptArr[2] = format;

runOptionArrayProp schedRunOptions = new runOptionArrayProp();
schedRunOptions.value = schedRunOptArr;
newSched.runOptions = schedRunOptions;
```

5. Set the properties of the schedule object and add it to the report.

```
//mark the schedule as active
booleanProp isActive = new booleanProp();
isActive.value = true;
newSched.active = isActive;

//Set the type of schedule
cognosdotnet_10_2.nmtokenProp scheduleType = new nmtokenProp();
scheduleType.value = "daily";
newSched.type = scheduleType;

//make the schedule for every x minutes
nmtokenProp period = new nmtokenProp();
period.value = "minute";
newSched.dailyPeriod = period;

//set that every x minutes to be every 1 minute
positiveIntegerProp runFreqProp = new positiveIntegerProp();
runFreqProp.value = "1";
newSched.everyNPeriods = runFreqProp;
...
searchPathSingleObject reportSearchPath = new searchPathSingleObject();
reportSearchPath.Value =
    ((baseClass) reportToSchedule.baseclassobject).searchPath.value;
// add the schedule to the report
addOptions ao = new addOptions();
ao.updateAction = updateActionEnum.replace;
baseClass newBc = connection.CBICMS.add(
    reportSearchPath,
```

```
new baseClass[] { newSched },
ao)[0];
```

Explanation

To create the schedule, you first create a `schedule` object and specify values for its properties. The following are examples of the properties that may be specified:

- `active`
- `credential`
- `dailyPeriod`
- `endDate`
- `endType`
- `everyNPeriods`
- `options`
- `startDate`

See the description of the `schedule` class in the chapter on classes in the *Software Development Kit Developer Guide* for a complete list and description of the properties.

If the report contains prompts, you must also specify parameters in the `parameters` property.

The schedule in a `schedule` array is passed to the `add(parentPath, objects, options)` method and the schedule is created in the content store.

Steps when writing your own programs

1. Create an array of `schedule` objects.
2. Create a new `schedule` object.
3. Create a new `addOptions` object.
4. Create a `dateTimeProp` object for the starting date and time for the `schedule` object, and set its value to the required start date and time. Assign this `dateTimeProp` object to the `startDate` property of the `schedule`.
5. Build an array of `option` objects for the `schedule` object. These options will apply whenever the scheduled report runs. Assign this array to the `value` property of the `options` property of the `schedule` object.
6. Set the values for the other properties of the `schedule` object.
7. Add the `schedule` object to the `schedule` array.
8. Set desired values for the properties of the `addOptions` object.
9. Call the `add(parentPath, objects, options)` method, passing the search path, the `schedule` array, and the `addOptions` object as parameters.

Example - modify a report

Each IBM Cognos Analytics report is defined by a report specification. The report specification is an XML document that describes the report data query and the appearance of the report, including the columns shown, header and footer information, and colors. You can change the content and appearance of a report by modifying the report specification. For example, you can change a column title, or change the way report data is grouped.

When you modify a report in Reporting, you are changing the report specification. You can also change the report specification through the Software Development Kit. If you want to run the new report without saving it, use the `runSpecification(specification, parameterValues, options)` method to run the report specification.

Methods

You can use the following methods when modifying a report.

- `add(parentPath, object, options)`
- `runSpecification(specification, parameterValues, options)`
- `query(objectPath, parameterValues, options)`
- `update(object, options)`
- `validateSpecification(specification, parameterValues, options)`

See the Methods chapter in the Software Development Kit *Developer Guide* for the permissions and capabilities required for these methods.

Java code

To see the code in context, view the following samples:

- `installation_location/sdk/java/ReportSpec/EditReportSpec.java`
- `installation_location/sdk/java/ReportSpec/ReportObject.java`

The following Java code snippets demonstrate how you can modify the report specification. This example modifies the columns in a report.

Steps for a Java program

1. Extract the report specification.

```
try
{
    String reportPath = report.getBaseClassObject().getSearchPath().getValue();

    Option[] qOpts = new Option[2];

    ReportServiceQueryOptionBoolean upgradeSpecFlag =
        new ReportServiceQueryOptionBoolean();
    upgradeSpecFlag.setName(ReportServiceQueryOptionEnum.upgrade);
    upgradeSpecFlag.setValue(true);

    ReportServiceQueryOptionSpecificationFormat specFormat =
        new ReportServiceQueryOptionSpecificationFormat();
    specFormat.setName(ReportServiceQueryOptionEnum.specificationFormat);
    specFormat.setValue(SpecificationFormatEnum.report);

    qOpts[0] = upgradeSpecFlag;
    qOpts[1] = specFormat;

    AsynchReply qResult =
        connect.getReportService().query(
            new SearchPathSingleObject(reportPath),
            new ParameterValue[] {},
            qOpts);

    if ( (qResult.getStatus() == AsynchReplyStatusEnum.working)
        || (qResult.getStatus() == AsynchReplyStatusEnum.stillWorking) )
    {
        while ( (qResult.getStatus() == AsynchReplyStatusEnum.working)
            || (qResult.getStatus() == AsynchReplyStatusEnum.stillWorking) )
        {
            qResult = connect.getReportService().wait(qResult.getPrimaryRequest(),
                new ParameterValue[] {},
                new Option[]
            {});
        }
    }

    // extract the report spec
    if (qResult.getDetails() != null)
    {
        for (int i = 0; i < qResult.getDetails().length; i++)
        {
```

```

        if (qResult.getDetails()[i] instanceof AsynchDetailReportObject)
        {
            reportSpec = ( (AsynchDetailReportObject)qResult.getDetails()[i])
                .getReport().getSpecification().getValue();
        }
    }
}
}

```

2. Parse the specification into an XML object tree.

```

//This strips out the leading XML declaration
sMetaData = sMetaData.substring(sMetaData.indexOf(">") + 2);
SAXReader xmlReader = new SAXReader();
ByteArrayInputStream bais =
    new ByteArrayInputStream(sMetaData.getBytes("UTF-8"));
oDom = xmlReader.read((InputStream)bais);

```

3. As an example, modify the XML.

```

public void addColumn(
    String p_sColumnName,
    String p_sExpression,
    int position)
{
    addDataItem(p_sColumnName, p_sExpression);
    oAPI.addListColumn(p_sColumnName, position);
}

```

4. Add the specification back into the Content Store.

```

Report rpt = new Report();
AddOptions addOpts = new AddOptions();
TokenProp rptDefaultName = new TokenProp();
AnyTypeProp ap = new AnyTypeProp();
rptDefaultName.setValue(reportName);
String reportXML = getXML();

int iStartReport = reportXML.indexOf("<report");
String reportOut = reportXML.substring(iStartReport);

ap.setValue(reportOut);
rpt.setDefaultName(rptDefaultName);
rpt.setSpecification(ap);
addOpts.setUpdateAction(UpdateActionEnum.replace);

String parentPath = parent.getSearchPath().getValue();
try
{
    connection.getReportService().add(new SearchPathSingleObject(parentPath),
        rpt, addOpts);
}

```

C# code

To see the code in context, view the following sample:

installation_location/sdk/csharp/CreateReport/CreateReport.cs

The following C# code snippets demonstrate how you can modify the report specification. This example creates a report specification and adds a column to the report. It uses the .NET Framework classes to manipulate XML.

Steps for a C# program

1. Create a report specification.

```

XmlDocument xmlDoc = new XmlDocument();
XmlNode root,
    currentNode,
    nextNode,
    listColumnsNode,
    cssNode,

```

```

        currentStyleNode,
        styleNode,
        selectionNode;
XmlAttribute attribNode;

string defaultPackageName, strName, strReportXML;
report reportName = new report();
addOptions addOptions = new addOptions();
tokenProp tokenName = new tokenProp();
anyTypeProp anyTypeName = new anyTypeProp();
reportServiceReportSpecification rspecReport =
    new reportServiceReportSpecification();

string myNS = "http://developer.cognos.com/schemas/report/7.0/";

//
//root report element
//
root = xmlDoc.CreateNode(XmlNodeType.Element, "report", myNS);
attribNode = xmlDoc.CreateAttribute("expressionLocale");
attribNode.Value = "en-us";
root.Attributes.Append(attribNode);

xmlDoc.AppendChild(root);

//Add all the rest of the foundation report elements

```

2. Add a column to the report specification.

```

currentNode = xmlDoc.CreateNode(XmlNodeType.Element,
    "dataItemLabel", myNameSpace);
attribNode = xmlDoc.CreateAttribute("refDataItem");
attribNode.Value = columnLabel;
currentNode.Attributes.Append(attribNode);
...
currentNode = xmlDoc.CreateNode(XmlNodeType.Element,
    "dataItemValue", myNameSpace);
attribNode = xmlDoc.CreateAttribute("refDataItem");
attribNode.Value = columnValue;
currentNode.Attributes.Append(attribNode);

```

3. Add the specification to the Content Store.

```

tokenName.value = strName;
anyTypeName.value = strReportXML;

reportName.defaultName = tokenName;
reportName.specification = anyTypeName;
addOptions.updateAction = updateActionEnum.replace;
reportName.specification.value = strReportXML;

searchPathSingleObject targetPath = new searchPathSingleObject();
targetPath.Value = targetLocation;

connection.CBIRS.add(targetPath, reportName, addOptions);

```

Explanation

These examples modify a report specification by following several steps. First, the report specification is retrieved from the content store. The report specification is then parsed into a tree structure and modified using a DOM parser. It is then converted back to XML. In the Java sample, it is saved to the content store using the add method.

Steps when writing your own programs

1. Query the content store using the method `query(objectPath, parameterValues, options)` to retrieve the specification for your report.

The query method returns an array of details, including the report. The specification is the value of the `specification` property for the returned report.

2. Modify the report specification to meet your requirements using your preferred technique.

Examples of different ways to do this are as follows:

- Use an XML parser, such as a DOM parser, with your programming language.
 - Save the report specification as an XML file, and use any text editor to make your changes. In some XML editors, you may be able to associate the report specification with the IBM Cognos document type definition (DTD). This ensures that your changes result in a report specification that is still valid.
3. Use the method `validateSpecification(specification, parameterValues, options)` to confirm that the new report specification is valid.
 4. To save the new report, do one of the following:
 - Use the method `update(object, options)` to overwrite the existing report in the content store with the new specification.
 - Use the method `add(parentPath, object, options)` to add a new report to the content store. Ensure that the search path you use for the new report is unique and that you do not overwrite another report.

Example - run the script player

You must have installed Framework Manager before you run this example. The sample action logs can be found in *installation_location/webcontent/samples/Models/gosales_scriptplayer*. The *gosales_scriptplayer.lst* file in the same location can be used to run the action logs in sequence. This action generates a model named *gosales_scriptplayer* and publishes a package to the content store.

The action logs are described here.

01gosaddsrc.xml

Creates the model and adds a data source.

02goslangdef.xml

Defines the languages used by the model.

03gosmodqs.xml

Modifies a query subject.

04gosrenam.xml

Renames columns.

05gosprops.xml

Updates properties.

06gosorg.xml

Adds namespaces.

07goslangimp.xml

Imports a set of translations using text files stored in the same location.

08gospac.xml

Creates and publishes a *gosales_scriptplayer* package.

To run the script player, open a command prompt in *installation_location/bin* and run the following command:

```
BmtScriptPlayer -l ../webcontent/samples/Models/gosales_scriptplayer/  
gosales_scriptplayer.lst
```

Chapter 4. Managing security

IBM Cognos provides a flexible architecture that allows you to customize security according to your needs and integrate easily with your existing infrastructure.

The BI Bus API provides application developers with the ability to:

- log in and out using a security provider or a custom authentication provider to establish a security context for applications
- manage users, groups, and roles
- apply policies to control access to objects in the content store
- grant capabilities to allow specified users, groups, or roles to use specific functionality

Authenticating Users

Authentication is the process of verifying the identity of an individual, application, or device using an established identification process. Individuals, or users, are represented in IBM Cognos Analytics by `bibus » account` class objects created in the content store. Users must be authenticated before they can be authorized to use any secured resources.

The IBM Cognos Analytics security model leverages one or more authentication sources accessed through authentication providers.

To use IBM Cognos Analytics, an application using the BI Bus API must successfully connect to IBM Cognos Analytics and log on as a named user or an anonymous user. An application using the BI Bus API connects to IBM Cognos Analytics using an instance of a web service class.

To log on as a named user, the application must provide a credential, usually in the form of a user ID and password, as required by your authentication provider configuration.

For anonymous access, logging on is not required. If anonymous access is enabled and there is no logon or passport included in a request, anonymous access is assumed.

Authentication Providers

User authentication in IBM Cognos Analytics is managed by authentication providers. Authentication providers define users, groups, and roles used for authentication. User names, IDs, passwords, regional settings, personal preferences are some examples of information stored in the providers. If you set up authentication for IBM Cognos Analytics, users must provide a valid credential, such as user ID and password, at logon time.

Custom Authentication Provider

An API is provided with the IBM Cognos Software Development Kit that allows you to develop a custom authentication provider solution for your applications. If your provider is not already supported, you can use this API to create your own authentication provider, or create a trusted signon to an existing provider.

For information about supported providers, visit the *IBM Cognos Customer Center*.

For information about the developing a custom solution for your own authentication provider, see the *IBM Cognos Analytics Custom Authentication Provider Developer Guide*.

Connect to IBM Cognos Analytics

Applications using the BI Bus API connect to IBM Cognos Analytics by creating an IBM Cognos Analytics service class object and using it to issue a request. When you send a connection request to the IBM Cognos Analytics server, you provide the server URL. If you are connecting through IBM Cognos

Connection, you launch your Web browser and supply the gateway URL provided by your administrator, which is of the form `http[s]://host:port/ibmcognos/bi/v1/disp`

You cannot use this method of access with your SDK applications because the gateway is configured by default to use the following URI:

```
http://localhost:9300/p2pd/servlet/dispatch/ext
```

This causes it to block requests from SDK applications.

Depending on your network configuration, set up your SDK application to do one of the following:

- connect directly to the dispatcher using the internal dispatcher URI (`http://localhost:9300/p2pd/servlet/dispatch`)
- connect to an additional, dedicated gateway that is configured to connect to the dispatcher using the internal dispatcher URI. You must configure appropriate security for this gateway. This method is useful when the SDK application is outside a network firewall.

Note: Do not change your main gateway to use the internal dispatcher URI. Doing so reduces the security of the IBM Cognos Analytics portal and studios.

For more information about the internal dispatcher URI, see the *IBM Cognos Analytics Installation and Configuration Guide*.

Sample code illustrating how to connect to IBM Cognos Analytics can be found in [Chapter 12, “Services,”](#) on page 129.

Log On

Applications using the BI Bus API may log on to IBM Cognos Analytics using an account defined by an authentication provider. For information about securing capabilities, see [“Managing Capabilities”](#) on page 51.

Before you begin

You must include your credential ([“The Credentials Parameter”](#) on page 47) with the logon.

Optionally, you can specify one or more roles during log on. Choosing a role when you log on determines the access permissions you have. If no role is specified, you are granted the access permissions of all the roles and groups to which you belong. If one or more roles are specified, you have the access permissions of the specified roles, as well as the access permissions of all the groups to which you belong.

If you configured IBM Cognos Analytics with multiple namespaces, you can log on to all available namespaces using a BI Bus API application, provided that you have valid credentials for these namespaces. You cannot log onto the same namespace multiple times.

Note: If your environment has Secure Sockets Layer (SSL) applied, all logon messages sent to and from IBM Cognos Analytics are secured based on the SSL configuration in IBM Cognos Analytics.

For information about sample applications that use these methods, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

Procedure

1. Use the [authentication](#) » [logon\(credentials, roles\)](#) method to log on to a configured namespace.
2. To specify a role to log on as, use the [roles](#) parameter.

The [authentication](#) » [logon\(credentials, roles\)](#) method requires a string array for roles. Even if you do not pass IBM Cognos Analytics values, you must still include the roles parameter in your method call.

Tip: You can use the [authentication](#) » [selectRoles\(roles\)](#) method at any time to change the role for the current session.

For more information, refer to [“Using the selectRoles Method has no Effect”](#) on page 117.

Sample Code in Java

If you are writing your application in Java, log on to IBM Cognos Analytics using the following code:

```
StringBuffer credentialXML = new StringBuffer();
credentialXML.append("<credential>");
credentialXML.append("<namespace>");
credentialXML.append(namespace);
credentialXML.append("</namespace>");
credentialXML.append("<username>");
credentialXML.append(uid);
credentialXML.append("</username>");
credentialXML.append("<password>");
credentialXML.append(pwd);
credentialXML.append("</password>");
credentialXML.append("</credential>");
...
XmlEncodedXML credentialXEX = new XmlEncodedXML();
credentialXEX.setValue(credentialXML.toString());
connection.getCMService().logon(credentialXEX, null);
```

The Credentials Parameter

A credential, or set of credentials, contains information about the identity of a user, typically a user name and password. IBM Cognos Analytics uses a generic representation of credentials to access secured resources, such as database accounts or authentication namespaces.

The credentials parameter of the [authentication](#) » [logon\(credentials, roles\)](#) method contains authentication information encoded as XML. If the toolkit you are using does not escape reserved characters in XML strings that are sent to an IBM Cognos Analytics server, you must explicitly escape these characters in the XML credentials string in your application.

Log Off

Use the [authentication](#) » [logoff\(\)](#) method to terminate the user session and remove the passport from the [bibus](#) » [CAM](#) object.

An application logs off only once to end the session for all namespaces.

Managing Authorization

Authorization is the process of determining what entitlements an authenticated consumer has. A consumer can be an individual, application, or device. Entitlements control whether a consumer can access specific data (files or objects) in a system, or whether they can perform specific actions, such as use functionality provided by an application.

You can manage authorization in IBM Cognos Analytics in the following ways:

- You can modify users, groups, and roles as defined by your authentication provider or in IBM Cognos Analytics. You cannot create users in IBM Cognos Analytics.
- You can control access to objects in the content store, such as folders, reports, data sources, and configuration objects through the use of policies.
- You can grant capabilities to users, groups, and roles to control access to specific functionality in IBM Cognos Analytics.
- You can assign access permissions to certain profiles, limiting IBM Cognos Analytics - Reporting functionality to specific users, groups, and roles.

When an application calls a BI Bus API method, IBM Cognos Analytics checks to ensure that the caller has the appropriate permissions and capabilities before processing the request.

For information about the permissions and capabilities required to use a method, see the reference topic for the method.

Before You Begin

To manage authorization in IBM Cognos Analytics, you must have the following access permissions:

Goal	Permissions Required
Use the administrative tools.	execute permission to the Administration secured function
Use the Directory tool to modify group and role memberships.	execute permission to the Directory secured feature
Use the Capabilities tool to modify access permissions to capabilities.	execute permission to the Capabilities secured feature
Modify access permissions to an object.	setPolicy permission for the object, such as a group or role
Access child objects, such as groups and roles.	read and traverse permission for the parent object, the Cognos namespace
Add a new group or role.	write permission to the parent Cognos namespace
Add a new member to a group or role.	write permission for the object, the "Cognos" group or role, and traverse permission for the parent "Cognos" namespace

Note: Members of the Server Administrators, Report Administrators, Portal Administrators, and Metrics Administrators roles automatically have access to their respective administrative tools. For more information, see [Chapter 34, "Initial content store settings,"](#) on page 1607.

Managing Users, Groups, and Roles

Users, groups, and roles are used for authentication, authorization, and administration. In IBM Cognos Analytics, you can use the users, groups, and roles created in authentication sources and the groups and roles created in IBM Cognos Analytics.

You cannot create users in IBM Cognos Analytics.

Create a Group or Role in the Cognos Namespace

Groups and roles are created in IBM Cognos Analytics to address application security and administration needs specific to IBM Cognos Analytics. You can use the BI Bus API to create groups and roles in the Cognos namespace.

You can also add members to an existing group or role (["Add Members to a Group or Role in the Cognos Namespace"](#) on page 49), or delete a group or role from the Cognos namespace (["Remove a Group or Role from the Cognos Namespace"](#) on page 49).

Procedure

1. Log on to IBM Cognos Analytics using an account with the necessary permissions.
2. Create a new [bibus » group](#) or [bibus » role](#) object.
3. Specify the properties you want for your [bibus » group](#) or [bibus » role](#).
4. Use the [bibus » addOptions](#) class to specify the options for the add operation.

5. Use the `report » add(parentPath, object, options)` method to create the new group or role object.

Add Members to a Group or Role in the Cognos Namespace

You can use the BI Bus API to modify the membership of a group or role in the Cognos namespace by adding new members. For example, you want to add a user to the Server Administrators role so that user can create deployment packages.

You can also create a new group or role ([“Create a Group or Role in the Cognos Namespace” on page 48](#)) or remove members from a group or role ([“Remove Members from a Group or Role in the Cognos Namespace” on page 49](#)) in the Cognos namespace.

Procedure

1. Log on to IBM Cognos Analytics using an account with the necessary permissions.
2. Create the `bibus » account`, `bibus » group`, or `bibus » role` objects to add to the group or role.
Note: Roles can also have other roles as members.
3. If you are adding members to a group or role that has an existing membership list, use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the `members` property or the `members` property.
4. Add the new `bibus » account`, `bibus » group`, or `bibus » role` objects to the membership list.
5. Use the `content » update(objects, options)` method to save the changes in the content store.

Remove a Group or Role from the Cognos Namespace

You can use the BI Bus API to remove an unused group or role from the Cognos namespace when you maintain your IBM Cognos Analytics security system.

You can also remove members from a group or role in the Cognos namespace ([“Remove Members from a Group or Role in the Cognos Namespace” on page 49](#)).

Procedure

1. Log on to IBM Cognos Analytics using an account with the necessary permissions.
2. Use the `content » delete(objects, options)` method to remove the group or role object.

Remove Members from a Group or Role in the Cognos Namespace

You can use the BI Bus API to modify the membership of a group or role by removing members.

You can also remove a group or role ([“Remove a Group or Role from the Cognos Namespace” on page 49](#)) or add members to a group or role in the Cognos namespace ([“Add Members to a Group or Role in the Cognos Namespace” on page 49](#)).

Procedure

1. Log on to IBM Cognos Analytics using an account with the necessary permissions.
2. Use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the `members` property or the `members` property.
3. Create a new membership list and add the members from the existing membership list, except for the members or governors you want to remove.
4. Use the `content » update(objects, options)` to save the changes in the content store.

Managing Policies

A security policy in IBM Cognos Analytics defines a set of entitlements that determine access rights for individual users, groups, or roles.

In IBM Cognos Analytics, you can secure your organization's data by managing security policies for objects in the content store. You specify which users, roles, or groups have access to a specific report or other content in the content store. You can use the BI Bus API to set security policies on any object derived from the [bibus » baseClass](#) class, such as folders, reports, and data sources.

You can:

- grant access to an object
- revoke access to an object
- deny access to an object

Set the Security Policies for a Content Store Object

Setting security policies requires adding and removing [bibus » policy](#) objects to and from the [policies](#) property of a content store object. The [policies](#) property contains an array of [bibus » policy](#) objects, each specifying an [bibus » account](#), [bibus » group](#), or [bibus » role](#), in the [bibus » policy » securityObject](#) property property.

Before you begin

You must add one [bibus » policy](#) object for each [bibus » account](#), [bibus » group](#), or [bibus » role](#), for which you want to define security. When you specify a [bibus » policy](#) for an object, you must specify the policy completely.

Setting access permissions requires updating the [permissions](#) property of [bibus » policy](#) objects. Updating the [permissions](#) property includes specifying the [bibus » permission » name](#) property of the [bibus » permission](#), which can be read, write, execute, setPolicy, or traverse, and specifying the value for [bibus » permission » access](#) property, such as [grant](#).

You must have one [bibus » permission](#) object for each permission you want to grant, such as one for execute and another for write permission. Therefore, for every [bibus » policy](#) object in the array, there can be up to five [bibus » permission](#) objects.

If a policy is not defined for an object, it is acquired from the parent object.

Objects that are created by Content Manager when IBM Cognos Analytics is initialized have policies set when they are created. For information about these predefined security objects, see [Chapter 34, “Initial content store settings,”](#) on page 1607.

Procedure

1. Log on to IBM Cognos Analytics using an account with the necessary permissions ([“Before You Begin”](#) on page 48).
2. Use the [content » query\(searchPath, properties, sortBy, options\)](#) method to retrieve the [bibus » baseClass » policies](#) property for the object.
3. To change the security policy, do one of the following:
 - To add or remove a policy, add or remove instances of the [bibus » policy](#) to or from the [value](#) property of the [bibus » baseClass » policies](#) object, and specify the [bibus » policy » securityObject](#) property for each [bibus » policy](#) object.
 - To modify an existing policy, traverse the [bibus » policy](#) array to find the object that matches the [bibus » baseClass » searchPath](#) property to the [bibus » policy » securityObject](#) property you want to modify.
4. To change the [permissions](#) property for a [bibus » policy](#) object, do one of the following:
 - To grant permission, add a [bibus » permission](#) object, set the [bibus » permission » name](#) property to the desired value, and set the [bibus » permission » access](#) property to [grant](#).
 - To revoke permission, delete the [bibus » permission](#) object. If it is the last [bibus » permission](#) object in the [bibus » policy](#), you can remove the [bibus » policy](#) object.

- To restrict access, add a `bibus » permission` object, set the `bibus » permission » name` property to the desired value and set the `bibus » permission » access` property to `deny`. If you set the `bibus » permission » access` property to `deny`, access is always restricted, regardless of other permissions.
5. If you want the `policies` property of all child objects to acquire the values of this content store object, set the `bibus » schemaInfo » applyUpdateToDescendants` property to `true`.
 6. Use the `content » update(objects, options)` method to save the changes in the content store.

Managing Capabilities

You permit an `bibus » account`, `bibus » group`, or `bibus » role` to use IBM Cognos functionality in specific contexts by granting both global capabilities and object-level capabilities.

The mechanisms and structures used to assign capabilities support fine-grained policy configurations, similar to those of a security policy. For example, to allow an `bibus » account`, to create reports using IBM Cognos Analytics - Reporting on the Sales `bibus » package`:

- a member of the Directory Administrators `bibus » role` grants the `bibus » account` the global `canUseReportStudio` capability
- a member of the Report Administrators `bibus » role` grants the `bibus » account` the object-level `canUseReportStudio` capability on the Sales `bibus » package`

Similarly, you determine the functionality available to an `bibus » account`, `bibus » group`, or `bibus » role`, by querying global capabilities for who has been granted or denied them, and then querying individual objects, such as an instance of the `bibus » package`, for who has specific capabilities on the object. The `effectiveUserCapabilities` property returns what the current `bibus » account` is entitled to do with an object. To simplify individual `bibus » account`, UI development, `effectiveUserCapabilities` property includes capabilities inherited from a superior container, such as a `bibus » folder`, and globally. Administrative UI development requiring the acquisition of multiple `bibus » account`, `bibus » group`, or `bibus » role` capabilities involves querying and matching global and object-level capabilities.

The `bibus » package` class supports policies that specify the set of capabilities granted to an `bibus » account` when using a particular `bibus » package`. The `bibus » content` and `bibus » folder` classes also include the `userCapabilityPolicies` property to simplify capability administration. Capability policies for an object are acquired from a superior object, such as a `bibus » folder`, unless specified in the object.

The Public Folders predefined object defines default capability policies for a contained `bibus » package`. The Everyone `bibus » group` is granted all capabilities that can be specified on an object. By default, the `canUseObjectCapabilities` capability is not granted to any `bibus » role`. A member of the Directory Administrators `bibus » role` must grant the capability to someone before object capabilities can be assigned.

Granting Capabilities

The following steps show how to grant both global and object-level capabilities. It does not matter which level of capability (global or object) is granted first, only that both be granted to enable functionality on an object for an `bibus » account`, `bibus » group`, or `bibus » role`.

Steps for granting global-level capabilities

Perform the following procedure to grant global-level capabilities.

Procedure

1. Query the capability in the content store, specifying the `policies` property, `bibus » baseClass » searchPath` property, and `bibus » baseClass » defaultName` property properties in the `properties` parameter of the query method.
2. Change the `policies` property to include a policy granting the `bibus » account`, `bibus » group`, or `bibus » role` `execute` and `traverse` permission on the capability.

3. Use the `content » update(objects, options)` method to save the changed `policies` property, in the content store.

Sample Java code illustrating how to grant capabilities is provided with the IBM Cognos Software Development Kit.

Steps for granting object-level capabilities

Perform the following procedure to grant object-level capabilities.

Procedure

1. Query the `bibus » package` specification in the content store, specifying `policies` property, `bibus » baseClass » searchPath` property, `bibus » baseClass » defaultName` property and the `userCapabilities` property of interest.
2. If the `userCapabilities` property already includes a `bibus » userCapabilityPolicy` for the object, change the `bibus » userCapabilityPolicy` to include a `bibus » userCapabilityPermission` object granting the capability to the `bibus » account`, `bibus » group`, or `bibus » role`.

If the `bibus » userCapabilityPolicy` does not already exist, create a new `bibus » userCapabilityPolicy` that includes a `bibus » userCapabilityPermission` for granting the capability to the `bibus » account`, `bibus » group`, or `bibus » role` and change the `userCapabilities` property to include the new `bibus » userCapabilityPolicy`.

3. Use the `content » update(objects, options)` method to save the object in the content store.

Querying Capabilities Using the BI Bus API

The following procedure show how to use the capabilities of the current `bibus » account` to determine which objects should be presented in a UI.

Sample Java code illustrating how to query capabilities is provided with the IBM Cognos Software Development Kit.

Procedure

1. Query the object, specifying the `effectiveUserCapabilities` property in the `properties` parameter of the query method.
2. Check the values in the `effectiveUserCapabilities` property for the presence of the capabilities of interest on the object

Querying Capabilities Using the Capability Cookie Bit Index

The capability cookie value can be used to determine the user capabilities assigned to a user independent of any particular `bibus » package`. The `userCapabilityPolicies` property defined for a `bibus » package` can restrict the capabilities assigned to a user when using that `bibus » package`, but cannot grant additional capabilities.

The cookie value is an encoded bit sequence that is digitally signed to allow IBM Cognos Analytics software to detect tampering. Each bit in the unencoded bit sequence represents a user capability. The bits are identified from right to left starting from bit 0. A user has the capability if the corresponding bit in the cookie value is 1.

Procedure

1. Extract the encoded bit sequence from the cookie value that contains the signature:

- Retrieve the `userCapabilities` or `userCapabilitiesExcookie`.

The `userCapabilities` cookie supports a bit index from 0-127. The `userCapabilitiesEx cookie`, added in version 11.1.2, extends the number of supported bits beyond 127.

- URL decode the cookie value in accordance with RFC 2396 to replace any escaped characters with their unescaped form.
 - Remove the signature by deleting the ampersand (&) and everything following it.
2. Separate the sequence of hexadecimal digits at each semicolon (;).
Zeroes may be added on the leftmost end of each chunk to make up eight hexadecimal digits.
 3. Convert each chunk of hexadecimal digits into a sequence of bits.
 4. Concatenate the chunks into one bit sequence.

Decoding the user capability cookie value

The following example shows how to decode the cookie value 1%3B5f0f5%3B233c517e%3B393c7a4%26ARQAAACGJI7702Pf4a4Qg0G%2Bv10gQjvLQ8QQfEb2TvXfW9g7FRHJmGKZig8l, and demonstrates how to construct the bit sequence from the cookie value.

1. The escaped characters are replaced with their unescaped form:
1;5f0f5;233c517e;393c7a4&ARQAAACGJI7702Pf4a4Qg0G+v10gQjvLQ8QQfEb2TvXfW9g7FRHJmGKZig8l
2. The signature is removed by deleting the ampersand (&) and everything following it:
1;5f0f5;233c517e;393c7a4
3. The hexadecimal digit sequences are separated at each semicolon into the following chunks:
00000001 (seven zero digits added to the leftmost end), 0005f0f5 (three zero digits added to leftmost end), 233c517e, and 0393c7a4 (one zero digit added to leftmost end).
4. Each hexadecimal digit sequence is converted to a sequence of bits:
bits: 00000000000000000000000000000001, 00000000000001011111000011110101, 00100011001111000101000101111110, and 00000011100100111100011110100100.
5. The bit sequences are concatenated to produce a single bit sequence:

```
000000000000000000000000000000010000000000001011111000011110101
001000110011110001010001011111100000011100100111100011110100100
```

Each bit corresponds to a specific capability (some bits are reserved). For example, in the preceding sequence, bit 5 is set to 1; therefore, we can determine that the user has the [canUseCognosViewerRunWithOptions](#) capability. Bit 96 is set to 1; therefore, we can determine that the user has the [canUseRepository](#) capability.

Security Prerequisites for Requests

Many of the methods available in the IBM Cognos Software Development Kit have specific requirements for access permissions and capabilities. If the account used in the security context of the calling application does not satisfy these requirements, the method call will result in a fault. The fault may occur immediately, or at a later time if the action results in a subsequent operation not performed at the time of the call.

Each method description in [Chapter 14, “Methods,” on page 171](#) specifies the requirements for performing the specified operation.

IBM Cognos Analytics uses additional criteria to determine whether a specification is considered trusted or not trusted when tasks are performed on objects based on one of the following classes:

- [bibus » reportServiceReportSpecification](#)
- [bibus » reportServiceAnalysisSpecification](#)
- [bibus » reportServiceQuerySpecification](#)
- [bibus » reportServiceMetadataSpecification](#)
- [bibus » reportServiceDrillThroughSpecification](#)
- [bibus » metadataServiceLineageSpecification](#)

A specification is trusted if it is digitally signed or was received through the internal dispatcher entry point:

```
http://server_name:9300/p2pd/servlet/dispatch
```

A specification is not trusted if it is not digitally signed and was received through the external dispatcher entry point:

```
http://server_name:9300/p2pd/servlet/dispatch/ext
```

Requests from SDK applications are not allowed through the external dispatcher entry point. For more information, see [“Connect to IBM Cognos Analytics” on page 45](#). Your applications must connect to IBM Cognos Analytics using the internal dispatcher URI noted.

Depending on the existence of a digital signature, and the way in which the request was received, additional capabilities may be required to perform the operation. This provides an additional level of protection to prevent potentially malicious attacks by running a report that is intentionally written in a harmful way.

Chapter 5. Managing content

Content Manager is a service that maintains and retrieves all IBM Cognos information, including the following:

- content information, such as folders, reports, outputs, and jobs
- configuration information, such as the set of dispatchers and services
- security information, such as namespaces, groups, roles, accounts, and contacts
- external resource information, such as data sources
- administrative information, such as import and export deployments

The content store is a database that stores the information managed by Content Manager. Content is stored as a hierarchical collection of objects, similar to a hierarchy of folders and files. You can manage this content in Content Manager using methods such as the following:

- [content » add\(parentPath, objects, options\)](#)
- [content » query\(searchPath, properties, sortBy, options\)](#)
- [content » update\(objects, options\)](#)
- [content » delete\(objects, options\)](#)
- [content » move\(objects, targetPath, options\)](#)
- [content » copy\(objects, targetPath, options\)](#)

For information about methods, see [Chapter 14, “Methods,” on page 171](#).

Before using the IBM Cognos Software Development Kit, you should be familiar with the content store and how to specify a search path to access the objects in it.

For information about search paths, see [Chapter 32, “Search path syntax,” on page 1565](#).

Data Integrity

Content Manager deals with complex and constantly changing data that must be shared in real time across multiple report and presentation services. To ensure data integrity and performance, all content data is stored in a relational database referred to as the content store.

Internal Consistency

To maintain consistency within the content store, if Content Manager detects an error anywhere in a request, no portion of the request is executed. For example, if a request to add several objects fails because the current user does not have write permission for one of the specified parent objects, no objects are added.

Concurrency Control

Content Manager operates under the assumption that most changes do not interfere with one another. By default, Content Manager handles concurrent transactions by allowing the last update to an object to complete. For example, if two people edit the same report version in IBM Cognos Connection, the last change received by Content Manager will be retained. This approach always allows updates to proceed, but it can result in some changes being lost in the case of simultaneous edits.

If you use the BI Bus API, you can use optimistic concurrency control to prevent users from accidentally overwriting each other's changes. Optimistic concurrency control relies on object versions to detect conflicting transactions.

For example, two users or programs retrieve the properties of an object when the version number is 1 and then they both try to update the properties of the object. The first request succeeds, but the second request fails because the version of the object in the content store is now 2.

When optimistic concurrency control is used, the following requests fail if the value of the [version](#) property changed since the property was last retrieved:

- add requests where the option `updateAction` is set to `update`
- `delete` requests
- `update` requests

To use optimistic concurrency control, you must retrieve the `version` property for the object in a query request, and then include the retrieved value of the `version` property in the request that modifies or updates the object. This ensures that the request fails if the version number of the object in the content store is not the same as the version number passed in that request.

If you do not retrieve the `version` property and include it in the update request, the update succeeds even if the version number changed. Users will not be aware of changes that occurred since they queried the original data.

Classes and Properties

Each object in the content store is an instance of a class.

Each object has properties that are unique to its class. However, some of its properties may be inherited from other classes such as the [bibus » baseClass](#) or [bibus » uiClass](#) classes.

Each class has a purpose. Some classes, such as the [bibus » ancestorInfo](#) class, define properties inherited by other classes. Other classes, such as the [bibus » addOptions](#) class, only define a parameter passed to a method. The classes specified by the [bibus » classEnum](#) enumeration set represent the objects in the content store. Some of these classes inherit properties directly from the [bibus » baseClass](#) class, while others inherit the properties of the [bibus » baseClass](#) class through the [bibus » uiClass](#) class.

The [bibus » baseClass](#) class defines basic properties, such as the name of the object, its class, when it was created and last modified, its ancestors, whether it has children, and the security policies applied to it.

For more information about classes, see [Chapter 15, “Classes,” on page 397](#).

Property Values and Characteristics

A property of a particular object has a value and associated characteristics.

The characteristics of a property are specified by the [bibus » schemaInfo](#) class. The characteristics of a property determine whether the property can be modified or searched, and how Content Manager applies properties to contained objects.

You can update the value of most properties. However, there are some read-only properties that you cannot update. The value of these properties is set by IBM Cognos.

Multilingual Properties

The value of a multilingual property is actually a set of values. Each value is associated with a locale. For example, if you have both English-speaking and French-speaking users, you can use the multilingual `name` property of the [bibus » baseClass](#) class to store an English name and a French name for a report.

General Properties

You can retrieve a general property without having read permission for the object. You need only traverse permission for all ancestors of the object.

Changing the Properties of an Object

Change the properties of an object to control the way it appears and behaves. The properties of an object depend on the type of object. For example, reports have properties that control run options but folders do not.

For information about modifying the access permissions of an object, see [Chapter 4, “Managing security,”](#) on page 45.

To edit the properties of an object, use the [content » update\(objects, options\)](#) method.

Retentions Property

IBM Cognos keeps multiple versions of some classes of objects to maintain a history of changes or events, such as the time a report was last executed and the status of that execution. Only non-interactive tasks such as jobs, schedules, or emailed reports create history objects. For more information, see the [bibus » history](#) class.

To limit the number of versions IBM Cognos keeps, you can set the `retentions` property on the parent that contains the versions. The following table shows parent classes and their versioned child classes.

Parent Class	Versioned Child Classes
bibus » agentState	bibus » agentOutputHotList
bibus » baseAgentDefinition	bibus » history
bibus » baseDataIntegrationTask	bibus » history
bibus » baseReport	bibus » history , bibus » reportVersion
bibus » baseRSSTask	bibus » history
bibus » contentTask	bibus » history
bibus » exportDeployment	bibus » history
bibus » importDeployment	bibus » history
bibus » indexUpdateTask	bibus » history
bibus » jobDefinition	bibus » history
bibus » memo	bibus » history
bibus » package	bibus » model
bibus » planningTask	bibus » history
bibus » storedProcedureTask	bibus » history
bibus » webServiceTask	bibus » history

The `retentions` property contains a set of retention rules, each associated with a specified class of child object that the parent can contain. You can apply the retention rule to instances of each class. For example, a [bibus » baseReport](#) object can have child [bibus » history](#) and [bibus » reportVersion](#) objects, each with a retention rule associated with it. These rules specify values for deleting child objects. Content Manager limits retention by either the number of versions or the duration. For example, you can specify that Content Manager keeps the latest three versions, or keeps versions created for three months.

Relationships Between Objects

All objects are organized in a hierarchy that starts with a single root object.

Objects are related to other objects through either parent-child relationships or references.

These relationships describe the location of objects in the content store, and are reflected in the elements of the search path for each object.

In the following section, objects described as parents and children refer to containment, not inheritance. The parent or containing objects do not necessarily pass on their properties to the child or contained objects. Also, the child or contained objects do not necessarily assume the attributes or characteristics of the parent or containing object.

Parents and Children

Some objects act as containers for other objects. In a parent-child relationship, the container object is the parent and the objects it contains are the children. For example, a `folder` can contain child objects of the following classes:

- [bibus » folder](#)
- [bibus » report](#)
- [bibus » query](#)
- [bibus » reportView](#)
- [bibus » jobDefinition](#)
- [bibus » URL](#)
- [bibus » shortcut](#)

Parent-child containment relationships are used by Content Manager to construct the `searchPath` property of the `baseClass` class.

In the following diagrams, a parent-child containment relationship appears as a solid line between the parent and child classes with the parent appearing above the child. For example, the following diagram shows that a `folder` object may be a parent to one or more `report` or `folder` objects.

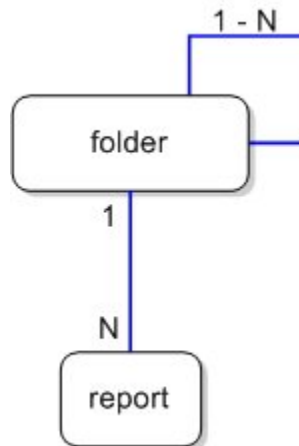


Figure 2. Folder object parent-child relationships

The following diagram shows an example of a folder-to-folder containment relationship in a content store. The folder objects, and any descendant folder objects, can each contain multiple folder and report objects.

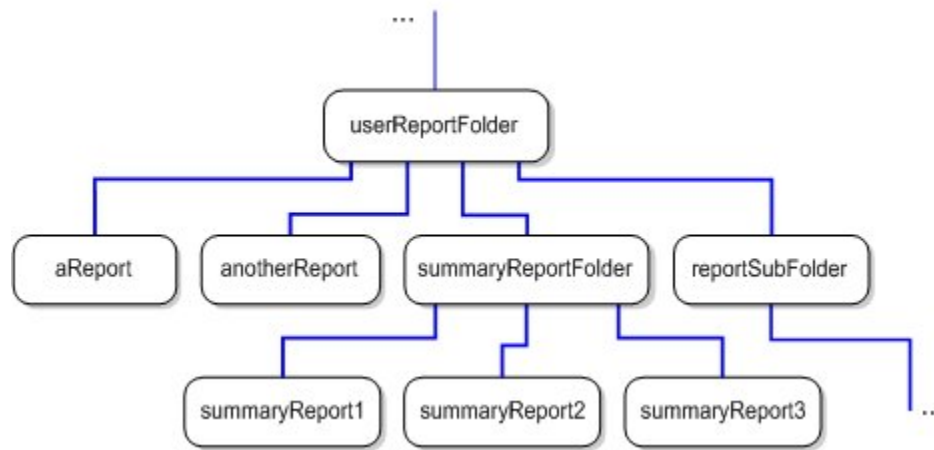


Figure 3. Folder-to-folder containment relationship

The following examples show how to use search path syntax to express a folder-to-folder relationship:

```

.../folder[@name="userReportFolder"]/folder[@name="reportSubFolder"]
.../folder[@name="userReportFolder"]/folder[@name="summaryReportFolder"]
  
```

The folder-to-report relationships shown in the diagram are expressed using similar search path syntax:

```

.../folder[@name="userReportFolder"]/report[@name="aReport"]
.../folder[@name="userReportFolder"]/report[@name="anotherReport"]
.../folder[@name="userReportFolder"]/folder[@name="summaryReportFolder"]/
report[@name="summaryReport1"]
.../folder[@name="userReportFolder"]/folder[@name="summaryReportFolder"]/
report[@name="summaryReport2"]
.../folder[@name="userReportFolder"]/folder[@name="summaryReportFolder"]/
report[@name="summaryReport3"]
  
```

References

Objects may have properties that reference other objects.

For example, the `target` property of a `bibus » shortcut` object references another object, such as a `bibus » report`, `bibus » reportView`, `bibus » query`, or `bibus » URL` object.

In the following diagrams, a reference appears as a solid line between the related classes with an arrowhead to show the direction of the relationship. For example, the following diagram shows that many `bibus » shortcut` objects may reference one `bibus » report` object.

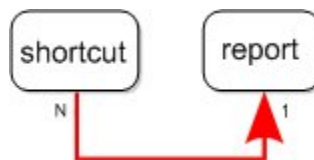


Figure 4. References between shortcut objects and reference object

For general information about search paths, or about the syntax and grammar of search paths, see [“Search Path Syntax”](#) on page 1567.

Querying the Content Store

Use the `content » query(searchPath, properties, sortBy, options)` method to retrieve objects from the content store. When you use the `content » query(searchPath, properties, sortBy, options)` method, you must specify the appropriate search path to identify the objects you want to retrieve. For information about search paths, see Chapter 32, “Search path syntax,” on page 1565.

By default, only the `searchPath` property for each object is returned. To retrieve other properties, specify them in the `properties` parameter.

It may also be necessary to specify options for the query. For example, to retrieve the `bibus » schemaInfo` details for an object, you must set the `schemaInfo` property of the `bibus » queryOptions` object to `true`.

Example: Querying the Content Store in Java

To see the code in context, view the sample in the following location:

`installation_location/sdk/java/Security/Logon.java`

This Java code sample demonstrates how to query the content store:

```
PropEnum props[] =
    new PropEnum[] {PropEnum.searchPath, PropEnum.defaultName,
PropEnum.policies,
        PropEnum.userName, PropEnum.notificationEMail };
Account myAccount = null;
...
BaseClass bc[] =
    connection.getCMService().query(
        new SearchPathMultipleObject("~"), props, new Sort[] {}, new
QueryOptions());
    if ((bc != null) && (bc.length > 0))
    {
        for (int i = 0; i < bc.length; i++)
        {
            myAccount = (Account)bc[i];
        }
    }
...
    output =
        output.concat(
            "Your alert email address is: "
            + myAccount.getNotificationEMail().getValue());
```

Steps

1. Create a `bibus » searchPathMultipleObject` object specifying the objects to retrieve.
2. Create a `bibus » sort` object. You can use an empty object if the default sort order is acceptable.
3. Create an array of properties from the `bibus » propEnum` enumeration set.
4. Create a `bibus » queryOptions` object and set its properties. You can use an empty object if no options are required for the query.
5. Call the `content » query(searchPath, properties, sortBy, options)` method.
6. Access the properties for each individual object returned in the array of `baseClass` objects.

Organizing Objects

Organize objects in a meaningful way so that users can easily find reports, files, or Web sites in the portal.

You may decide to create a folder hierarchy by using nested folders to group objects by type or by frequency of use. The folder structure should be logical and should support the chosen method of grouping.

Use meaningful names and detailed descriptions to identify objects.

Creating a Folder Object

Use folders to organize objects, such as reports, URLs, groups, and roles. You can organize reports by creating a folder hierarchy.

To create a folder, use the [content » add\(parentPath, objects, options\)](#) method with the [bibus » folder](#) class.

Creating a URL Object

Create a URL for the files and Web sites you use most frequently. You click a URL in the portal to open the file or Web site in the browser. After opening a URL, click the back button in your browser to return to the portal.

To create a URL, use the [content » add\(parentPath, objects, options\)](#) method with the [bibus » URL](#) class.

Creating a Shortcut Object

Create a shortcut to access objects from another location in the portal. A [bibus » shortcut](#) object can refer to a [bibus » URL](#), [bibus » baseAgentDefinition](#), [bibus » basePowerPlayClass](#), [bibus » pagelet](#), [bibus » query](#), [bibus » folder](#), [bibus » jobDefinition](#), [bibus » report](#), or [bibus » reportView](#) object.

To create a shortcut, specify a [bibus » shortcut](#) object for the [objects](#) parameter of the [content » add\(parentPath, objects, options\)](#) method.

You can change access permissions for a shortcut object, but the access permissions for the target object do not change.

If you want to use a generic report as the underlying structure for additional reports, make a copy of the report instead of creating a shortcut.

If you want to keep the underlying report specification of a report, but want the choice of another report format, language, delivery method, or run option, create a report view. For more information, see [“Creating Report Views” on page 73](#).

Copying an Object

Copy an object to create a replica of that object in another location. When you make changes to a source object in one folder, those changes are not reflected in copies of the object in other folders.

To copy an object, use the [content » copy\(objects, targetPath, options\)](#) method.

Moving an Object

Move an object to remove it from the current path and place it in another path or folder.

You may decide to move an object if a folder becomes so full that it is difficult to locate particular objects.

To move an object, use the [content » move\(objects, targetPath, options\)](#) method.

Controlling the Visibility of Objects

Controlling the visibility of objects is not a security measure; it is a convenient means for limiting UI clutter and avoiding the inadvertent use of objects that users might not understand fully. This ability to control object visibility is not supported for objects that do not use the IBM Cognos namespace.

Use the searchable, calculated [shown](#) property of [bibus » uiClass](#) objects to determine whether objects should be visible to users. For example, the query predicate, `[@shown=true]`, filters for visible objects.

Inputs to the algorithm that calculates `shown` are:

- the `hidden` property of the `bibus » uiClass` class
- the `bibus » portalOptionEnum » showHiddenObjects` value of the `bibus » portalOptionEnum` enumeration set
- the `bibus » userCapabilityEnum » canUseShowHiddenObjectsPreference` value of the `bibus » userCapabilityEnum` enumeration set
- the `owner` of the objects

The following table shows how the algorithm that calculates the `shown` property combines its inputs to produce values. The inputs are listed in the first four columns. The dashes indicate values that can be either true or false without affecting the value of `shown`.

Table 8. shown property calculations

<code>hidden</code>	<code>showHiddenObjects</code>	<code>canUseShowHiddenObjectsPreference</code>	<code>object owner ?</code>	<code>shown</code>
true	true	true	-	true
true	true	false	true	true
true	true	false	false	false
true	false	-	-	false
true	-	-	-	true

Implications of Shortcuts and Other Object Reference Mechanisms

Shortcut objects provide a mechanism to access other objects and are not automatically hidden when the object they reference is hidden. Since a `bibus » shortcut` is a `bibus » uiClass` object, each `bibus » shortcut` has its own `shown` property. For example, a `bibus » shortcut` to a hidden folder allows users to navigate to the folder and see any non-hidden content.

Model-based drill-through definitions that refer to hidden drill targets are not implicitly hidden. This permits administrators to prevent users from seeing target reports except when performing a drill through.

When allowing the report author to specify a drill-through target report at authoring time, the authoring application should honor the `shown` property. The Drill Through Definitions pages in IBM Cognos Connection and IBM Cognos Analytics - Reporting provide this functionality.

When viewing job steps or agent tasks, if the `hidden` property for an existing job step or agent task is true then the user interface displays the corresponding icon as transparent. The icon also appears as a `bibus » shortcut` in the portal. When users add job steps or agent tasks, use the `shown` property to determine how the icon displays.

There are special considerations when consuming references to hidden objects. For example, when the list of authored drill-through targets is shown in the IBM Cognos Viewer right-click menu or in the Go To page, all targets are displayed regardless of the `shown` property of the target report. This is the same behavior as for a non-hidden shortcut in the portal. The display of model-based drill-through targets should be determined by the value of the drill-through definition's `shown` property, but should not use the `shown` property value of the target object.

Implications of Authority and Ownership

Users with sufficient authority can alter other users' preferences and the properties of their objects with sometimes unexpected results. For example, an administrator can set a user's `showHiddenObjects`

preference to `false`. If the user hides an object (sets its `hidden` property to `true`), the object will seem to disappear if the application honors the value of the object's `shown` property.

Disabling an Object

Disable an object to prevent users from accessing it.

Disabling objects is useful when you want to perform maintenance operations. For example, you can disable a folder while you reorganize its content. When you disable a folder, its contents are also disabled.

If an object is disabled, all shortcuts to this object show the disabled icon in the portal.

You must have read and write privileges to an object to enable or disable it.

To disable an object, set the `disabled` property to `true`, and then use the `content » update(objects, options)` method.

Deleting an Object

Delete an object to permanently remove it from the content store. You may decide to delete an object because it is outdated or no longer satisfies your requirements.

Deleting the source object for a shortcut removes only the source object. The shortcut objects remain but have an invalid reference icon in the portal and are not accessible.

Deleting a shortcut or a report view removes only the selected object and not the source object.

You must have read and write permissions for the object you are attempting to delete. You must also have write and traverse permissions for the current folder.

To delete an object, use the `content » delete(objects, options)` method.

For more information about deleting objects, see “Remove Dispatchers from the Environment” on page 87.

Setting the Order of Objects

Set the order of objects to change the way folders and objects are sorted in the portal. You cannot set the order for objects on multiple folders simultaneously. You may decide to organize objects by level of usage and place objects that you use daily at the top of the list.

By default, objects are sorted alphabetically. Objects added after the order is specified are shown at the end of the list in the default order.

To set the order, you must have read and write permissions for all objects in the folder and read and traverse permissions for the folder containing the objects.

To specify the order of objects on query, set the `order` property of the `bibus » sort` class, and then use the `content » update(objects, options)` method.

Managing User Accounts

Directory administrators can delete or copy user accounts in Content Manager using the `content » deleteAccount(objectPath, options)` method and `content » copyAccount(sourceAccountPath, targetAccountPath, options)` method respectively.

Deleting a User Account

Delete an account to remove it permanently from the content store. Deleting an account removes the account and any descendents in the namespace.

If the user whose account was deleted logs on, Content Manager creates a new account with original defaults.

If the user whose account was deleted is logged on when the account is deleted, Content Manager invalidates the user's passport before deleting the account.

You must have `execute` permissions for the Directory secured function to delete an account.

To delete a user account, use the `content » deleteAccount(objectPath, options)` method.

Copying a User Account

Copy a user account to duplicate information from an existing source account to either a sub-folder of an existing target account or to a new target account.

Copying the source account does not copy credentials or session objects. Also, Content Manager copies My Folders items from the source account to a new folder in the target account.

Content Manager changes the owner property of any object owned by the source account to refer to the target account, but does not modify or repair any other references to the source account.

You must have `execute` permissions for the Directory secured function to copy an account.

To copy a user account, use the `content » copyAccount(sourceAccountPath, targetAccountPath, options)` method.

Maintain the Content Store

Use the `bibus » contentTask` class to define content store maintenance tasks, such as verifying the consistency of security data.

To run `bibus » contentTask` objects, use the `asynch » run(objectPath, parameterValues, options)` method with the `monitorService`. You can specify which task is performed by setting the options when the object is run. Content tasks can be run as part of an agent or job, or can be scheduled to run independently.

To run a `bibus » contentTask` object, you must log on as a user who has read access to all objects in the security system. If you want to schedule a `bibus » contentTask` object to run later, you must include a reference to an existing `bibus » credential` object with the necessary permissions, including read access to any security systems that the `bibus » contentTask` object may require.

For the `objectPath` input parameter of the `asynch » run(objectPath, parameterValues, options)` method, provide the path to the `bibus » contentTask` object in the `bibus » adminFolder` object.

Use the `checkSecurityReferences` option of the `bibus » contentTaskOptionEnum` enumeration set to check the consistency of security data in the content store with the defining security provider. Inconsistencies can occur if accounts, groups, or roles used with IBM Cognos are subsequently removed from the security system. These inconsistencies are reported in the `detail` property of the `bibus » baseHistoryDetail` class.

Use the `repairSecurityReferences` option to repair inconsistencies in the security data by removing data from the content store that is no longer defined by the security provider. Removing this information from the content store can recover space in the store and improve performance.

To identify the namespaces to check or repair, use the `securityReferencesContext` option with both the `checkSecurityReferences` and `repairSecurityReferences` options. Ensure that the program logs on as a user that has access to all the namespaces in question. If no namespace is specified, the task runs on all namespaces the user is logged on to.

If no options are specified, the default options of the `bibus » contentTask` class are used.

To specify that objects of certain classes in a deployment should be upgraded when they are imported, use the `upgradeClasses` option.

Chapter 6. Running tasks

IBM Cognos consists of a service-based architecture that groups functional areas into discrete web services. These services use a common API. Each service manages specific runnable objects (objects that can be executed) according to the functional area the service is responsible for. For example, the `reportService` handles report-specific tasks and has a number of runnable objects associated with it. These objects include queries, reports, and report views.

A common service API provides a consistent, generic interface for running tasks with the following key features:

- A standardized API is shared by various services and provides a consistent mechanism for performing long-running, server-based tasks.
- A shared options model provides a mechanism for specifying options to satisfy a request based on a common definition used by different services. This provides the ability to specify options for multiple tasks in a single list.
- A common history model implements a shared history class hierarchy that can contain additional types of details provided by various services.

For more information about the individual services available, see [“IBM Cognos Analytics Services” on page 5](#).

Using `run()`

When you use the `asynch » run(objectPath, parameterValues, options)` method to execute an object, you call the service that implements that method directly. For example, if you want to run a report, you use the `asynch » run(objectPath, parameterValues, options)` method implemented by `reportService` to create a request to run the task.

A request for a specific task is created by an IBM Cognos client. Requests can originate from the Web portal, as a result of an object's schedule firing, or through the BI Bus API.

To satisfy a request to perform a task, one or more services may be called upon. For example, if running a report requires sending the output of the report as an email attachment to a list of recipients, the `reportService` runs the report and interacts with the `deliveryService` to email the report.

Because multiple services can interact to satisfy a request, you can provide a list of `options` as input parameters that apply to each service.

In the example shown in the following figure, a client application calls the `asynch » run(objectPath, parameterValues, options)` method to run a job. One of the job steps runs a report, and sends the report output in an email attachment to a list of specified recipients.

In this example, the `jobService`, `reportService`, and the `deliveryService` all interact to satisfy the request. The `asynch » run(objectPath, parameterValues, options)` method call to the `jobService` contains all the options to satisfy the request. The following enumerations specify the available options for the runnable objects:

- `bibus » asynchOptionEnum` enumeration set is common to all services that implement the `asynch` method set and define options for the asynchronous conversation initiated to process the request
- `bibus » runOptionEnum` enumeration set defines options specific to the report, such as whether to burst the report, the output format, or the output language of the report
- `bibus » deliveryOptionEnum` enumeration set defines options specific to the delivery service responsible for emailing the report, such as the recipient list or the subject line of the email

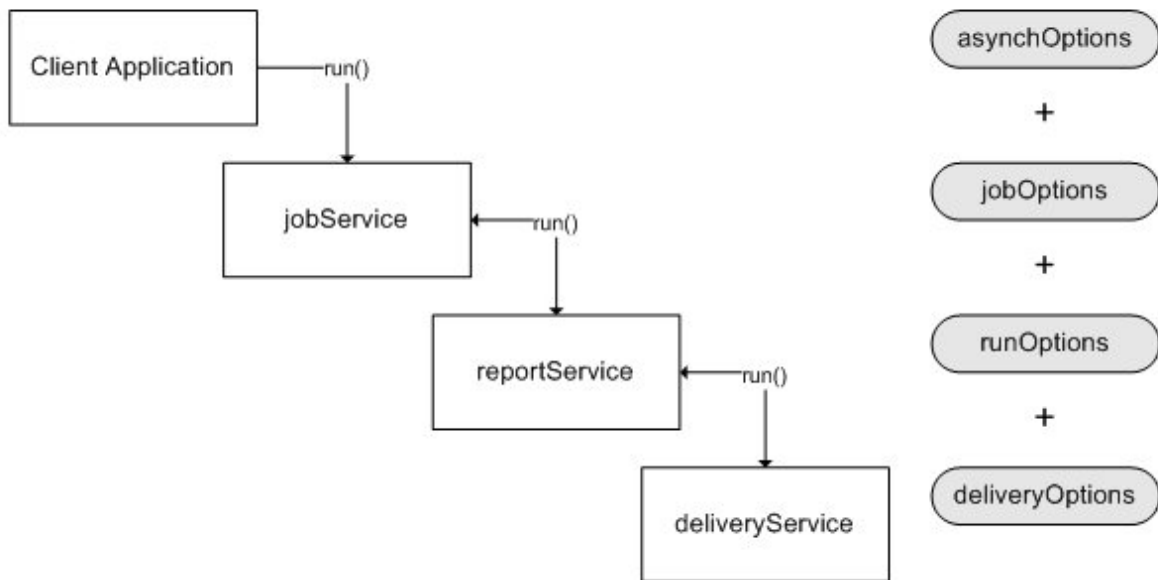


Figure 5. Execution of a run method to run a job

Services that interact to perform a task or set of tasks obtain options from the original request, or from other locations. For more information about how options are determined, see [“Specifying Options and Parameters”](#) on page 66. Options that are not relevant to a specific service are ignored by that service.

How IBM Cognos Builds a Request

IBM Cognos builds a request to run an object in one of the following three ways:

- a client interaction, such as a method call

When you invoke a method that supports options (such as `asynch » run(objectPath, parameterValues, options)`, `event » runAt(startTime, objectPath, parameterValues, options)`, or `asynch » runSpecification(specification, parameterValues, options)`), the service builds the request based on the rules defined in [“Running an Object Using a Method Call”](#) on page 68 and sends the request to the appropriate service for processing.

- an object's schedule

The `eventManagementService` manages `bibus » schedule` objects in the content store. When an object runs based on its schedule, the `eventManagementService` builds the request to run the object based on the rules defined in [“Running an Object Using the Schedule”](#) on page 68.

- another request

For both mechanisms of building a request, it is possible that the service called to satisfy the request might also build an additional request to another service. For example, when a report runs, the `reportService` can send a request to the `deliveryService` to send the report as an email attachment to a list of recipients.

Running a job can include one or more job steps that call other services.

Specifying Options and Parameters

A unified class hierarchy based on the `bibus » option` class provides a common mechanism for changing default behavior when running objects. This allows you to specify options based on a common definition that is understood by different services, depending on the task. Option values for multiple tasks are specified in a single list contained within the initial request.

An option is a way of changing default behavior when running an object based on predefined choices. For example, you can specify that the output format of a report be PDF, overriding the default of HTML, by setting the `bibus » runOptionEnum » outputFormat` value. Sibling classes derived from class `bibus » option` are implemented by each service for the runnable objects for the service.

The inheritance hierarchy of the `bibus » option` class family is shown in the following figure.

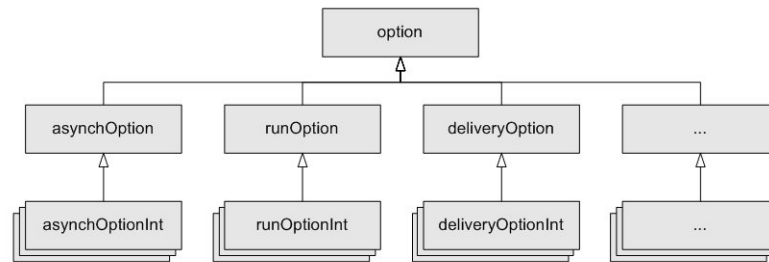


Figure 6. Inheritance hierarchy for the `option` family

For example, if a `bibus » jobDefinition` contains a `bibus » report`, a `bibus » contentTask`, and a `bibus » deploymentDetail`, you can pass options for all three tasks in the `asynch » run(objectPath, parameterValues, options)` method call for the job. When the request is processed, each service evaluates the options specific only to that service (for example, the `deliveryService` service only recognizes options of the `bibus » deliveryOption` class).

A `bibus » parameter` is user-defined (specified by a report author) and its value filters data for a specified object. For example, using a parameter named `CountryCode` in a filter expression filters data based on the country of origin. You can also specify parameter values for multiple tasks in a single list when making a request to run an object.

Ways of Specifying Options and Parameter Values

When IBM Cognos builds a request, it derives options and parameter values from

- input parameters to a method call
- the values of an object's properties
- user preferences (for the user associated with the `bibus » schedule` or the current `bibus » session`)
- system defaults

User preferences are defined in instances of the `bibus » account` class and are usually editable through the BI Bus API or the user interface. You cannot change system defaults.

Specifying options and parameters as input parameters is a way of overriding the default behavior of an object. The behavior of an object is defined by the object's properties or an ancestor of the specified object, or by user preferences and system defaults.

You can set run options and parameter values as properties on most objects in the content store associated with a task. For example, a `bibus » reportVersion`, `bibus » schedule`, or `bibus » contentTask` all have properties for specifying options and parameter values.

You can set specific options as input parameters to methods depending on which object is being executed. If a method call to a service calls additional services, you can provide options appropriate to any of the services and the objects they manage. For example, when you use the `asynch » run(objectPath, parameterValues, options)` method to run a job (`bibus » jobDefinition` class), one of the job steps could be to run a report. You can configure the report to send an email when it is run. In this case, the `jobService`, `reportService`, and `deliveryService` all participate in satisfying the request. You can specify options for the job (`bibus » asynchOption` class), the report (`bibus » runOption` class), and the email (`bibus » deliveryOption` class) as input parameters to the method call.

How IBM Cognos Determines Search Order When Building a Request

IBM Cognos uses a search order to determine which settings (options and parameter values) are applied when a request is initiated to run an object. You can specify different settings in different places to dictate different behaviors based on how an object is run. For example, you can schedule a report with one set of run options, but use a different set of options when running the report interactively.

Because options and parameter values can be specified in various places, a predefined search order exists for determining where to obtain the options and parameter values used to build the request that runs the object.

Running an Object Using a Method Call

Supplying options and parameter values as input parameters to a method provides you with a way to change how an object is run.

When you use a method call, the first instance of a run option or parameter is used where the search order is:

1. input parameters to the method call
2. the properties of the object in the content store being run, if the object exists in the content store
3. user preferences for the user or the current session
4. service defaults set in the [bibus » configuration » serviceDefaultOptions](#) property (for some services)
5. system defaults

Note: The [asynch » runSpecification\(specification, parameterValues, options\)](#) method builds a report based on a specification supplied by the user. Options and parameter values are not derived from an object in the content store in this case.

Note: Administrators can constrain option values using the [bibus » configuration » overrideOptions](#) property. For more information, see [“Constraining Option Values and Setting Service Default Values” on page 69](#)

What's new

New in Version 10.1.0 – “Accessibility” on page 1859

This topic was updated.

Running an Object Using the Schedule

Specifying options and parameter values as part of the schedule of an object provides you with a way to change how an object is run in the future.

When an object runs based on its [bibus » schedule](#), the first instance of an option or parameter is used where the search order is

1. the [bibus » schedule](#) of the object
2. the properties of the object in the content store being run
3. user preferences for the user associated with the schedule or the current session
4. service defaults set in the [bibus » configuration » serviceDefaultOptions](#) property (for some services)
5. system defaults

Note: Administrators can constrain option values using the [bibus » configuration » overrideOptions](#) property. For more information, see [“Constraining Option Values and Setting Service Default Values” on page 69](#)

What's new

New in Version 10.1.0 – “Accessibility” on page 1859

This topic was updated.

Setting Common Options

Many of the options you can specify in the BI Bus API may be implemented by multiple services. For example, [bibus » asynchOptionEnum](#) options are implemented by multiple services that support the [asynch](#) method set. Options that are specific to a [bibus » baseReport](#) object can be set in other objects that may deal with reports in some way, such as in a [bibus » jobStepDefinition](#) object.

Setting PDF Options

You can specify options to control the ability to alter or manipulate PDF documents created by IBM Cognos. These options can be specified when running reports and allow for more granular control of accessibility options and print quality, and enforce control of these options using password protection on a PDF document.

Use the [bibus » pdfOptionEnum](#) enumeration set to specify options anywhere other options are specified. For example, PDF options can be specified for a [bibus » baseReport](#) object, a [bibus » jobDefinition](#) object, a [bibus » jobStepDefinition](#) object, an [bibus » agentDefinition](#) object, or an [bibus » agentTaskDefinition](#) object.

Specifying PDF options has no effect if the value specified for [bibus » runOptionEnum » outputFormat](#) value is not [PDF](#). If a value is not specified for [bibus » pdfOptionEnum » ownerPassword](#) value, any of the values specified for PDF options can be overridden by a consuming application.

Security settings for PDF report output can be set by anyone with [write](#) access to the [bibus » baseReport](#) object. Encryption strength and key sizes for PDF documents are set using IBM Cognos Configuration.

A user running a report who does not have [write](#) access for the report cannot set PDF options if the report output is persisted in the content store.

Constraining Option Values and Setting Service Default Values

The [bibus » configuration » overrideOptions](#) property allows you to constrain which option values can be used when an request is run, for some options. If you want to limit option values for all requests, specify the allowed values on the [bibus » configuration » overrideOptions](#) property.

If you constrain the option values in the [bibus » configuration » overrideOptions](#) property, you may need to add one of these values to the [bibus » configuration » serviceDefaultOptions](#) property to ensure that a valid value is provided as a default.

The [bibus » configuration » serviceDefaultOptions](#) property allows you to set a default option value for some services. The value that you set in this property effectively changes the system default value. Use this property to set a new service default option value when

- you have constrained option values using the [bibus » configuration » overrideOptions](#) property, and
- the system default value is not included in the list of values that you have specified on the [bibus » configuration » overrideOptions](#) property

For example, if you set the [accessibilityFeatures](#) option value on the [overrideOptions](#) property, to [true](#), you must also set the [accessibilityFeatures](#) on the [bibus » configuration » serviceDefaultOptions](#) property option to [true](#) because the default value is [false](#). If you set both of these option values to [true](#), the server will generate all reports with accessibility features enabled.

What's new

New in Version 10.1.0 – “Accessibility” on page 1859

This topic was updated.

Managing Event History

The BI Bus API employs a generic event history model that provides a single method for recording history for all services, avoiding variations in the information stored. This model is based on a shared [bibus » history](#) class hierarchy that can contain additional types of details provided by various services, if required.

You can control the quantity of [bibus » history](#) records that the [monitorService](#) generates by specifying the [bibus » monitorOptionEnum » writeCompleteHistory](#) value on a job request. Setting it to `false` causes a [bibus » history](#) record for only the root task to be recorded if all the tasks executed by the job complete successfully. Setting it to `true`, or omitting it, causes a complete history of all invoked tasks to be recorded.

If a job error occurs, complete task history is recorded for the job, regardless of the value of [writeCompleteHistory](#).

Setting this option on a job request overrides the stored option values for root tasks at runtime.

The [monitorService](#) oversees scheduled tasks executed by the [eventManagementService](#) service and manages history objects for those tasks in the content store. The [monitorService](#) creates a [bibus » history](#) object whenever a scheduled task runs and maintains additional details about the task for failover purposes. A history object is created whenever an agent, agent view, or job is run. A history object is also created for any task that is run on a scheduled basis, or for a report that is emailed, printed or saved, or is run in response to an external event.

You can use the [asynch » run\(objectPath, parameterValues, options\)](#) method as implemented by the [monitorService](#) directly if you want to create your own [bibus » history](#) objects when running a non-scheduled task. Note that this can create additional overhead for your application and may impact performance.

The [jobService](#) service and [agentService](#) service always use the [monitorService](#) service to maintain history information even when running a non-scheduled task.

The [monitor » background\(conversation\)](#) method is implemented by the [monitorService](#) service to allow you to relinquish control of an asynchronous conversation and allow it to run in the background. It is also possible to run a task in the background by calling the [monitorService](#) directly, specifying the [bibus » monitorOptionEnum » background](#) value.

The [bibus » eventRecord](#) for a task includes information such as the event ID, request time, status, and a reference ([bibus » eventRecord » history](#) property) to the history object created by the [monitorService](#) when the task was executed.

To view the run history for an object, use the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

You must have read and write permissions for the object, and read or traverse permissions for the containing object.

Scheduling Tasks

You can create a schedule to run an object, such as reports, jobs, deployments, or agents, at a recurring date and time. You can schedule objects to run by the minute, hourly, daily, weekly, monthly, or yearly.

You must have read and write permissions on the source object to create or otherwise manage a schedule. Only one schedule can be associated with an object. For reports, you can create report views and then create a schedule for each report view if you require multiple schedules for the same report. A report or other object can also be associated with another schedule by being part of a job, which has its own schedule. If you no longer need a schedule, you can disable it without losing any of the scheduling details. You can then enable the schedule again at a later time. For more information, see the [bibus » schedule » active](#) property.

Note: Setting a task to run more often than the underlying data is refreshed can create unnecessary load on the system by performing redundant operations. For example, if a cube is only refreshed on a weekly basis, scheduling reports based on the cube data on a daily basis provides no added benefit.

Viewing and Managing an Object's Schedule

After you create a [bibus » schedule](#) for an object, you can view the status and the request, start, and completion time of all background executions, and filter scheduled tasks based on user, dispatcher, or service. You can also manage schedules by modifying or canceling them, and assign priorities on scheduled tasks. You must have read and write permissions for the object.

To add one or more new schedules, use the [content » add\(parentPath, objects, options\)](#) method.

To view schedules or to view information about pending events currently held in queue, call the [eventManagementService](#) service using the [runSpecification\(specification, parameterValues, options\)](#) method.

To modify schedules, use the [content » update\(objects, options\)](#) method.

To set the priority for a scheduled task, use the [event » updateEvents\(events\)](#) method.

To run a scheduled object once, independently of the schedule associated with the object, use the [asynch » run\(objectPath, parameterValues, options\)](#) method. Use the [event » runAt\(startTime, objectPath, parameterValues, options\)](#) method to execute an object at a specified time. You can use the options and parameters associated with the object, or you can modify them for this run only by providing them as input parameters to these methods. For more information, see “[Specifying Options and Parameters](#)” on page 66.

Using Intraday Scheduling

Intraday scheduling lets users schedule tasks to run on a specified day, week, month, or year, and run multiple occurrences on the specified day at a specified interval. For example, a task scheduled to run on the first Monday of each month can be set to run hourly on that day in a specified recurrence window (for example, between 6 AM and 5 PM).

This type of scheduling benefits users who require more options than those provided by normal scheduling and can be invaluable to administrators managing server load. For example, if a warehouse requires inventory reports during the work day period when employees are present, an intraday schedule can be configured to run a report from Monday to Friday hourly between 8 AM and 5 PM. This provides employees with the most up-to-date information on demand, and allows administrators to manage a more predictable load than if interactive reports were run on an as needed basis. It also removes the load created by running the same report after hours when it is not needed, allowing other load-intensive system maintenance tasks to be run.

Administrators can restrict which types of intraday scheduling options are available to users and groups with the following capabilities:

- [canUseSchedulingByDay](#)
- [canUseSchedulingByHour](#)
- [canUseSchedulingByMinute](#)
- [canUseSchedulingByMonth](#)
- [canUseSchedulingByWeek](#)
- [canUseSchedulingByYear](#)

You can set up intraday scheduling by setting the [bibus » schedule » type](#) property to one of the intraday values defined in the [bibus » scheduleTypeEnum](#) enumeration set. Refer to the information provided for each of the values to determine what additional properties need to be set on the [bibus » schedule](#) instance.

Creating a Trigger-based Schedule

You can use the BI Bus API to define schedules that are run based on external occurrences.

Use the `event » trigger(triggerName)` method to advise IBM Cognos of an external occurrence. IBM Cognos queues tasks associated with schedules that share the same value for the `triggerName` property for execution.

Steps

1. Create an instance of the `bibus » schedule` class.
2. Set the `type` property to the value `trigger`.
3. Set the `triggerName` property to the name of the external occurrence.
4. Add the schedule to the object.

Administrators can restrict whether trigger-based scheduling is available to users and groups with the `canUseSchedulingByTrigger` capability.

Retrying Tasks

If a task fails, you can retry it based on the `bibus » history` object for the failed run. The retry will use the same parameter values and options as the original request, and will continue from the point of failure. If the task failed due to a database or network error, you can retry it without any changes once the issue is resolved. If a particular sub-task caused the task to fail, and you wish to try the run without that step, you can specify that it is to be skipped during the retry.

To retry a task, use the `asynch » run(objectPath, parameterValues, options)` method with the monitoring service. Specify the location of the `bibus » history` object for the failed run in a `restartHistoryLocation` option. Specify any steps to be skipped during the retry in a `skipTaskHistoryLocations` option.

Running Service-based Tasks

The BI Bus API includes a shared, asynchronous method set (the `asynch` method set), implemented by a number of services. This provides a common mechanism for executing objects that allows you to

- run a report, query, analysis, or report view
- run a specification to create a report output
- run an agent or a job
- perform a content management task
- import or export deployments
- send an email
- manage RSS channels

Running all these tasks using the `asynch` method set means that the basic operation of executing runnable objects are the same. For more information, see [“Using run\(\)” on page 65](#).

As each service is responsible for running discrete tasks, additional methods are available for performing specialized operations.

Running Reports

You can run a report interactively or non-interactively.

In interactive mode, you run the report immediately, independent of the report's schedule. You can

- run a single report using the `asynch » run(objectPath, parameterValues, options)` method
- run a report specification using the `asynch » runSpecification(specification, parameterValues, options)` method

In non-interactive mode, you create a schedule for a report or a job that includes the execution of a report as one of its steps to run at a future time. You can

- run multiple reports by creating a `bibus » jobDefinition` and creating a schedule for it

- schedule one or more reports to run at regular intervals
- run a report or a job at a later time using the [event](#) » [runAt\(startTime, objectPath, parameterValues, options\)](#) method

In all cases, you must have execute permissions for the object (for example, a report or job) and read and traverse permissions for the folder that contains the object.

The report output is automatically saved if you schedule a report, select multiple report formats or languages, or burst a specific report. You can specify how long to keep the report outputs for reports by modifying the [bibus](#) » [baseReport](#) » [retentions](#) property.

For more information about using schedules, see [“Scheduling Tasks” on page 70](#).

Creating Report Views

A report view ([bibus](#) » [reportView](#) class) is a reference to a report. It shares the same report specification as the source report, but you can specify different prompt values, schedules, options, or parameter values than the report from which it is based.

Creating a report view does not change the original report. You can determine the source report for a report view by viewing its [base](#) property. The report view properties also provide a link to the properties of the source report.

Before you can create a report view, you must have read privileges to the source report.

To create a report view, use the [report](#) » [add\(parentPath, object, options\)](#) method.

If you want to use a report as the underlying structure for additional reports, make a copy of the report instead of a report view.

If you want a report to appear in more than one location, create a shortcut. For more information, see [“Creating a Shortcut Object” on page 61](#).

Setting Default Prompt Values

Report authors can narrow the range of data in a report by creating prompts. Specifying default parameter values for a report eliminates the need for a user or application to provide prompt values when the report runs.

To specify default parameter values, use the [report](#) » [update\(object, options\)](#) method.

If a report that contains prompts is scheduled, you must save the prompt values or specify default values to ensure that values exist when the report runs according to the schedule. For more information, see [“Specifying Options and Parameters” on page 66](#).

To set default parameter values, you must have read and write permissions for the report, and read or traverse permissions for the folder that contains the report.

The prompts must already be defined before you can specify default values. For information about adding prompts or changing prompt definitions, see the *IBM Cognos Analytics - Reporting User Guide* and the *IBM Cognos Query Studio User Guide*.

Caching Prompt Data

System performance can be improved by caching prompt data for reports that have several of the following characteristics:

- They are used frequently in interactive environments.
- They contain a large number of non-cascaded prompts.
- They contain prompts with a large number of candidate values.
- They contain prompts that require complex queries to obtain the set of candidate values.

The report cache is used to construct prompt pages rather than obtaining the prompt values from the data source every time a report is executed interactively. When constructing prompt pages for a [bibus](#) »

`reportView` object, the IBM Cognos server attempts to use the cache saved in the `bibus » reportView` object. If the cache does not contain the necessary data, the server's attempts to use the cache saved in the `bibus » authoredReport` upon which the `bibus » reportView` is based. If the cache does not contain the data necessary to construct a prompt page, prompt data is obtained from the data source.

Applications can control when a prompt cache is built by using the `promptCacheMode` run option. The following describes how option values are used:

- `none` - disables prompt cache processing. This is the default.
- `create` - specifies that the report's prompt cache should be created with entries for locales specified by the `outputLocale` run option. If a report's prompt cache already exists, it is replaced.
- `refresh` - specifies that the report's prompt cache entries for locales specified by the `outputLocale` run option should be refreshed. If an existing prompt cache entry for a specified locale has expired, it is replaced. If a report's prompt cache entry for a specified locale does not exist, it is created. The read-only property `expirationTime` is used to determine whether the cache entry for a locale has expired.
- `update` - creates a prompt cache if necessary and then adds prompt cache entries for the locales specified by the `outputLocale` run option. Matching prompt cache entries are replaced. Prompt cache entries for other locales are not affected.

The cache is built when the `bibus » baseReport` object is run using the `asynch » run(objectPath, parameterValues, options)` method specifying the `promptCacheMode` run option with a value of `create` or `update`. The user running the `bibus » baseReport` object must have write access to the object to store the cache data in the content store. All parameter values are ignored during the construction of the report cache.

The report cache is locale specific. To construct a report cache to support multiple locales, use the `outputLocale` run option to specify the locales for which the report cache should be constructed. Both the language and country specified for `locale` must match those specified when a report is run. If a match occurs, the cached data is used to render the prompt controls. If a match does not occur, the cached data is not used and prompt data is retrieved from the data source. If the locale for a cache entry has only the language specified, then it will match any locale with that language regardless of the country.

Report caches will typically be updated by running reports under the control of a scheduled `bibus » baseAgentDefinition` or `bibus » jobDefinition` object using the `promptCacheMode =create` run option or when running reports interactively with `promptCacheMode =refresh`.

An instance of the `bibus » reportCache` class stored as a child of a `bibus » baseReport` object is subject to Content Manager retention rules, although the Content Manager applies retention rules only when objects are added. You can use the `expirationTime` property of report objects to filter for report objects that have not expired. The following query demonstrates this by comparing the `expirationTime` of a report object's cache to an arbitrary time:

```
<report search path>/reportCache[@expirationTime < '<time-value>']
```

This query retrieves report objects where they have not expired with respect to the given time.

Report cache data is not included when importing or exporting a deployment.

Delivering Reports

You can deliver reports by saving them in the Public Folders area of IBM Cognos Connection or in a file system, sending them by email or to a mobile device, or printing them.

You can specify these options in the run options of the primary request that runs a report or report specification. Alternatively, you can use the secondary `report » deliver(conversation, parameterValues, options)` method to deliver a report or report specification after the primary request has completed. This is not as efficient as including the delivery options in the primary request.

The following table shows, for each delivery mode, which methods can use that mode, and the associated run options.

Table 9. Methods and run options for delivery modes

Delivery mode	Methods	Run options
Save a report to a file system	asynch » run(objectPath, parameterValues, options) report » deliver(conversation, parameterValues, options)	archive
Save a report to public folders	asynch » run(objectPath, parameterValues, options) report » deliver(conversation, parameterValues, options)	saveOutput or saveAs
Send a report by email	asynch » run(objectPath, parameterValues, options) asynch » runSpecification(specification, parameterValues, options) report » deliver(conversation, parameterValues, options)	email
Send a report to a mobile device	asynch » run(objectPath, parameterValues, options) asynch » runSpecification(specification, parameterValues, options) report » deliver(conversation, parameterValues, options)	mobile
Print a report	asynch » run(objectPath, parameterValues, options) asynch » runSpecification(specification, parameterValues, options) report » deliver(conversation, parameterValues, options)	print

When you distribute reports as email attachments, they are not secured by the IBM Cognos security system.

To distribute reports to multiple recipients, create a distribution list. Distribution lists contain a collection of users, groups, roles, contacts, or other distribution lists.

If a recipient is not part of the IBM Cognos security system, you can create a contact for this person. The contacts you create can also be assigned as contacts for reports.

To create distribution lists and contacts, use the [content](#) » [add\(parentPath, objects, options\)](#) method.

You can burst reports so that each recipient receives a subset of the report data. For better performance, we recommend running a burst report as a job through the [jobService](#), instead of running it directly to [batchReportService](#).

You can change the burst options by modifying the report specification. The report specification includes the burst options as attributes of the burst element. For information about modifying report specifications, see [Chapter 23, “Using report specifications,”](#) on page 1457.

You can schedule the distribution of your reports so recipients receive them regularly.

Regardless of how you distribute reports, recipients must have read permissions for the reports and traverse permissions for the folders that contain the reports.

Running Jobs

You can use a job ([bibus » jobDefinition](#)) class to group reports, report views, deployments, content tasks, metrics tasks, or other jobs so that they can share the same [bibus » schedule](#). When a job runs, all the runnable objects specified by individual job steps in the job are executed.

The job service is responsible for handling the execution of jobs. For more information, see [jobService](#).

Jobs contain job steps ([bibus » jobStepDefinition](#) class), which are references to individual objects in the content store, and must be one of the allowed types specified by the [bibus » jobStepDefinition » stepObject](#) property. You can specify whether to run the steps all at once or in sequence by setting the [bibus » jobDefinition » sequencing](#) property to either [parallel](#) or [sequential](#).

When you run the steps in parallel and a step fails, the other steps in the job are unaffected and continue to run. However, the job has a [failed](#) status if one or more job steps fail.

When you run the steps in sequence, you can specify the order of the steps by setting the [bibus » uiClass » displaySequence](#) property. Each step runs only after the preceding step has completed successfully. If a step fails, the remaining steps do not run, and the job has a [failed](#) status. To override the default behavior and have the job steps continue, you can set the value of the [bibus » asynchOptionEnum » continueOnError](#) value to `true`. The job will still have a [failed](#) status.

The objects specified in steps can also have their own schedules. However, the settings contained in these schedules are ignored when running the job.

If a job includes a report that requires the user to provide input, you can specify parameter values for individual job steps by setting the [bibus » jobStepDefinition » parameters](#) property. When you run the job, the parameter values saved in the job step definition are used if none are saved in the report, preventing the need for user interaction.

For more information about setting options and parameter values, and how they are used, see “[Specifying Options and Parameters](#)” on page 66.

To create a job, use the [content » add\(parentPath, objects, options\)](#) method.

Running Agents

You can use agents to execute tasks based on defined conditions. Agents are managed by the [agentService](#) and can run reports, jobs, web service tasks, deployments, stored procedures, and other runnable objects in the content store. For more information, see [agentService](#).

The [bibus » agentDefinition](#) class defines the information for generating the agent's events, as well as the set of tasks that are executed as part of the agent. Tasks can be executed either sequentially or in parallel.

Tasks are defined by [bibus » agentTaskDefinition](#) objects. Each task identifies the type of event that it processes, along with the location of the object that is executed by the task. Options and parameters specified in the task are used by the task object when it executes.

An [bibus » agentDefinitionView](#) refers to an [bibus » agentDefinition](#), allowing that agent definition to be executed with different option and parameter values, or under the control of a different schedule.

Information that must be maintained between invocations of the agent is stored in the content store using an instance of the [bibus » agentState](#) class. Instances of [bibus » agentState](#) are contained by [bibus » transientStateFolder](#) instances. An instance of the [bibus » transientStateFolder](#) class is created automatically by Content Manager.

Instances of [bibus » agentState](#) class contain a [bibus » reportView](#) which contains the report output used to determine whether the conditions for the agent to run have been met. Two [bibus » output](#) instances are compared to construct an instance of [bibus » agentOutputHotList](#). This instance contains the set of events created by the execution of an agent.

For more information about creating and configuring agents, see the *IBM Cognos Event Studio User Guide*.

Filtering Events

It is possible to filter the events passed to an agent task ([bibus » agentTaskDefinition](#)) based on specific conditions. Specifying a data item name in the [bibus » agentTaskDefinition » filterDataItemName](#) property requires the agent to filter the events processed by the agent task. Only events that evaluate to a value of `true` for the named data item are processed by the agent task. The data item must be defined in the agent's condition report (the [bibus » report](#) object contained by the [bibus » agentDefinition](#) object).

If the [bibus » agentTaskDefinition » filterDataItemName](#) property does not contain a data item or contains a data item whose value does not evaluate to `true`, all events are passed to the agent task and no filtering occurs.

Running Content Manager Tasks

The Content Manager service manages information in the content store. For more information, see [contentManagerService](#).

You can run a [bibus » contentTask](#), [bibus » exportDeployment](#), or [bibus » importDeployment](#).

A [bibus » contentTask](#) is a generic object used to perform management functions in the content store. You can specify values for the [bibus » contentTaskOptionEnum](#) enumeration set that allow you to upgrade reports and queries from previous versions of the report specification to the latest IBM Cognos version, or to perform a content store consistency check or repair operation.

An [bibus » exportDeployment](#) or [bibus » importDeployment](#) can be run to distribute content into and out of the content store.

For more information about managing the content store, see [Chapter 5, “Managing content,”](#) on page 55.

For more information about deploying content, see [Chapter 8, “Deploying content,”](#) on page 95.

Running Metrics Tasks

The data integration service is responsible for running metrics-related tasks in the content store. You can run tasks related to the manipulation of metrics data for scorecarding in IBM Cognos Metrics Manager.

The classes [bibus » metricsFileImportTask](#), [bibus » metricsDataSourceETLTask](#), [bibus » metricsExportTask](#), and [bibus » metricsMaintenanceTask](#) are used to define tasks that are executed by [dataIntegrationService](#) service. A metric maintenance task can be run independently or used as a step in a job ([bibus » jobDefinition](#) class) or agent ([bibus » agentDefinition](#) class). The [bibus » dataIntegrationTaskOptionBoolean](#) class is used to specify metrics task options.

Three user capabilities ([canUseMetricStudioEditView](#), [canUseMetricStudio](#), and [canUseMetricsManagerAdministration](#)) control access to Metrics Manager functionality. The roles “Metrics Administrators” on page 1700, “Metrics Authors” on page 1700, and “Metrics Users” on page 1701, assign various user capabilities related to Metrics Manager during installation.

The [bibus » package » userInterfaces](#) property controls which user interfaces are able to use the package.

The [bibus » portalOption](#) class and its derivations are used to specify portal preferences. The [bibus » account » options](#) property stores the portal preferences for a user.

Running Notification Tasks

The delivery service is responsible for managing IBM Cognos notifications. Notifications include updates to RSS channels using either a shortcut or a URL, and the delivery of email messages. For more information, see [deliveryService](#).

The delivery service is primarily used by the agent, report, and batch report services ([agentService](#), [reportService](#), and [batchReportService](#)), but a client can use this service to send a notification to a user.

Understanding the Asynchronous Conversation

When IBM Cognos runs an object (a report, job, deployment, agent, and so on) the service processing the request initiates a conversation with the calling application. In some cases, a single, primary request is made to a service to execute a runnable object, and no conversation takes place as there are no secondary requests that can follow. For long-running, server-based tasks, one or more secondary requests may follow the primary request.

Before using the BI Bus API to execute long-running, server-based tasks, it is beneficial to understand the asynchronous conversation flow of primary and secondary requests described in this section.

Note: You must have a valid session to establish a conversation context. For more information, see [“Authenticating Users”](#) on page 45.

Primary and Secondary Requests

Asynchronous methods in the IBM Cognos include methods that are categorized as either primary requests or secondary requests.

A primary request initiates a conversation between the client and server, establishing a conversation context for subsequent communications. For example, the `asynch » run(objectPath, parameterValues, options)` method returns an `bibus » asynchReply` object. The `bibus » asynchReply » primaryRequest` property contains the conversation context and is required by secondary requests.

The secondary request requires the conversation context of the primary request that initiated the conversation. If you call the `report » deliver(conversation, parameterValues, options)` method for example, you pass the value provided in the `bibus » asynchReply » primaryRequest` property as the `conversation` parameter to the method call to establish the current context.

When calling secondary requests from a Java application, you will need to pass the BI Bus response header into the next request. See [“Managing Service Headers in Java Applications”](#) on page 97 for more information.

Tip: You can determine the list of possible secondary requests after any primary request by checking the `bibus » asynchReply » secondaryRequests` property. For more information, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Using wait()

The BI Bus API provides options to specify the amount of time a primary or secondary request can execute before IBM Cognos responds to the client. When a request cannot be satisfied within the specified time, the server responds with a status of `working`. The default values for wait thresholds for primary and secondary requests are specified using the `bibus » asynchOptionEnum » primaryWaitThreshold` value and the `bibus » asynchOptionEnum » secondaryWaitThreshold` value respectively.

When the client (portal request, client application, or another service) receives a status of `working`, it enters a wait loop to allow the server to complete the request. The client continues to call the `asynch » wait(conversation, parameterValues, options)` method until a status other than `stillWorking` is returned, as shown in the following diagram.

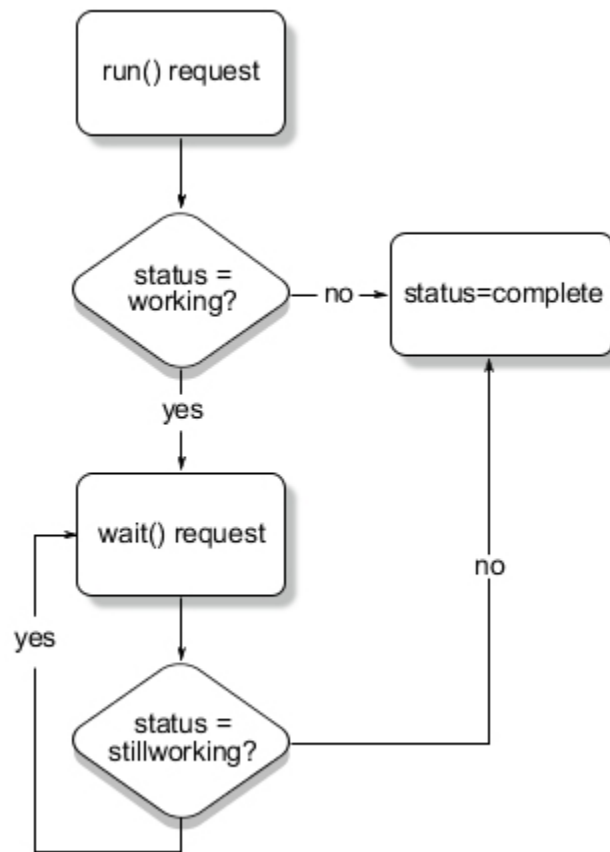


Figure 7. Wait loop for a run() request

The client or application can also choose to abandon the request by calling the `asynch » cancel(conversation)` method (if the status is `working` or `stillWorking`) or the `asynch » release(conversation)` method (if the status is `complete`, to free up resources). If the status is `conversationComplete`, the conversation is already terminated and no secondary requests can be made.

Important: In order to improve resource utilization, you should call the `asynch » release(conversation)` method if you are finished with a conversation context before it has reached a status of `complete`. Doing so can free up system resources earlier by discarding unneeded contexts before they are released automatically.

It is also possible to relinquish control of a conversation by calling the `monitor » background(conversation)` method passing the current conversation context as an input parameter. The `monitorService` takes control of the conversation and continues to call the `asynch » wait(conversation, parameterValues, options)` method until the task is complete. This can be useful if you want to allow your application to engage in other tasks without waiting for the conversation to end.

Optimizing the Asynchronous Conversation

You can bypass the use of the `report » getOutput(conversation, parameterValues, options)` method to retrieve output from the `reportService` service using the `asynch » wait(conversation, parameterValues, options)` method instead.

By setting `bibus » runOptionEnum » returnOutputWhenAvailable` value to `true`, your application does not need to call the `report » getOutput(conversation, parameterValues, options)` method to retrieve report output from the `reportService` service. Instead, you can get available output directly from the `bibus » asynchReply » details` property that is returned in the `wait(conversation, parameterValues, options) » result` return value.

When you use this option, the `reportService` service never returns `responseReady` in the `bibus » asyncDetailReportStatus » status` property.

What's new

New in Version 10.1.0 – “Batch Report Service/Report Service Optimizations” on page 1870

This topic was added.

The Conversation Context

The conversation context contains server state information and communicates this information to the client (for example, a client application). State information is retained in the event of a server failover or for load-balancing purposes.

The three major components of the conversation context are a primary request, a secondary request, and the response from the server that provides information about the status of the request.

Typical Asynchronous Conversation

The following diagram illustrates a typical asynchronous conversation using default settings.

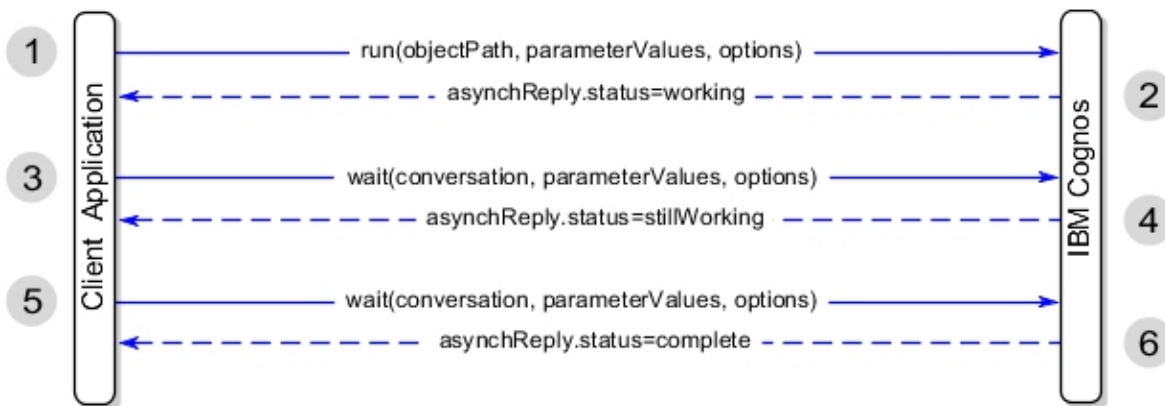


Figure 8. Asynchronous conversation between client application and BI Server

In this scenario, the following steps take place as part of the asynchronous conversation:

1. The conversation is initiated with a `asynch » run(objectPath, parameterValues, options)` request.
2. IBM Cognos could not satisfy the request in the allowed time, so a response with a status of `working` is returned in the `bibus » asyncReply » status` property.

The `bibus » asyncReply » primaryRequest` property is empty since the conversation is not yet complete.

The `bibus » asyncReply » secondaryRequests` property contains instances of the `bibus » asyncSecondaryRequest` class indicating that the next request can be either the `asynch » wait(conversation, parameterValues, options)` method or the `asynch » cancel(conversation)` method.

3. The first call to the `asynch » wait(conversation, parameterValues, options)` method indicates that the client is waiting for it to complete execution of the task initiated in the primary request.

The value provided as the `asynch » wait(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asyncReply » primaryRequest` property to establish the context of the conversation.

4. IBM Cognos could not satisfy the request in the allowed time, so a response with a status of `stillWorking` is returned in `bibus » asyncReply » status` property.

The `bibus » asyncReply » primaryRequest` property is still empty since the conversation is not yet complete.

The `bibus » asynchReply » secondaryRequests` property contains instances of the `bibus » asynchSecondaryRequest` class indicating that the next request can be either the `asynch » wait(conversation, parameterValues, options)` method or the `asynch » cancel(conversation)` method.

- The second call to the `asynch » wait(conversation, parameterValues, options)` method indicates that the client is still waiting for it to complete execution of the task initiated in the primary request.

The value provided as the `asynch » wait(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asynchReply » primaryRequest` property to establish the context of the conversation.

- IBM Cognos has satisfied the request in the allowed time, so a response of `complete` is returned in the `bibus » asynchReply » status` property.

The `bibus » asynchReply » primaryRequest` property is no longer empty and contains enough information to allow another server instance to continue the conversation, if applicable, in the event of load-balancing or server failure.

The `bibus » asynchReply » secondaryRequests` property contains instances of the `bibus » asynchSecondaryRequest` class that indicate that the next allowable request is the `asynch » release(conversation)` method.

Other secondary requests may be available, depending on which service is being called. For more information, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Typical Conversation to Run a Report

The following diagram illustrates a typical asynchronous conversation for running a report, using default settings.

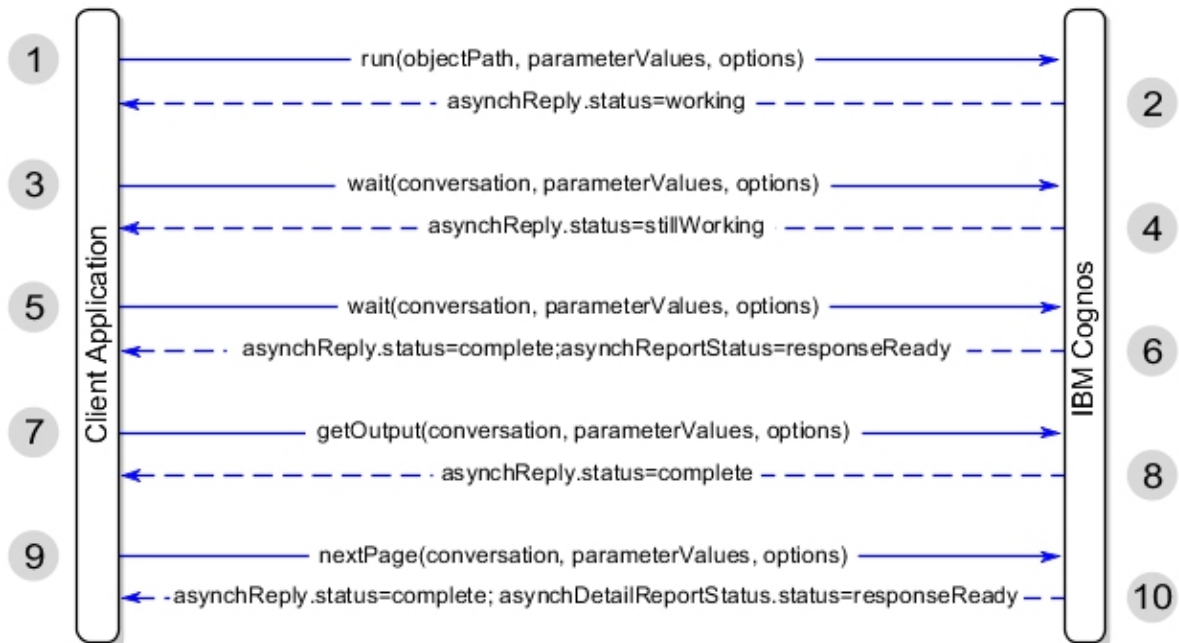


Figure 9. Asynchronous conversation for running a report

In this scenario, the following steps take place as part of the asynchronous conversation:

- The conversation is initiated with a `asynch » run(objectPath, parameterValues, options)` request.
- IBM Cognos could not satisfy the request in the allowed time, so a response with a status of `working` is returned in the `bibus » asynchReply » status` property.

The `bibus » asynchReply » primaryRequest` property is empty since the conversation is not yet complete.

The `bibus » asynchReply » secondaryRequests` property contains instances of the `bibus » asynchSecondaryRequest` class indicating that the next request can be either the `asynch » wait(conversation, parameterValues, options)` method or the `asynch » cancel(conversation)` method.

3. The first call to the `asynch » wait(conversation, parameterValues, options)` method indicates that the client is waiting for it to complete execution of the task initiated in the primary request.

The value provided as the `asynch » wait(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asynchReply » primaryRequest` property to establish the context of the conversation.

4. IBM Cognos could not satisfy the request in the allowed time, so a response with a status of `stillWorking` is returned in `bibus » asynchReply » status` property.

The `bibus » asynchReply » primaryRequest` property is still empty since the conversation is not yet complete.

The `bibus » asynchReply » secondaryRequests` property contains instances of the `bibus » asynchSecondaryRequest` class indicating that the next request can be either the `asynch » wait(conversation, parameterValues, options)` method or the `asynch » cancel(conversation)` method.

5. The second call to the `asynch » wait(conversation, parameterValues, options)` method indicates that the client is still waiting for it to complete execution of the task initiated in the primary request.

The value provided as the `asynch » wait(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asynchReply » primaryRequest` property to establish the context of the conversation.

6. IBM Cognos has satisfied the request in the allowed time, so a response of `complete` is returned in the `bibus » asynchReply » status` property.

The `bibus » asynchReply » primaryRequest` property is no longer empty and contains enough information to allow another server instance to continue the conversation, if applicable, in the event of load-balancing or server failure.

The value of the `bibus » asynchDetailReportStatus » status` property is `responseReady`, indicating that report output is ready and can be retrieved by calling the `report » getOutput(conversation, parameterValues, options)` method.

The `bibus » asynchReply » secondaryRequests` property contains instances of the `bibus » asynchSecondaryRequest` class that indicate that the next request can be one of the following methods:

- `paging » nextPage(conversation, parameterValues, options)`
- `paging » lastPage(conversation, parameterValues, options)`
- `report » deliver(conversation, parameterValues, options)`
- `report » render(conversation, parameterValues, options)`
- `asynch » release(conversation)`

Other secondary requests may be available, depending on which service is being called. For more information, see Chapter 17, “Secondary requests,” on page 1417.

7. The call to the `report » getOutput(conversation, parameterValues, options)` method retrieves the prepared output from the service (in this example, the report service).

The value provided as the `report » getOutput(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asynchReply » primaryRequest` property to establish the context of the conversation.

8. The requested action was successful so a response of `complete` is returned in the `bibus » asynchReply » status` property.

The `bibus » asynchReply » secondaryRequests` property contain instances of the `bibus » asynchSecondaryRequest` class that indicates that one of the next allowable requests is the `paging » nextPage(conversation, parameterValues, options)` method.

9. The call to the `paging » nextPage(conversation, parameterValues, options)` method moves to the next page of the report output.

The value provided as the `paging » nextPage(conversation, parameterValues, options) » conversation` parameter is obtained from the `bibus » asynchReply » primaryRequest` property to establish the context of the conversation.

10. The requested action was successful so a response of `complete` is returned in the `bibus » asynchReply » status` property.

Processing Constraints

Requests can originate from the web portal, as a result of an object's schedule firing, or through the IBM Cognos Software Development Kit. Regardless of the source, the `dispatcher` service monitors all secondary processes, ensuring that follow-up actions go to the correct instance of the appropriate service. Because the service records both synchronous and asynchronous conversations in its request history, you can track the amount of time spent in any stage of the process.

We recommend that you specify a socket timeout setting that is greater than the value for either the primary or secondary request wait threshold (`primaryWaitThreshold` and `secondaryWaitThreshold`). Otherwise, the dispatcher may cancel the request before the response is received.

Chapter 7. Dispatchers and services

You can use the BI Bus API to manage, monitor, and tune the IBM Cognos dispatchers and services. The objects you will work with are stored in the Configuration area of Content Manager.

To access dispatcher properties and change settings, you must have the `canUseAdministrationPortal` capability and execute permission for the `canUseServerAdministrationTool` secured feature.

For information about setting access permissions, see [Chapter 4, “Managing security,” on page 45](#).

To administer dispatchers and services, you can

- Read the availability of services and studios ([“Read the Availability of Services and Studios” on page 85](#))
- Read the status of dispatchers and services ([“Read the Status of Dispatchers and Services” on page 86](#))
- Start and stop dispatchers and services ([“Stop and Start Dispatchers and Services” on page 86](#))
- Remove dispatchers from the environment ([“Remove Dispatchers from the Environment” on page 87](#))
- Test dispatchers ([“Test Dispatchers” on page 87](#))
- Activate a Content Manager service ([“Activate a Content Manager Service” on page 87](#))
- Tune server performance ([“Tune Server Performance” on page 89](#))
- Set the logging level ([“Set the Logging Level” on page 88](#))
- Group dispatchers ([“Group Dispatchers in Configuration Folders” on page 90](#))
- Define server groups ([“Define Server Groups” on page 90](#))
- Configure advanced routing ([“Configure Advanced Routing” on page 90](#))
- Use advanced routing ([“Use Advanced Routing” on page 92](#))
- Hot swap a PowerCube ([“Hot Swapping PowerCubes” on page 93](#))

Standby Dispatchers

You can use IBM Cognos Configuration to configure standby dispatchers for failover recovery.

Before Using the IBM Cognos Software Development Kit for Server Administration

To perform the tasks described here, you must enter the search path for the dispatcher or service you want to monitor or configure. For information about search paths, see [Chapter 32, “Search path syntax,” on page 1565](#).

Read the Availability of Services and Studios

You can determine which IBM Cognos components are installed with a particular dispatcher using instances of the `bibus » installedComponent` class.

To view the availability of a service or studio, do one of the following:

- To view the availability of a specific service or studio, use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the `bibus » installedComponent » componentID` property, identifying the full path to the `bibus » installedComponent` object of the `bibus » dispatcher` specified in the `searchPath` parameter, including the name of the component in question.

For example, `/configuration/dispatcher[@name="<dispatcherName>"]/installedComponent[@name="<componentName>"]`.

- To view a full list of components installed on a dispatcher, use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the `bibus » installedComponent » componentID` property, identifying a search path that selects all instances of `bibus » installedComponent` contained by the desired `bibus » dispatcher` in the `searchPath` parameter.

For example, `/configuration/dispatcher[@name="<dispatcherName>"]/installedComponent`].

All installed and available services and components are returned.

Read the Status of Dispatchers and Services

To evaluate how IBM Cognos is performing, you can use the BI Bus API to retrieve information about the dispatchers and services configured for your environment. This information includes, among other details, the status of a service, the volume of requests, and the time required to handle the requests. Note that only the properties specified in the query are returned, along with the search path to the object of the query.

If a dispatcher has an unknown status, it may be too busy to show its information or it may not be responding. You can test the dispatcher (“Test Dispatchers” on page 87) to see whether it responds. If dispatchers and services are not performing as expected, you can tune their performance by changing their configuration settings (“Tune Server Performance” on page 89).

To read the status information, use the `content » query(searchPath, properties, sortBy, options)` method to locate the dispatcher or service you want to investigate, and retrieve the `bibus » dispatcher » runningState` property. The same method can be used to query the `runningState` property of each of the services contained by a dispatcher. For more information about available services, see “List of Services” on page 130.

You can also use IBM Cognos Administration to obtain metrics about the status of your dispatchers and services.

To monitor system metrics externally to IBM Cognos Administration, you can use Java Management Extensions (JMX), a technology that supplies tools to manage and monitor applications and service-oriented networks. The information from Content Manager obtained this way is current, as opposed to the information obtained using a BI Bus API request that is not updated on a regular basis by default.

Stop and Start Dispatchers and Services

You can use the BI Bus API to stop and start dispatchers and services. When you stop a dispatcher, all its services are stopped, except for the system service. The system service cannot be stopped. You must stop a dispatcher before you can delete it from Content Manager (“Remove Dispatchers from the Environment” on page 87).

Note: If you stop the active `contentManagerService`, or a dispatcher that contains the active `contentManagerService`, all users are locked out of IBM Cognos except for Server Administrators. Users must log on again when the service is restarted. This functionality can be useful when you perform emergency maintenance.

You can also use IBM Cognos Administration to obtain to stop and start dispatchers and services.

Stopping or starting the IBM Cognos service in IBM Cognos Configuration also stops or starts the dispatcher and its services.

You must have `execute` permissions on the `canUseServerAdministrationTool` secured feature to stop and start dispatchers and services.

Steps

1. Use the `dispatcher » stopService(servicePath, immediately)` method and the `dispatcher » startService(servicePath)` method to stop and start dispatchers or services.

To stop or start a particular service, use the `service` parameter of the method to name the service.

For a list of service names, see [“List of Services”](#) on page 130.

2. For the `reportService` or `batchReportService` service, set the `immediately` parameter of the `stopService` method to `true` to stop the service without completing any jobs that are running or queued.

Remove Dispatchers from the Environment

You can use the BI Bus API to remove a dispatcher if you no longer need it in the IBM Cognos environment.

You must first stop the IBM Cognos service using IBM Cognos Configuration. This will stop the dispatcher as well. If you delete a dispatcher without stopping the IBM Cognos service first, the dispatcher will automatically be reinstated in 30 seconds.

You can also use IBM Cognos Administration to remove a dispatcher. However, IBM Cognos Administration user interface refers to this process as unregistering a dispatcher.

Steps

1. Stop the IBM Cognos service using IBM Cognos Configuration.

This also stops the dispatcher.

2. Use the `content » delete(objects, options)` method to remove the dispatcher from Content Manager.

Test Dispatchers

To evaluate how IBM Cognos is performing, you can test the dispatchers configured for your environment. During the testing, you verify whether a dispatcher is responding, and view its uptime, which is the time in seconds during which a dispatcher works without failure.

For additional information, you can view the status of dispatchers and services ([“Read the Status of Dispatchers and Services”](#) on page 86) and review log messages ([“Set the Logging Level”](#) on page 88).

You can also use IBM Cognos Administration to obtain metrics about the status of your dispatchers and services.

If dispatchers and services are not performing as expected, you can tune their performance by changing their configuration settings ([“Tune Server Performance”](#) on page 89).

Step

- To test dispatchers, use the `dispatcher » ping(dispatcherPath)` method.

The `dispatcher » ping(dispatcherPath)` method returns the `upTime` and `version` information for the named dispatcher.

Activate a Content Manager Service

You can use the BI Bus API to manually activate a `contentManagerService` service instance that is in standby mode. When you activate a service, the currently active service switches to standby mode.

Only one Content Manager service can be active at any time.

You must have `execute` permissions to the `canUseServerAdministrationTool` secured feature to activate a Content Manager service.

To activate a Content Manager service, use the `content » activate(searchPath)` method.

Set the Logging Level

You can use the BI Bus API to configure the level of detail to log. By setting the logging level, you specify what detail of events and messages a service records in the log file or in the log database.

Possible logging levels are: `minimal`, `basic`, `request`, `trace`, or `full`. The default level is `minimal`. For more information, see the [bibus » auditLevelEnum](#) enumeration set.

The logging settings are inherited from parent objects, from the `configuration` class to the `configurationFolder`, `dispatcher`, and `service` classes. Child objects take their settings from the parent class unless they are otherwise specified. Therefore, you can set logging levels at the `configuration` object level for all dispatchers and services in the IBM Cognos environment, at the `bibus » configurationFolder` object level for all dispatchers and services in a configuration folder grouping, and at the `dispatcher` and `service` levels for an individual dispatcher and its services. You can set the logging level for an individual service using the logging property of the specific service class. For example, to set logging for the report service, use the `bibus » reportService » rsAuditLevel` property.

For a list of service names, see [Chapter 12, “Services,”](#) on page 129.

Logging Properties

To configure logging for services, use one or more of the following properties of the `configuration`, `configurationFolder`, `dispatcher`, or `service` class.

Each property relates to the logging level for a specific service. Note that audit settings for the dispatcher service do not propagate to its child services; for the purpose of logging, the dispatcher is treated as a distinct service.

- [asAuditLevel](#) property
- [brsAuditLevel](#) property
- [brsAuditNativeQuery](#) property
- [cmsAuditLevel](#) property
- [disAuditLevel](#) property
- [dispatcherAuditLevel](#) property
- [dsAuditLevel](#) property
- [emsAuditLevel](#) property
- [idsAuditLevel](#) property
- [issAuditLevel](#) property
- [iusAuditLevel](#) property
- [jsAuditLevel](#) property
- [mdsAuditLevel](#) property
- [mmsAuditLevel](#) property
- [msAuditLevel](#) property
- [pacsAuditLevel](#) property
- [prsAuditLevel](#) property
- [psAuditLevel](#) property
- [ptsAuditLevel](#) property
- [rsAuditLevel](#) property
- [rsAuditNativeQuery](#) property
- [ssAuditLevel](#) property

Steps

1. Use the [content » query\(searchPath, properties, sortBy, options\)](#) method to retrieve the logging properties of the [bibus » configuration](#) class, the [bibus » configurationFolder](#) class, the [bibus » dispatcher](#) class, or a named service class.
2. Modify the properties that you want to change.
3. Use the [content » update\(objects, options\)](#) method to save the new logging properties in the content store.

Tune Server Performance

By tuning the configuration settings of dispatchers and services, you can optimize the speed and efficiency of IBM Cognos.

You can also group dispatchers in configuration folders ([“Group Dispatchers in Configuration Folders” on page 90](#)) to set the configuration settings for more than one dispatcher at time.

Aside from tuning dispatchers, you may want to add dispatchers to your installation to meet the demands of users. Also, you may want to distribute your installation or upgrade the computer on which IBM Cognos is installed.

The level of logging ([“Set the Logging Level” on page 88](#)) can affect performance. When IBM Cognos logs more detail, more resources are allocated to logging and fewer resources are then available to run reports.

Models

Before you change any configuration settings, ensure that your models are optimized for reporting. For more information, see the *Framework Manager User Guide*.

Operating Systems

How IBM Cognos performs is closely related to the performance of the operating system of the computer where IBM Cognos is installed. Therefore, it is important to ensure that your operating system is tuned properly.

Tuning Properties

To tune server performance, use one or more of the following properties of the [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), or service class. Items that are listed as properties of a service, can also be set at the three higher levels, unless otherwise noted.

- [advancedSettings](#) property

For information about advanced settings, see *Advanced settings configuration*.

- [brsAffineConnections](#) property
- [brsExecutionTimeLimit](#) property
- [brsMaximumProcesses](#) property
- [brsNonAffineConnections](#) property
- [capacity](#) property
- [loadBalancingMode](#) property
- [bibus » configuration » routingTable](#) property

This property is only contained in the top-level configuration object.

- [rsAffineConnections](#) property
- [rsExecutionTimeLimit](#) property
- [rsMaximumProcesses](#) property

- [rsNonAffineConnections](#) property
- [rsQueueLimit](#) property
- [serverGroup](#) property

Before you change any settings, ensure that you test dispatchers and services (“Test Dispatchers” on page 87), review the pertinent log messages (“Set the Logging Level” on page 88), and understand your performance requirements.

Steps

1. Use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the configuration properties of the `bibus » configuration`, `bibus » configurationFolder`, `bibus » dispatcher`, or service object.
2. Modify the properties that you want to change.
3. Use the `content » update(objects, options)` method to save the new configuration settings in the content store.

Group Dispatchers in Configuration Folders

You can use the BI Bus API to group dispatchers so that you can apply the same configuration settings once to all the dispatchers and services in the folder.

When you move a dispatcher to a configuration folder, it automatically inherits the configuration settings (“Tune Server Performance” on page 89) of the folder. However, if you previously changed the default values of a dispatcher or service, the changed values are kept.

You can create a new `bibus » configurationFolder` object at the root of the `bibus » configuration` object, or in an existing `bibus » configurationFolder` object.

Steps

1. Create a new object for the new `bibus » configurationFolder`.
2. Use the `content » add(parentPath, objects, options)` method to save the new `bibus » configurationFolder` in the content store.
3. Use the `content » move(objects, targetPath, options)` method to move the dispatchers to the newly created `bibus » configurationFolder`.

Define Server Groups

To use advanced routing (“Configure Advanced Routing” on page 90), you must set server group names for the dispatchers or configuration folders to which you want reports to be routed.

Properties are inherited from parent objects. Therefore, if you define a `serverGroup` property at the `bibus » configurationFolder` level, child `configurationFolder` and `dispatcher` objects inherit the setting, unless otherwise set at the lower level.

Steps

1. Set the `serverGroup` property of either the `bibus » dispatcher` or `bibus » configurationFolder` object as desired.
2. Use the `content » update(objects, options)` method to save the new property in the content store.

Configure Advanced Routing

You can enable the use of advanced routing to control how tasks are distributed across servers. Advanced routing is recommended for environments with advanced configurations, such as distributed installations or segregated user communities.

To configure advanced routing, text strings called hints are defined in the `bibus » dashboard » routingHints` property. If you have servers that are optimized for use by different user groups, you can add hints to the group objects to enable routing to specific servers. For example, give the Sales group object a hint of `GOsales`, and the Finance user group a hint value of `Finance`.

The `bibus » routingTableEntry` object contains rules that can be matched with the `bibus » routingHintObject`. The `bibus » routingTableEntry` object contains the `rule` property, which is made up of the `hints` property of the `bibus » routingRuleObject` object. Using the example given above, a `bibus » routingTableEntry` for the Sales group lists `SalesServers` as the server group to handle a request from that group, and the `FinanceServers` as the server group to handle requests from the Finance group.

It is possible to assign multiple hints to one object. Hint values identify a set of objects: all objects of the same class with the same hint value. For example, a package could have hints `GOsales` and `PublicPackage`. The package is thus a member of two sets: packages with hint equal to `GOsales`, and packages with hint equal to `PublicPackage`. If there are several packages with hint `PublicPackage`, one routing rule could be defined to do specific routing for `PublicPackage` package requests for the Sales group, another routing rule could route requests for the Finance group, and a third more general rule could be defined to route all other requests for packages with hint `publicPackage`.

You must have server groups defined to use advanced routing. For information, see “[Define Server Groups](#)” on page 90.

Steps

1. Set the `routingHints` property of one or more of the `bibus » package` class, `bibus » account` class, `bibus » group` class, or `bibus » role` class, depending on how you want to configure advanced routing. These hints must also be listed in the `rule` property to be matched with a `serverGroupName` property.
2. Set the values for the `bibus » routingHintObject » objectClass` property and the `bibus » routingHintObject » value` property to set the rules for matching with server groups. For example, to match packages with hints of "GoSales" the `objectClass` is `package`, and the value is "GoSales."
3. Set the `serverGroupName` property to go with the `objectClass` property identified in the rule property. For example, route packages with hints of "GoSales" to the "SalesServers" server group.
4. Use the `content » update(objects, options)` method to save the new properties in the content store.

Configure Advanced Settings

You must have the following access permissions to the affected configuration and service objects to change advanced settings:

- read and write permissions to the object that you want to update
- traverse permission for the parent of the object that you want to update

For information about access permissions, see [Chapter 4, “Managing security,”](#) on page 45.

Though the property exists, there are no advanced settings available for the `bibus » dataIntegrationService`, the `bibus » logService`, the `bibus » metadataService` class, or the `bibus » systemService`. The `advancedSettings` properties of the `bibus » configurationFolder` and `bibus » dispatcher` object, while present, should not be set.

You specify global `advancedSettings` by using the `advancedSettings` of the `bibus » configuration` object. Values specified in this object are acquired by all contained objects unless the property of a contained object is set to override the global settings. You may need to override advanced settings to provide customized values for specific service instances, but this can unnecessarily increase the overhead of administration.

You can change these advanced settings at run time without restarting the service. However, changing these settings using the BI Bus API requires XML programming skills.

The following example shows the XML format for setting the `advancedSettings` property of a `bibus » contentManagerService` object.

```
<settings>
  <setting name="CM.SecurityQueryRequiresRead">false</setting>
</settings>
```

See also:

- [Chapter 33, “Advanced settings configuration,” on page 1577](#)
- [“Recommendation - Managing Long-Running or Resource-Intensive Tasks” on page 119](#)

Steps

1. Use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the existing `advancedSettings` property of the object you want to update, such as the root `bibus » configuration` object or a specific service.
2. Modify the values that you want to change or add by setting the `advancedSettings` property.
To find the setting names and values that you need to modify, see *Advanced settings configuration*.
3. Use the `content » update(objects, options)` method to save the changes in the content store.

Use Advanced Routing

You can use advanced routing to control how tasks are distributed across servers. Before you use advanced routing, ensure that you have performed the tasks listed in [“Configure Advanced Routing” on page 90](#).

After you define the routing hints and the routing table rules, you can send requests to make use of the advanced routing. When you send the request, you must add the `routingServerGroup` element to the `bibus » biBusHeader` object as a direct child of `bibus » biBusHeader`, as shown in the following example. In this example, a request is forwarded to the GOSales server group:

```
<biBusHeader xsi:type="tns:biBusHeader"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <routing>
    <routingServerGroup>GOSales</routingServerGroup>
  </routing>
</biBusHeader>
```

To use advanced routing for objects that exist in the content store, see [“Steps to Use Advanced Routing for Content Store Objects” on page 92](#).

To use advanced routing for inline specifications (objects that do not exist in the content store), see [“Steps to Use Advanced Routing for Inline Specifications” on page 93](#).

Steps to Use Advanced Routing for Content Store Objects

1. Use the `content » query(searchPath, properties, sortBy, options)` method to retrieve the `routingServerGroup` property of the object you want to run.
This calls the `content » determineRouting(objectPaths)` method, which returns the appropriate server group to run the object, based on the hints provided.
2. When you send the run request for the object, set the `routingServerGroup` element in the `bibus » biBusHeader` object to the value returned from Step 1.
For information, see the XML example in this topic introduction.

Steps to Use Advanced Routing for Inline Specifications

1. Use the `content » determineRouting(objectPaths)` method, passing the `bibus » searchPathSingleObject` to the objects that may contain hints for the `objectPaths` parameter.

For example, pass the search path to the package object related to a report you are running.

Note: It is not necessary to pass `bibus » account`, `bibus » group`, or `bibus » role` objects as these are inferred from the Passport of the user that is logged on. If the object you are running does not contain hints, for example a `bibus » contentTask`, pass an empty array for the `objectPaths` parameter, and the `content » determineRouting(objectPaths)` method will use information from the Passport of the logged on user to find hints on the `bibus » account`, `bibus » group`, or `bibus » role` objects to which that user belongs.

2. Set the `routingServerGroup` element in the `bibus » biBusHeader` object to the result of the `content » determineRouting(objectPaths)` method.

If a nil value is returned, regular dispatcher routing is used.

Hot Swapping PowerCubes

You can transition users to a new version of a PowerCube without requiring the complete shutdown of the application.

If the PowerCube is available on a subset of the servers used by the application, it is necessary to change the properties of those dispatchers only (the `bibus » dispatcher » brsDataSourceChange` property and the `bibus » dispatcher » rsDataSourceChange` property). This advanced procedure allows data source connections in use by other servers to be maintained.

If the affected dispatchers are all located in a folder, it is possible to change the properties `brsDataSourceChange` and `rsDataSourceChange`, and allow the contained dispatchers to acquire the value from the folder.

Steps

1. Construct the new version of the PowerCube.
2. Move the PowerCube to the new server, assigning the cube a new name.
For example, append the current date to the file name.
3. Update the `bibus » dataSourceConnection » connectionString` property so that it references the new version of the PowerCube.
4. Set the `bibus » configuration » brsDataSourceChange` property and the `bibus » configuration » rsDataSourceChange` property to the current time.

When the dispatcher detects that the configuration has changed, the transition to the new PowerCube version begins. The dispatcher relays the new configuration to all affected services (the batch report service and the report service). The services disconnect from all data sources. Subsequent requests cause new connections to be opened using the new connection strings.

5. Enable acquisition to allow contained objects to use the new value by setting the `bibus » schemaInfo » applyUpdateToDescendants` property to `true`.

Chapter 8. Deploying content

Deployment involves moving applications from one installation to another. In IBM Cognos, you deploy packages, top-level folders, or the entire content store database from a source environment to a target environment. In a typical scenario, you deploy folders, reports, and report views from your development environment to a test environment and then to a production environment. You can also deploy data between different operating systems.

You can use the BI Bus API to deploy package contents, as well as Cognos namespaces, data sources, distribution lists, and contacts. You can automate the following tasks:

- creating a deployment specification
- submitting deployment export and import requests
- accessing deployment history

You can also schedule content manager deployments. You can schedule the deployments independently, or in a job or agent. For detailed information about deployment such as planning, options, and conflict resolution, see the *Administration and Security Guide*.

Creating a Deployment Specification

To create a new export deployment specification, add an `bibus » exportDeployment` object to an `bibus » adminFolder` object in your source content store.

To create a new import deployment specification, add an `bibus » importDeployment` object to an `bibus » adminFolder` object in your target content store.

Add, retrieve, and manipulate these objects with the content manager methods `content » add(parentPath, objects, options)`, `content » query(searchPath, properties, sortBy, options)`, `content » update(objects, options)`, and `content » delete(objects, options)`.

Exporting Data From the Content Store

You export data from the content store to a deployment archive. The location of the deployment archive is an IBM Cognos startup parameter.

To export data, use the `asynch » run(objectPath, parameterValues, options)` method, specifying the search path for the `bibus » exportDeployment` object in the `objectPath` parameter. Both `import` and `export` options must be specified for the `bibus » exportDeployment` object.

Moving the Deployment Archive

If the source and target environments use the same content store, you can import your data without physically moving the deployment archive. If they use different databases, you must move the deployment archive to the target environment.

We recommend that you copy the deployment archive to a secure location, because the deployment archive may not be encrypted.

Importing Data Into the Content Store

You import data from a deployment archive into the target environment. The location of the deployment archive is an IBM Cognos startup parameter.

To import data, use the `asynch » run(objectPath, parameterValues, options)` method, specifying the search path for the `bibus » importDeployment` object in the `objectPath` parameter. The `import` option must be specified for the `bibus » importDeployment` object.

Accessing Deployment History

The deployment archive is created when you export data from the source environment. You move it to the target environment and then import it into the target. This archive contains all the data that is deployed.

Associated with each archive is the set of deployment options that were applied during the export and import sequence. You can retrieve these options from the content store for reference purposes.

Listing the Deployment Archives

You can retrieve the history for a specific deployment from the deployment archives referenced in the content store. This is useful when you want to verify that your deployment ran as expected. The default location for the deployment archives is *installation_location/deployment*. You can change this location by using IBM Cognos Configuration.

To list the deployment archives, use the `deployment » listArchives()` method.

Retrieving a Set of Deployment Options

You can use the BI Bus API to retrieve the deployment options used in a previously executed deployment operation.

To view the deployment options for an archive, use the `deployment » getDeploymentOptions(archive, options)` method.

Chapter 9. Coding practices and troubleshooting

This section provides the following:

- coding guidelines to follow when working with the IBM Cognos Software Development Kit and its related samples ([“Coding Practices” on page 97](#))
- information about problems you may encounter when working with the IBM Cognos Software Development Kit and potential solutions ([“Troubleshooting” on page 110](#))
- recommended practices to follow when developing your applications ([“Application Development Guidelines” on page 118](#))

For troubleshooting information that is not specific to the IBM Cognos Software Development Kit, see the *IBM Cognos Troubleshooting Guide*.

Coding Practices

This section includes information about coding practices for using the IBM Cognos Software Development Kit with various programming languages. Depending on your programming language and your goals, you may need to:

- manage service headers ([“Managing Service Headers in Java Applications” on page 97](#))
- handle exceptions generated by IBM Cognos ([“Handling Exceptions” on page 99](#))
- use language-specific naming conventions ([“Language-specific Naming Conventions” on page 108](#))
- follow certain guidelines for using URLs ([“Recommendations for URLs” on page 108](#))
- specify values of type `xsdDate` and `xsdDateTime` ([“Specifying Parameter Values of Type `xsdDateTime` or `xsdDate`” on page 109](#))
- replace illegal characters in MIME headers ([“Replacing Illegal Characters in MIME Headers” on page 110](#))
- interpret messages that use the `Response.Write` format ([“Interpreting Messages that Use `Response.Write` Format” on page 110](#))
- localize API elements for non-English locales ([“Localizing API Elements for Non-English Locales” on page 110](#))
- test connections to data sources ([“Test Connections to Data Sources” on page 110](#))

Managing Service Headers in Java Applications

When you call a method on a service, the response header for the service may include important information that is required for subsequent requests. Some method calls require information from a previous call to the service, such as tracking information that indicates which server handled the previous request. For example, a `wait()` request must be routed to the same server that handled the primary request.

When writing a Java application, when a method requires the header to include information from the return header of a previous method call, you must include code to copy the response header into the `bibus » biBusHeader` object. Before each method call on a service, you should update the header based on the response header value from the previous call. After retrieving the response header, clear the headers before setting the `bibus » biBusHeader` to the value of the response header.

Note: The .NET toolkit manages the service headers automatically for C# applications.

Updating the BI Bus header (Java)

The following code snippet demonstrates how to manage the headers for a service in a Java application, including how to retrieve a `bibus » biBusHeader` object from a `SOAPHeaderElement`

object. To see this code in context, view the Java samples Common/CRNConnect.java and Common/BIBusHeaderHelper.java:

installation_location/sdk/java/Common/CRNConnect.java

installation_location/sdk/java/Common/BIBusHeaderHelper.java

```
public ReportService_PortType getReportService
    (boolean isNewConversation, String RSGroup)
{
    BiBusHeader bibus = null;
    bibus =
        BIBusHeaderHelper.getHeaderObject(((Stub)repService).getResponseHeader(
            "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader"),
            isNewConversation, RSGroup);

    if (bibus == null)
    {
        BiBusHeader CMBibus = null;
        CMBibus =
            BIBusHeaderHelper.getHeaderObject(((Stub)cmService).getResponseHeader(
                "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader"),
                true, RSGroup);
        ((Stub)repService).setHeader(
            "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader",
            CMBibus);
    }
    else
    {
        ((Stub)repService).clearHeaders();
        ((Stub)repService).setHeader(
            "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader",
            bibus);
    }
    return repService;
}

// handle service requests that do not specify new conversation for
// backwards
// compatibility
public SystemService_PortType getSystemService()
{
    return getSystemService(false, "");
}

//Use this method when copying headers, such as for requests to services
public static BiBusHeader getHeaderObject(
    SOAPHeaderElement SourceHeader, boolean isNewConversation, String
RSGroup)
{
    if (SourceHeader == null)
        return null;

    BiBusHeader bibus = null;
    try {
        bibus = (BiBusHeader)SourceHeader.getValueAsType(BUS_QNAME);
        // Note BUS_QNAME expands to:
        // new QName("http://developer.cognos.com/schemas/bibus/3/",
"biBusHeader")

        //If the header will be used for a new conversation, clear
//tracking information, and set routing if supplied (clear if not)
        if (isNewConversation){

            bibus.setTracking(null);

            //If a Routing Server Group is specified, direct requests to
it
            if (RSGroup.length(>0) {
                RoutingInfo routing = new RoutingInfo(RSGroup);
                bibus.setRouting(routing);
            }
        }
    }
}
```

```

        }
        else {
            bibus.setRouting(null);
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
return bibus;
}

```

Handling Exceptions

This section provides information about handling IBM Cognos exceptions using various programming languages.

Java Exceptions

IBM Cognos errors are reported to Java applications as `org.apache.axis.AxisFault` exceptions. The following methods of the `AxisFault` exception contain IBM Cognos error information:

- `getMessage()` - returns the exception message as a string
- `getDetails()` - returns an array of XML `org.w3c.dom.Node` objects containing the exception details; this array corresponds to the [bibus » faultDetail](#) class.

Use the `javax.xml.transform.XPathAPI.selectSingleNode()` method to find the IBM Cognos error code and severity. Use the `javax.xml.transform.XPathAPI.selectNodeList()` method to find the IBM Cognos error details.

If the basic `AxisFault` exception indicates that extended fault information is available, you must process the [bibus » CAM » exception](#) property element found in the [bibus » biBusHeader](#) class. For more information, see the [bibus » CAM » exception](#) property.

Catching AxisFault (Java)

The following code shows you how to use a `try` block to catch the `org.apache.axis.AxisFault` exception:

```

try
{
    ... Code using one or more Cognos methods.
    ...
}
catch( org.apache.axis.AxisFault cognos_exception )
{
    ... Exception-handling code.
    ...
}

```

Handling an IBM Cognos Exception (Java)

The following code shows you how to extract the important exception details. To see this code in context, view the Java sample `runreport\CognosBIException.java`:

```

import javax.xml.xpath.*;
import org.apache.axis.AxisFault;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

```

```

public class CognosBIException
{
    private AxisFault _exception = null;

    /**
     * Create a CognosBIException object.
     * @param ex An AxisFault thrown by an IBM Cognos method call.
     */
    public CognosBIException(AxisFault ex)
    {
        _exception = ex;
    }

    /**
     * Return the exception message.
     * @return The exception's message string.
     */
    public String getMessage()
    {
        return _exception.getMessage();
    }

    /**
     * Return the exception severity.
     * @return The exception severity string.
     */
    public String getSeverity()
    {
        try
        {
            Node n =
                getSingleNode(
bibus/3/" and " (/*[namespace-uri()=\"http://developer.cognos.com/schemas/
                    local-name()=\"severity\"])[1]");

            return new String(n.getFirstChild().getNodeValue());
        }
        catch (Exception ex)
        {
            return new String("");
        }
    }

    /**
     * Return the exception errorCode.
     * @return The exception errorCode string.
     */
    public String getErrorCode()
    {
        try
        {
            Node n =
                getSingleNode(
bibus/3/" and " (/*[namespace-uri()=\"http://developer.cognos.com/schemas/
                    local-name()=\"errorCode\"])[1]");

            return new String(n.getFirstChild().getNodeValue());
        }
        catch (Exception ex)
        {
            return new String("");
        }
    }
}

```

```

    * Return the exception's messageStrings.
    *
    * @return The exception messageString array of strings.
    */
public String[] getDetails()
{
    try
    {
        NodeList nodes =
            getNodeList(
                "//*[namespace-uri()=\\"http://developer.cognos.com/schemas/
bibus/3/\\" and
                local-name()=\\"messageString\\"];

        String retval[] = new String[nodes.getLength()];
        for (int idx = 0; idx < nodes.getLength(); idx++)
        {
            retval[idx] =
                new String(nodes.item(idx).getFirstChild().getNodeValue());
        }

        return retval;
    }
    catch (Exception ex)
    {
        return new String[] { "" };
    }
}

/**
 * Convert this CognosBIException into a string for printing.
 *
 * @return A string representation of the CognosBIException.
 */
public String toString()
{
    StringBuffer str = new StringBuffer();

    str.append("Message:      ").append(getMessage()).append("\n");
    str.append("Severity:      ").append(getSeverity()).append("\n");
    str.append("ErrorCode:   ").append(getErrorCode()).append("\n");
    str.append("Details:\n");
    String details[] = getDetails();
    for (int i = 0; i < details.length; i++)
    {
        str.append("\t").append(details[i]).append("\n");
    }

    return str.toString();
}

/**
 * Return a Node from the exception based on supplied search path
 * intended to return a single Node.
 *
 * @param searchString An XPath expression
 *
 * @return Node n.
 */
public Node getSingleNode(String searchString) throws
    XPathFactoryConfigurationException, XPathExpressionException
{
    XPath xpath = XPathFactory.newInstance().newXPath();

    XPathExpression xpathExpr = xpath.compile(searchString);

    Node n = (Node)xpathExpr.evaluate(_exception.getFaultDetails()
[0].getParentNode(),
        XPathConstants.NODE);

    return n;
}

```

```

    }

    /**
     * Return a NodeList from the exception based on supplied search
     path
           intended to return one or more Nodes.
     *
     * @param searchString An XPath expression
     *
     * @return NodeList nl.
     */
    public NodeList getNodeList(String searchString) throws
        XPathFactoryConfigurationException, XPathExpressionException
    {
        XPath xpath = XPathFactory.newInstance().newXPath();

        XPathExpression xpathExpr = xpath.compile(searchString);

        NodeList nl = (NodeList)xPathExpr.evaluate(
            _exception.getFaultDetails()[0].getParentNode(), XPathConstants.NODESET);

        return nl;
    }

    /**
     * Convert a SoapException into a CognosBIException string.
     *
     * This is the same as creating a CognosBIException and calling
     * its ToString() method.
     *
     * @param ex The AxisFault to format.
     * @return A string representation.
     */
    static public String convertToString(AxisFault ex)
    {
        CognosBIException exception = new CognosBIException(ex);
        return exception.toString();
    }
}

```

Handling an IBM Cognos biBusHeader Exception (Java)

The following code shows you how to extract the CAM element of the [bibus](#) » [biBusHeader](#) class. To see this code in context, view the Java sample `Common\BiBusHeaderException.java`:

```

import org.apache.axis.client.Stub;
import org.apache.axis.message.SOAPHeaderElement;
import javax.xml.namespace.QName;

import com.cognos.developer.schemas.bibus._3.BiBusHeader;
import com.cognos.developer.schemas.bibus._3.CAMException;
import com.cognos.developer.schemas.bibus._3.Message;
import com.cognos.developer.schemas.bibus._3.PromptInfo;
import com.cognos.developer.schemas.bibus._3.ContentManagerService_PortType;
/**
 * Extract the interesting bits from a biBusHeader after a biBusHeader
 * fault.
 */
public class BiBusHeaderException
{
    private CAMException _exception = null;
    /**
     * Create a BiBusHeaderException object.
     *
     * @param cmService ContentManagerService object in use during the
     last exception.*/
    public BiBusHeaderException(ContentManagerService_PortType cmService)

```



```

    {
    // Pull the CAM exception out of the biBusHeader.
    // BiBusHeader bibus_header =
    // ((Stub)cmService).getHeaderObject("", "biBusHeader");
    // BiBusHeader bibus = (BiBusHeader)bibus_header.getObjectValue();
    // _exception = bibus.getCAM().getException();
    // try {
    //     SOAPHeaderElement temp =
    //     ((Stub)cmService).getResponseHeader(
    //         "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader");
    //     BiBusHeader bibus_header =
    //     (BiBusHeader)temp.getValueAsType(new QName(
    //         "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader"));
    //     ((Stub)cmService).setHeader(
    //         "http://developer.cognos.com/schemas/bibus/3/", "biBusHeader",
    //         bibus_header);
    //     }
    //     catch(Exception e) {
    //     }
    // }
    /**
    * Get the Severity string from this BiBusHeaderException.
    *
    * @return The Serverity string (a severityEnum in string form).
    */
    public String getSeverity()
    {
        return new String(_exception.getSeverity().toString());
    }
    /**
    * Get the errorCodeString from this BiBusHeaderException.
    *
    * @return The errorCodeString.
    */
    public String getErrorCode()
    {
        return new String(_exception.getErrorCodeString());
    }
    /**
    * Get the details (messageString), if any, from this
    BiBusHeaderException.
    *
    * @return An array of strings containing the detail messages.
    */
    public String[] getDetails()
    {
        Message msg[] = _exception.getMessages();
        if(msg == null)
        {
            return new String[] {"null"};
        }
        String retval[] = new String[msg.length];
        for (int idx = 0; idx < msg.length; idx++)
        {
            retval[idx] = new String(msg[idx].getMessageString());
        }
        return retval;
    }
    /**
    * Get the promptInfo (and useful captions/displayObjects inside) to
    * facilitate prompting the user, if this is a recoverable exception.
    *
    * @return The promptInfo object from the exception.
    */
    public PromptInfo getPromptInfo()
    {
        return _exception.getPromptInfo();
    }
    /**
    * Convert this BiBusHeaderException into a string for printing.
    *
    * @return A string representation of the BiBusHeaderException.
    */

```

```

*/
public String toString()
{
    StringBuffer str = new StringBuffer();
    str.append("Severity      :").append(getSeverity()).append("\n");
    str.append("ErrorCode   :").append(getErrorCode()).append("\n");
    str.append("Details      : \n");
    String details[] = getDetails();
    for (int i = 0; i < details.length; i++)
    {
        str.append("\t").append(details[i]).append("\n");
    }
    return str.toString();
}
/**
 * Convert a biBusHeader exception into a BiBusHeaderException string.
 * This is the same as creating a BiBusHeaderException and calling
 * its toString() method.
 * @param crn The Service object that experienced the exception.
 * @return A string representation.
 */
static public String convertToString(ContentManagerService_PortType
cmService)
{
    BiBusHeaderException exception = new BiBusHeaderException(cmService);
    return exception.toString();
}
}

```

C# .NET Exceptions

IBM Cognos errors are reported to C# .NET applications as `System.Web.Services.Protocols.SoapFault` exceptions. The following methods of the `SoapFault` exception contain IBM Cognos error information:

- `Message` - the exception message as a string
- `Detail` - a `System.Xml.XmlNode` object containing the exception details. This is the [bibus » faultDetail](#) class in XML form.

Use the `System.Xml.XmlNode` object's `SelectSingleNode()` method to find the IBM Cognos error code and severity. Use the `SelectNodes()` method to find the IBM Cognos error details.

If the basic `SoapFault` exception indicates that extended fault information is available, you must process the `exception` property of the CAM element found in the [bibus » biBusHeader](#) class. For more information, see the [bibus » CAM » exception](#) property.

Catching SoapFault (C# .NET)

The following code shows you how to use a `try` block to catch the `System.Web.Services.Protocols.SoapFault` exception:

```

try
{
    ...
    // Code using one or more Cognos SDK methods.
    ...
}
catch( System.Web.Services.Protocols.SoapFault cognos_exception )
{
    ...
    // Exception-handling code.
    ...
}

```

Handling an IBM Cognos Exception (C# .NET)

The following code shows you how to extract the important exception details. To see this code in context, view the C# sample SamplesCommon\SamplesExceptionHandler.cs:

```
using System;
using System.Web.Services.Protocols;
using System.Xml;
using System.Text;

namespace SamplesCommon
{
    /// <summary>
    /// Simple exception handling object for use with IBM Cognos.
    /// </summary>
    public class ExceptionHelper
    {
        private SoapException _exception = null;

        /// <summary>
        /// Create an ExceptionHelper object.
        /// </summary>
        /// <param name="ex">A SoapException thrown by a call to an
        /// IBM Cognos service.</param>
        public ExceptionHelper( SoapException ex )
        {
            _exception = ex;
        }

        /// <summary>
        /// Return the exception message.
        /// </summary>
        public string Message
        {
            get
            {
                return _exception.Message;
            }
        }

        /// <summary>
        /// Return the exception severity.
        /// </summary>
        public string Severity
        {
            get
            {
                XmlNode severityNode =
                    _exception.Detail.SelectSingleNode( "//severity" );
                if (severityNode != null)
                {
                    return severityNode.InnerText;
                }
                return "";
            }
        }

        /// <summary>
        /// Return the exception errorCode.
        /// </summary>
        public string ErrorCode
        {
            get
            {
                XmlNode errorNode =
                    _exception.Detail.SelectSingleNode( "//errorCode" );
                if (errorNode != null)
                {
                    return errorNode.InnerText;
                }
                return "";
            }
        }
    }
}
```

```

    }
}

/// <summary>
/// Return the exception messageStrings.
/// </summary>
public string[] Details {
    get {
        XmlNodeList nodes =
            _exception.Detail.SelectNodes( "//messageString" );
        string[] retval = new string[nodes.Count];
        for( int idx = 0; idx < nodes.Count; idx++ ) {
            retval[idx] = nodes[idx].InnerText;
        }

        return retval;
    }
}

/// <summary>
/// Convert this exception into a string for printing.
/// </summary>
/// <returns>A string representation of the exception.</returns>
public override string ToString() {
    StringBuilder str = new StringBuilder();

    str.AppendFormat( "Message:      {0}\n", Message );
    str.AppendFormat( "Severity:     {0}\n", Severity );
    str.AppendFormat( "ErrorCode:  {0}\n", ErrorCode );
    str.AppendFormat( "Details:\n" );
    foreach( string s in Details ) {
        str.AppendFormat( "\t{0}\n", s );
    }

    return str.ToString();
}

/// <summary>
/// Convert a SoapException into an exception string.
///
/// This is the same as creating an ExceptionHelper and calling
/// its ToString() method.
/// </summary>
/// <param name="ex">The SoapException to format.</param>
/// <returns>A string representation of the exception.</returns>
static public string ConvertToString( SoapException ex ) {
    ExceptionHelper exception = new ExceptionHelper( ex );

    return exception.ToString();
}
}
}
}

```

Handling an IBM Cognos biBusHeader Exception (C# .NET)

The following code shows you how to extract the CAM element of the [bibus » biBusHeader](#) class. To see this code in context, view the C# sample `SamplesCommon\SamplesHeaderExceptionHelper.cs`:

```

using System;
using System.Text;
using cognosdotnet_2_0;

namespace SamplesCommon
{
    /// <summary>
    /// Extract the interesting bits from a biBusHeader after a
    biBusHeader
    /// fault.
    /// </summary>

```

```

public class HeaderExceptionHandler
{
    private CAMException _exception = null;

    /// <summary>
    /// Create a HeaderExceptionHandler object.
    /// </summary>
    /// <param name="crn">The contentManagerService1 object in use
    /// during the last exception.</param>
    public HeaderExceptionHandler( contentManagerService1 cmService )
    {
        // Pull the CAM exception out of the biBusHeader.
        _exception = cmService.biBusHeaderValue.CAM.exception;
    }

    /// <summary>
    /// Get the Severity string from this biBusHeader exception.
    /// </summary>
    public string Severity {
        get {
            return _exception.severity.ToString();
        }
    }

    /// <summary>
    /// Get the errorCode string from this biBusHeader exception.
    /// </summary>
    public string ErrorCode {
        get {
            return _exception.errorCodeString;
        }
    }

    /// <summary>
    /// Get the details (messageString) from this biBusHeader exception.
    /// </summary>
    public string[] Details {
        get {
            string[] retval = new string[_exception.messages.Length];

            for( int idx = 0; idx < _exception.messages.Length; idx+
+ ) {
                retval[idx] = _exception.messages[idx].messageString;
            }

            return retval;
        }
    }

    /// <summary>
    /// Get the promptInfo (and useful captions/displayObjects inside)
to
    /// facilitate prompting the user, if this is a recoverable
exception.
    /// </summary>
    public promptInfo PromptInfo {
        get {
            return _exception.promptInfo;
        }
    }

    /// <summary>
    /// Convert this biBusHeader exception into a string for printing.
    /// </summary>
    /// <returns>A string representation of the biBusHeader exception.</
returns>
    public override string ToString() {
        StringBuilder str = new StringBuilder();

        str.AppendFormat( "Severity: {0}\n", Severity );
        str.AppendFormat( "ErrorCode: {0}\n", ErrorCode );
        str.AppendFormat( "Details:\n" );
    }
}

```

```

        foreach( string s in Details ) {
            str.AppendFormat( "\\t{0}\\n", s );
        }
        return str.ToString();
    }

    /// <summary>
    /// Convert a biBusHeader exception into a string.
    /// This is the same as creating a HeaderExceptionHandler and
calling    /// its ToString() method.
    /// </summary>
param>    /// <param name="ex">The Service object that threw the exception.</
returns>    /// <returns>A string representation of the biBusHeader exception.</
    static public string ConvertToString( contentManagerService1
cmService ) {
        HeaderExceptionHandler exception = new
HeaderExceptionHandler( cmService );

        return exception.ToString();
    }
}
}
}

```

Language-specific Naming Conventions

This section includes information about naming conventions that are used with various programming languages with the IBM Cognos Software Development Kit.

The IBM Cognos Java Toolkit

To conform with Java programming conventions, the names of BI Bus API classes are modified so that the first character in the name is uppercase.

The name of the `ibibus » package` class conflicts with a Java keyword. Therefore, the class name in Java is prefixed with an underscore (`_`) so that the name becomes `_package`.

C# with the .NET Framework

In C# .NET, class names that conflict with C# keywords such as `namespace` are generated with modified names.

Recommendations for URLs

Depending on your goals, some or all of the following URL-programming recommended practices may apply to your situation:

- Ensure that your URLs do not include spaces. For example, if you use JavaScript for a post declaration, you must convert any spaces into the `%20` form. (If you enter values using the `get` technique, the Web browser handles this encoding for you.)
- Use the equal sign (=) followed by a single-quotation-mark-enclosed space when submitting empty form variables to accommodate the notational requirements of all web servers. For example, to end a complex type array in a post declaration, use the following syntax: `...name='EA' value=' '/>`
- Use `&backURL=` syntax to specify the URL location to return to, when users click **Return** in their output window.

Tip: To avoid launching a new browser window, you can specify a target name of `"_self"` as an attribute of the `<a>` anchor tag.

URL Encoding of Special Characters

Content Manager can process all characters in object names and search paths, without restrictions. However, if you reference objects or content store locations in your URL commands, you must replace illegal characters with their hexadecimal code equivalents, preceded by a percent sign. This is known as URL encoding.

The URL sample programs automatically substitute valid URL-encoded equivalents for the special characters used after the method call. This is a requirement to comply with the HTML 4.01 specification recommended by W3C, and ensures that method calls in URLs are correctly interpreted by IBM Cognos. For more information about HTML 4.01, see the W3C Web site.

To avoid illegal characters in a URL command, replace as follows:

- For asterisks, which are reserved for indicating all children of a search path node, use %2A.
- For colons, which are reserved for separating an axis from a name, such as the ":Anonymous" CAMID, use %3A.
- For periods, which are reserved for object selection expressions, use %2E.
- For square brackets, which are reserved for delineating predicate expressions, use %5B for [and %5D for].
- For slashes, which are reserved as path separators, use %5C for \ or %2F for /.
- For percent signs, which are reserved for encoding the above illegal characters, use %25.

The following code provides an example of URL encoding. To see more examples of URL encoding, examine the addresses that appear when you click the embedded links (or command buttons) in the URL ASP sample files.

```
&method=execute&
m_obj=%2fcontent%2fpackage%5b%40name%3d%27G0%20Sales%20and%20Retailers%27%5d%2f
folder%5b%40name%3d%27Documentation%20Report%20Samples%27%5d%2freport%5b%40
name%3d%27Order%20Product%20List%27%5d
```

URL Security

Because URL commands can bypass IBM Cognos access controls, we recommend that you fully discuss security issues with your administrator before implementing them.

Browser Cookie Support

If a user logs on before issuing a URL request, the browser cookie already contains their IBM Cognos Access Manager passport ID. The dispatcher automatically extracts this ID and appends it to the [bibus » biBusHeader](#) class before forwarding the request to the target service provider for processing.

Anonymous Logon Support

If a user attempts to log on and no authorized passport ID is detected in their browser cookie, the Web gateway or dispatcher can enable anonymous logon for this request. Before the service request is forwarded, an anonymous ID is appended to the [bibus » biBusHeader](#) class. Anonymous logon only occurs if a [bibus » CAM](#) passport is not found in the browser cookie and if IBM Cognos is configured to support this option.

Specifying Parameter Values of Type xsdDateTime or xsdDate

If you are running a report that requires a parameter of type `xsdDateTime` or `xsdDate`, you must ensure that the parameter value you specify has the correct format.

Parameter values of type `xsdDateTime` must have the format `YYYY-MM-DDThh:mm:ss.fff` as specified in section 5.4 of ISO 8601 (also known as Coordinated Universal Time, or UTC). The date and time portions are separated by the letter T. Note that fractional seconds are optional. Examples of `xsdDateTime` values are `2001-12-31T17:35:00.000` and `2001-12-31T17:35:00`.

Parameter values of type `xsdDate` must be specified in the same format as the first portion of the `xsdDateTime` data type (YYYY-MM-DD). An example of an `xsdDate` value is 2001-12-31.

Replacing Illegal Characters in MIME Headers

You cannot use spaces and other illegal characters in any part of a MIME header, including attachment content IDs.

To ensure proper handling of MIME messages, any spaces and other illegal characters are automatically replaced with the appropriate equivalents, such as underscores, in accordance with RFC 2047 coding standards.

Interpreting Messages that Use Response.Write Format

IBM Cognos uses the more efficient short form for its `.asp` variable definitions and outputs, but reverts to the long form for Simple Object Access Protocol (SOAP) fault notifications.

To interpret error messages returned by the IBM Cognos servers, look for the `Response.Write` (long-form) equivalent. For example, if your request for `<%= [variableName] %>` fails, look for a generated fault message that refers to `Response.Write[VariableName]`.

Localizing API Elements for Non-English Locales

All phrasing in the IBM Cognos SOAP transport layer is only in English, regardless of the content or product locale.

In locales where English is not the language of business, you can translate the BI Bus API elements that are exposed to users.

Test Connections to Data Sources

Use this procedure to test a connection between a report server instance managed by a specific dispatcher and your database.

Steps

1. Set the `bibus » conversationContext » nodeID` property to the `dispatcherID` of the connection you want to test.
2. Call the `dataSource » testDataSourceConnection(connectionString, credentials)` method, passing as input parameters the connection string to your database, your username, and your password.
3. Repeat this procedure for each of the connections you want to test.

Troubleshooting

This section lists known problems you may encounter when working with the IBM Cognos Software Development Kit, and includes suggestions for resolving them.

End of line characters removed for MHT and XLWA output

When `bibus » outputEncapsulationEnum` enumeration set is set to `none` and `MHT` or `XLWA` is specified as the output format, the end of line character sequence, Carriage Return (#13) and Line Feed (#10), is normalized into a single Line Feed (#10) character within the encapsulated XML document, leaving only the Line Feed character. As many Microsoft® Windows applications cannot process these formats without the full end of line character sequence, the output is treated as invalid by consuming applications.

If you are specifying either of these formats in this manner for report output, you should employ one of the following methods to correct the XML output:

1. Introduce a post-processing step in your SDK application to replace all instances of LF characters with CRLF.

2. Set the value of the `RSVP.ATTACHMENTENCODING.BASE64EXTENDED` advanced setting to `true`. This results in MHT and XLWA output to be base64 encoded. Your application would then need to decode the output appropriately before use.

What's new

New in Version 10.2.1 – “End of line characters removed for MHT and XLWA output” on page 1833

This topic was added.

Long-running Processes Do Not Complete

IBM Cognos components operate asynchronously, and some requests may take longer than expected. If you encounter problems during long-running processes, you may have to alter the default timeout settings.

To resolve this type of problem, see *Advanced settings configuration*.

Data Source Connections are Ambiguous

You can set up your batch report jobs to run against different IBM Cognos data sources, organized by geography or some other logical grouping, and specifying multiple data source connections and data source signons for each. If you do, when the reports run IBM Cognos issues a prompt asking you or your application to resolve the ambiguous connection, and then any ambiguous data source signons for that connection. To avoid this problem, you can explicitly list in the job credentials which data source you are using as shown in the following example.

```
<credential>
  <dataSourceConnection>
    CAMID("&quot;:&quot;);
    /dataSource[@name='gosales']
    /dataSourceConnection[@name='gosales']
  </dataSourceConnection>
  <dataSourceSignon>
    CAMID("&quot;:&quot;);
    /dataSource[@name='gosales']
    /dataSourceConnection[@name='gosales']
    /dataSourceSignon[@name='sa']
  </dataSourceSignon>
  <username/><password/>
</credential>
```

UpdateMetadata Method Fails Unless the IBM Cognos Software Development Kit is Installed On The Server

Before you can call the `metadata » updateMetadata(request)` method, you must install the IBM Cognos Software Development Kit on the IBM Cognos Analytics server. For information about installing the IBM Cognos Software Development Kit, see the *IBM Cognos Software Development Kit Installation and Configuration Guide*.

Using updateMetadata With Multiple Servers

If you are running multiple metadata servers, when you call the `metadata » updateMetadata(request)` method, the file location specified in the `model` attribute of the `mdprovider` request must be available to all the metadata servers.

Problems Encountered with Object Retention in Content Manager

Two problems can arise if you are not careful in setting up the retention rules for your IBM Cognos objects:

- Setting the `bibus » retentionRule » maxObjects` property object to 0 disables retention processing. Where possible, setting its value to 0 should be avoided as this can result in unpredictable content store growth, which can degrade overall system performance.
- If you base the retention of your executable objects on number of occurrences instead of the lifespan of the content store object, you may trigger unwanted deletions if your schedule changes.

To avoid these problems, do the following:

- When deploying, specify a `maxObjects` property of 1 to avoid overloading your database with multiple deployment histories, especially if these routinely encompass the entire content store.
- Ensure that the retention duration for each object is either greater than or equal to the duration of the job that executes it. Otherwise, a purge action may inadvertently remove the history for part of a job before the history for the rest of the job is removed, creating unwanted gaps.
- For best results, the choice between retention methods (number of occurrences versus time-based) should be driven by business needs.

SAP BW Issues Error Messages

When using an SAP BW data source, you may encounter the following error:

```
BAP-ERR-0002 BAPI error occurred in function module BAPI_MDDATASET_CHECK_SYNTAX.
Error occurred when starting the parser.
```

This error is usually an indication of an overloaded SAP BW server. Possible workarounds are to restart the IBM Cognos server or to close all open connections from the SAP BW Administrator Workbench.

Problems Using the Java Print Sample on Windows

The Java toolkit samples include a print sample, which has methods to add a printer, print a specified report, and so on.

On a Windows installation, the IBM Cognos Windows service is configured by default to run using a Local System Account. To use a shared network resource, such as a network printer, the service must be configured to log on as a valid network account. Otherwise, you cannot use network printers with the toolkit print samples. Also, to administer printers, you must have execute permissions for the Administration secured function, and you must have write permissions for the Cognos namespace.

Report Output in CSV Format Contains an Extra Character

When you run a report that has the run option that specifies output encapsulation set to none, and the output format set to CSV, the output you retrieve contains an extra character. The output is in standard Unicode UTF-16 Little Endian character format. The extra character in the output is the UTF-16 Byte Order Mark (FF FE). You must write code to remove the extra character if your programming language does not automatically remove the UTF-16 Byte Order Mark.

Base64 Decoding XLS or singleXLS Formats Creates Invalid Files

Earlier versions of IBM Cognos returned base64-encoded output in responses to requests when the specified output format was [CSV](#), [PDF](#), [XLS](#), or [singleXLS](#). Output in the [XLS](#) and [singleXLS](#) formats is no longer base64 encoded when it is returned in a response because these are not binary formats. If you have an application that decodes base64-encoded output, remove or disable this functionality for [XLS](#) and [singleXLS](#) formats.

Note that this does not apply to output returned in an attachment. An attachment is returned if the value of the `attachmentEncoding` option is [MIME](#) or [MIMECompressed](#).

IBM Cognos Analytics Reports Run More Slowly than Expected

If you upgrade from a previous release of IBM Cognos Analytics without upgrading your report specifications to the new schema, the report service may take longer than expected to run your queries and reports. This is because the report service must repeatedly upgrade to the new schema, every time you run these reports.

To optimize performance, we recommend that you write back the upgraded report specification to Content Manager, using one of the following methods:

- For a one-time upgrade, load the query or report into the appropriate IBM Cognos Analytics authoring tool and then save it.
- For a batch job, use the BI Bus API to achieve the same result. Use the `asynch » run(objectPath, parameterValues, options)` method with the `bibus » contentTask` class to upgrade objects in the content store. Use the `upgradeClasses`, `upgradeClassesContext`, and `upgradeClassesReferenceContext` options to specify which objects to upgrade. For information about Content Manager, tasks, see [“Maintain the Content Store”](#) on page 64.

Note: You may not want to upgrade your report specifications if you have an application that manipulates report specifications based on the old schema. You may have to accept poorer performance until you have the time and resources to update your application.

Unable to View Metadata in Query Studio After Adding a Query

This occurs if you use the `content » add(parentPath, objects, options)` method to add a query to a parent that is not the associated `package`. In previous versions of IBM Cognos ReportNet, to ensure that the correct metadata was used when adding a query using the `content » add(parentPath, objects, options)` method, it was necessary to set the `metadataModel` property of the `bibus » authoredReport` class. In IBM Cognos, the functionality has changed so that the report service searches a number of properties to determine the correct metadata when it runs the query. It first examines the `modelPath` element of the report specification for the search path of the published Framework Manager model. If this element is not set, the report service then examines the `metadataModel` property of the `bibus » authoredReport` class. If the metadata source still cannot be determined, then the `metadataModelPackage` property of the `bibus » authoredReport` class is used. For this reason, you must ensure that your application correctly handles adding queries, because the metadata specified in the report specification is always used if the `modelPath` element is set, regardless of whether the `metadataModel` or `metadataModelPackage` properties are also set.

If your application adds queries using the `content » add(parentPath, objects, options)` method, do the following:

- set the `modelPath` element of the report specification to the search path of the correct target metadata.

Client Applications Cannot Connect to IBM Cognos

In IBM Cognos ReportNet, client applications could connect to the gateway. In IBM Cognos Analytics, the gateway is configured by default to use the following internal dispatcher URL:

```
http://localhost:9300/p2pd/servlet/dispatch/ext
```

This causes it to block requests from client applications.

Depending on your network configuration, modify your application to do one of the following:

- connect directly to the dispatcher using the Internal Dispatcher URI (`http://localhost:9300/p2pd/servlet/dispatch`)
- connect to a dedicated gateway that is configured to connect to the dispatcher using the Internal Dispatcher URI

Note: Do not change your main gateway to use the Internal Dispatcher URI. Doing so reduces the security of the IBM Cognos Analytics portal and studios.

IBM Cognos Portal Services Extended Applications Portlet Not Working

When you try to select an application in the IBM Cognos extended applications portlet, you may see the following error message:

SDK-ERR-0001 Cannot read the file 'applications.xml'. Make sure that the Cognos Extended Applications Portlet is installed and configured properly.

This problem can be caused by one of the following:

- The IBM Cognos Software Development Kit is not installed.
- The IBM Cognos Software Development Kit is not properly configured.
- The application index specifies the wrong location for the entry JSP file.

IBM Cognos Software Development Kit is not installed

For information about installing the IBM Cognos Software Development Kit, see the IBM Cognos Software Development Kit *Installation and Configuration Guide*.

IBM Cognos Software Development Kit is not Properly Configured

Configure the IBM Cognos Software Development Kit to access the applications index.

Steps

1. Start IBM Cognos Configuration.
2. In the **Explorer** window, under **Environment**, click **Cognos Portal Services**.
3. Ensure that the location specified for the applications.xml file is valid. Use an absolute path.

Wrong Location Specified in Application Index

For each application, the applications.xml file contains a URL that specifies the location of the entry JSP file for the application. By default, the file defines each location as localhost:9300. This location works only if the IBM Cognos server runs on port 9300 and the IBM Cognos Software Development Kit is installed on the same computer. Otherwise, you must specify an absolute path for each application.

Steps

1. Open the applications.xml file in a text editor.
2. For each JSP application listed, modify the URL to replace localhost:9300 with the server name and port used to run the JSPs.
3. Save the applications.xml file.
4. If IBM Cognos is part of a distributed installation with multiple dispatchers, update all instances of the applications.xml file.

All instances of the applications.xml file must be identical.

A Namespace with a defaultName Property Cannot be Updated

If an update is attempted on a `bibus » namespace` object with the `defaultName` property populated, the following error message is issued.

CM-REQ-4021 The property "defaultName" for the namespace "Default" can only be updated using the configuration tool.

You must use IBM Cognos Configuration to configure `bibus » namespace` objects. These objects are treated differently from other objects in the content store.

IBM Cognos Access Manager will Attempt to Authenticate a User who Runs a Report Associated with an Unauthenticated Data Source

An authentication error can occur if a client application attempts to run a report that is associated with a data source that is configured to use a signon from an external namespace. When the user is not authenticated into the referenced namespace, the IBM Cognos server attempts to authenticate the user in the namespace.

If this authentication attempt fails, a [bibus](#) » [CAMEException](#) is raised and the application must log the user into the additional namespace before rerunning the report.

Multithreaded Java Client Causes Deserializer Not Found Error

If a thread in a multithreaded Java application sends a request before the Axis objects are properly initialized, it will cause a deserializer not found error.

To prevent this error, call a BI Bus API method such as [authentication](#) » [logon\(credentials, roles\)](#) from the main thread before allowing other threads to send requests.

For more information about writing multithreaded applications, see [“Recommendation - Managing biBusHeader Objects in Multithreaded Applications”](#) on page 120.

Enabling Apache Axis Logging

You can enable Apache Axis logging in your Java applications.

Steps

1. Modify `install_location\webapps\p2pd\WEB-INF\classes\log4j.properties`.

- To write logging data to the file `install_location/logs/axis.log`, modify `log4j.properties`, with the modified lines indicated.

```
### Axis Specific ###
```

```
# axis only logs at org.apache.axis level, so turn that off
```

```
log4j.logger.org.apache.axis=DEBUG, LOGFILE <--MODIFIED
```

```
...
```

```
### WSIF Specific ###
```

```
...
```

```
# LOGFILE is set to be a File appender using a PatternLayout.
```

```
...
```

```
log4j.appender.LOGFILE.File=../../../../../logs/axis.log <--MODIFIED
```

```
log4j.appender.LOGFILE.Append=true
```

```
log4j.appender.LOGFILE.Threshold=DEBUG <--MODIFIED
```

```
...
```

- To write logging data to the console, modify `log4j.properties`, with the modified lines indicated.

```
### Axis Specific ###
```

```
# axis only logs at org.apache.axis level, so turn that off
```

```
log4j.logger.org.apache.axis=DEBUG, CONSOLE <--MODIFIED
```

```
...
```

```
### WSIF Specific ###
```

```
...
# CONSOLE is set to be a ConsoleAppender using a PatternLayout.
...
log4j.appender.CONSOLE.Threshold=DEBUG <--MODIFIED
...
```

2. Include the location of `log4j.properties` (`install_location\webapps\p2pd\WEB-INF\classes`) in the CLASSPATH of your Java application, or include the `log4j.properties` file in your project.
3. Include `install_location\sdk\java\lib\log4j-1.2.8.jar` in the CLASSPATH of your Java application.

Using Axis Attachment Support in Java Applications

Java applications can use the attachment support included in the package `com.cognos.org.apache.axis.attachments`.

When including an attachment in a SOAP request from an application to the server, the character string `[attribute]` must be prepended to the attachment content ID so that the content ID is recognized as an attachment reference, and not as a character string.

For example, if the content ID is `cid:228C08BCF81B84BD3B23C34B01F17261`, then when setting the value of the content ID, it must be specified as `[attribute]cid:228C08BCF81B84BD3B23C34B01F17261`.

Unable to View Running Reports or Jobs

You may be unable to view the status or delete reports and jobs that are running or pending. This occurs because Content Manager may delete the history for reports or jobs before they finish running.

For example, you select to keep five occurrences of the run history for a report. The report is scheduled to run every minute, but takes more than five minutes to run. After five minutes elapse, there are more than five pending reports and therefore only the five most recent are kept.

Because histories may have been deleted, it may appear that there are no reports or jobs currently running. In fact, the reports and jobs continue to run, regardless of whether they have retained their run history.

Value `keyTransformation of Enumeration Set validateSeverityEnum Only Partially Works`

If you set the `bibus » validateSeverityEnum » keyTransformation` value, you will get back all errors, warnings, and information messages, but you may get little or no information back regarding the transformation steps from the report specification to the native query sent to the data source. While the documented description for the `keyTransformation` information is correct, the feature is only partially implemented at present and may not return data.

triggerName Issues

This section contains issues related to the `triggerName` parameter.

A `triggerName` Containing the Question-Mark Symbol (?) may be Misinterpreted

The event `» trigger(triggerName)` method has been modified so that a question-mark (?) in the `triggerName` property will be interpreted as being followed by a list of folder search path strings.

If you have a `triggerName` property that contains a question-mark (?) and is not followed by a list of folder search path strings, change it to remove the question-mark.

A triggerName may not Contain the Ampersand Symbol (&) in a Folder Name

The `event » trigger(triggerName)` method can receive a list of folder search path strings as part of the `triggerName` property. For example,

```
<trigger-name>?f=<folder-search-path>&f=<folder-search-path>...f=<folder-search-path>
```

The list of folder search paths is not parsed correctly when an ampersand (&) appears in the folder name. For example,

```
trigger("DAILY_REPORT?f=/content/package[@name='test']/  
folder[@name='sales&revenue']")
```

To avoid this issue, URL-encode the folder and package name. The preceding example would appear as follows:

```
trigger("DAILY_REPORT?f=%2fcontent%2fpackage%5b%40name%3d'test'%5d%2f0-width-  
spacefolder%5b%40name%3d'sales%260-width-spacerevenue'%5d")
```

The Caption of an Unrecoverable Error is not Displayed in the Browser

Error messages take precedence over captions. If an error message and a caption are added to an unrecoverable error, only the error message will be displayed. If an error message is not added and a caption is present, then the caption will be displayed.

To work around this issue, add the content of the caption to the error message, or add only a caption and not an error message to an unrecoverable error.

Using the selectRoles Method has no Effect

When executing the `authentication » selectRoles(roles)` method, the security roles specified in the `CAMPassport` are not changed or updated.

The Logged Information for Multiple Report Runs are Collapsed under the Same Request Id

When executing a `asynch » run(objectPath, parameterValues, options)` request for multiple reports in a single session, the `bibus » conversationContext » id` property is not refreshed. To avoid this issue when running multiple reports in a single session, clear the `bibus » conversationContext` in the `bibus » biBusHeader` when you send the `asynch » run(objectPath, parameterValues, options)` or `asynch » runSpecification(specification, parameterValues, options)` request. Do not re-use the same conversation context to run multiple reports.

Deprecated Methods may Timeout before Returning a Response

When using deprecated synchronous methods, your connection may timeout before a response is returned from the IBM Cognos server.

To avoid this timeout, use the equivalent asynchronous method instead.

Performance Problems in Java Applications due to Serialization of many Items with xs:nil value of true

Java applications may take a long time to run due to the serialization of a large number of items that have the `xs:nil` attribute value of `true`.

To solve this issue in applications that use Axis 1.4, add the following code (for each service used) to the application:

```
AxisEngine e = <sdkServiceLocator>.getEngine();
e.setOption(AxisEngine.PROP_SEND_NIL, Boolean.FALSE);
```

To disable suppression of null values, change the setOption setting from Boolean.FALSE to Boolean.TRUE.

An example of the code to suppress elements in contentManagerService requests is shown here.

```
import org.apache.axis.AxisEngine;
...
private ContentManagerService_ServiceLocator cmServiceLocator = null;
private ContentManagerService_PortType cmService = null;
cmServiceLocator = new ContentManagerService_ServiceLocator();
cmService = cmServiceLocator.getContentManagerService(serverURL);
...
cmServiceLocator = new ContentManagerService_ServiceLocator();
cmService = cmServiceLocator.getContentManagerService(serverURL);
...
AxisEngine e = cmServiceLocator.getEngine();
e.setOption(AxisEngine.PROP_SEND_NIL, Boolean.FALSE);
```

With this modification, SOAP elements like the following will be suppressed:

```
<dataEncoding xsi:type="ns5:encodingEnum" xsi:nil="true"/>
```

Use of SystemMetricThresholds in the Object Model Differs from the Documentation

The way that SystemMetricThresholds are created differs from the standard SDK methods. The systemMetricThresholds class and the systemMetricEnum enumeration introduced for this release to support this feature cannot be used as documented in the *IBM Cognos Software Development Kit Developer Guide*.

ui.drillThroughTargetParameterValues Not For External Use

When using the URL API in IBM Cognos Analytics to launch reports, previous documentation indicated that the ui.drillThroughTargetParameterValues could be used to specify values to fulfil a prompt as part of a drill through request. This parameter is not intended for that purpose and is for use by internal IBM Cognos components only.

To specify values to satisfy prompts, pass the prompt names and values as additional arguments to the URL using the following form: p_Year=2005

For example, to specify the Year prompt value for a report called *Revenue By Product Brand*, use:

```
http://localhost/ibmcognos/bi/v1/disp?b_action=cognosViewer&ui.object=/
content/package[@name='Sales and Marketing (cube)']/
folder[@name='Reporting Report Samples']/
/report[@name='Revenue by Product Brand (2005)']
&ui.action=run&p_Year=2005
```

Application Development Guidelines

This section lists recommended guidelines to follow when developing applications with the IBM Cognos Software Development Kit.

Recommendation - Managing Long-Running or Resource-Intensive Tasks

Read this section to learn how to take advantage of the built-in capabilities of the monitor service ([monitorService](#)) to manage queues for all server instances.

You should follow this recommendation if you have tasks that are either long-running, system-resource intensive, or both, and you wish to ensure that your tasks do not interfere with the pre-configured load-balancing and queue management functionality of IBM Cognos.

Call the [monitorService](#) instead of a target service to allow the monitor service to manage tasks across dispatchers. You can specify the [background](#) option in the request if you do not wish to maintain the asynchronous conversation in your application.

The monitor service monitors activity to determine the capacity of services managed by a [dispatcher](#) to process requests.

A request is sent to the [dispatcher](#) from the application queue based on activity information collected by the [monitorService](#) in order to maximize the use of resources. For example, if a service instance that belongs to dispatcher "A" has more available capacity than an instance that belongs to dispatcher "B", the [monitorService](#) forwards the request to dispatcher "A".

If you send a request directly to a service, you cannot take advantage of this functionality and the monitor service is normally not involved in managing the request. The [dispatcher](#) instance for the target service still has its own queue management and basic load-balancing, but it is local to that [dispatcher](#) instance only. As a result, resources may not be used as effectively as they could be.

Advantages of calling the monitor service directly include the following:

- The monitor service queues your request and forwards it to a target service with available capacity. If a server or the target service fails, the request is maintained in the queue and can be re-submitted.
- A [bibus](#) » [history](#) object is created.

Disadvantages of calling the monitor service directly include the following:

- Creating a large number of [bibus](#) » [history](#) objects can be resource intensive.

For example, if you run a large number of small reports, each run request requires that a [bibus](#) » [history](#) object be created.

- Your request may not be processed as quickly.

Passing your request directly to the monitor service instead of the target service requires an additional processing step as well as additional communications with the content store to create a [bibus](#) » [history](#) object. This may not always be more efficient if the available services are performing at capacity, or if there is only one service instance that can satisfy the request.

Advantages of calling the target service directly include the following:

- Your request may be processed more quickly.
- The overhead of creating a [bibus](#) » [history](#) object is avoided, if a [bibus](#) » [history](#) object or the ability to re-submit a failed request is not required.

Disadvantages of calling the target service directly include the following:

- System resources may not be fully utilized as the load-balancing functionality of the monitor service is not being taken advantage of.
- Overall performance could be affected as the request cannot be managed beyond the target service's local [dispatcher](#).
- A [bibus](#) » [history](#) object may not be created, depending on the type of request. Without a [bibus](#) » [history](#) object, the task cannot be re-submitted if a system or service failure occurs.

For more information about retrying failed tasks, see [“Retrying Tasks” on page 72](#).

Recommendation - Managing `biBusHeader` Objects in Multithreaded Applications

Read this section to learn about managing `biBusHeader` objects in multithreaded applications that send requests to IBM Cognos.

You should follow this recommendation if more than one thread in your application sends requests to IBM Cognos Software Development Kit. It is important to manage `biBusHeader` objects because clients and IBM Cognos services use them to track information related to specific requests. If one thread modifies a `biBusHeader` object while it is being used by another thread, it can cause exceptions or inaccurate results.

There are several approaches to managing `biBusHeader` objects that you can use in your applications.

- Use one connection for each thread, or create a connection pool.

This is the simplest but least efficient solution to implement.

- Share a single connection among multiple threads, but synchronize access to it when the `biBusHeader` object is used. For example, if one thread is running a report and paging through it, another thread must wait to run a different report.

This solution is more efficient, but more complicated to implement.

- Use the IBM Cognos WSDL file to generate your code and modify it to manage the header for multiple threads.

This approach can be used in Java programs, with Axis calls.

This is the most efficient method, but it is also the most complicated. It may be useful in a utility that performs a large number of requests.

Chapter 10. Upgrading SDK applications

To take advantage of new features in IBM Cognos Analytics, upgrade your IBM Cognos Software Development Kit applications to comply with the latest version. Some features of previous releases are deprecated in a current release, and will not be available in future releases. You can make minor changes so that your existing SDK applications can function with a current release, however, we recommend that you fully upgrade your SDK applications to the latest version if possible.

Before you can upgrade your SDK applications, you must upgrade your server software from the previous version and install the IBM Cognos Software Development Kit.

General Guidelines for Upgrading your Applications

To upgrade your SDK application, do the following:

- Create a backup of your SDK application source code.
- Install and upgrade IBM Cognos Analytics.

For instructions, see the *IBM Cognos Analytics Installation and Configuration Guide*.

Important: If your application creates, modifies, or saves report specifications, do not upgrade your report specifications when prompted. In addition, do not save your reports until your application can be upgraded to comply with the IBM Cognos report specification schema.

- If your application creates, modifies, or saves report specifications, modify the access permissions for your reports to prevent your users from inadvertently saving them with the new report specification schema. For information about securing your reports based on a group or role, see the *IBM Cognos Analytics Administration and Security Guide*.
- Install the IBM Cognos Software Development Kit.

For instructions, see the *IBM Cognos Software Development Kit Installation and Configuration Guide*.

- If your SDK application creates, modifies, or saves report specifications, do the following:
 - Make the code changes required for your application to correctly manipulate the IBM Cognos version of the report specifications.
 - Upgrade the reports in the content store by opening each report and saving it. You can do this manually or you can use the report upgrade sample in the folder *installation_location/sdk/java/ReportUpgrade/* to upgrade all the reports in a selected folder. For information about using this sample, see the *Java_ReportUpgrade_Explains.html* file that is included with the sample.
 - Update the access permissions for the upgraded reports so that your users can access them normally.

Using What's New Appendices to Review Changes

The What's New appendices describe changes to the IBM Cognos Software Development Kit methods, classes, and other objects introduced in each successive release of IBM Cognos Analytics. You should review these appendices before upgrading your applications.

Upgrading to IBM Cognos Analytics Version 10.2.0

If you are upgrading to IBM Cognos Analytics Version 10.2.0, you must make the following changes to your SDK applications.

Changes to .NET Framework support

Changes to the .NET interface to support .NET Framework 4.0 have been implemented in IBM Cognos Analytics version 10.2.0. Your .NET project must now refer to the `cognosdotnet_10_2.dll` file. The Cognos namespace in the `cognosdotnet_10_2.dll` file is `cognosdotnet_10_2`. If you use the .NET Framework, complete the following steps.

1. Replace references to `cognosdotnet_2_0.dll` with `cognosdotnet_10_2.dll`.
2. Replace references to `cognosdotnetassembly_2_0.dll` with `cognosdotnetassembly_10_2.dll`.
3. Replace references to the `cognosdotnet_2_0` namespace with `cognosdotnet_10_2`.

Upgrading to IBM Cognos Analytics Version 10.1.0

If you are upgrading to IBM Cognos Analytics Version 10.1.0, you must make the following changes to your SDK applications.

Upgrading Java Applications for Axis 1.4

The default Java toolkit in the current version (found in `installation_location/sdk/java/lib`) of the IBM Cognos Software Development Kit is based on Axis 1.4 instead of Axis 1.1, which was used in previous versions of the IBM Cognos Software Development Kit.

In Cognos Analytics v11.0 (and earlier), there is an Axis 1.1 version of the toolkit located in `installation_location/sdk-compat/java/lib`. If you have existing SDK applications that use Axis 1.1, you can continue to use them by including the .jar files from the Axis 1.1 toolkit. Alternatively, you can upgrade your applications to Axis 1.4. The required changes are described here.

Note: The Java SDK code samples shipped with the IBM Cognos Software Development Kit require using the Axis 1.4 toolkit.

Changing Port Type Class Names

The class name generated for service port types has changed from `<serviceName>_Port` to `<serviceName>_PortType`. You must now update references to these classes. For example, change references to the `ContentManagerService_Port` class to `ContentManagerService_PortType`.

Changing Getter and Setter Method Names

The getter and setter method names for the following classes have changed from `getValue()` and `setValue()` to `get_value()` and `set_value()`, respectively.

- [bibus](#) » [addressSMTP](#)
- [bibus](#) » [dataSourceCommandBlock](#)
- [bibus](#) » [guid](#)
- [bibus](#) » [metadataModelExpression](#)
- [bibus](#) » [metadataModelItemName](#)
- [bibus](#) » [searchPathMultipleObject](#)
- [bibus](#) » [searchPathSingleObject](#)
- [bibus](#) » [specification](#)
- [bibus](#) » [xmlEncodedXML](#)

For example,

```
SearchPathMultipleObject spObject = new
SearchPathMultipleObject(); spObject.set_value(<search_path>);
```

may be changed to

```
SearchPathMultipleObject spObject = new
SearchPathMultipleObject();
spObject.set_value(<search_path>);
```

where `<search_path>` is a valid search path.

Moving Classes to the `org.apache.axis.constants` Package

You must change references to the `org.apache.axis.enum` and `com.cognos.org.apache.axis.enum` packages to the `org.apache.axis.constants` and `com.cognos.org.apache.axis.constants` packages, respectively.

Retrieving and Clearing `biBusHeader` Objects

In an application using the Axis 1.1 toolkit, the following code retrieves the `biBusHeader` object from a response:

```
BiBusHeader bibus = (BiBusHeader) ((Stub)cmService).getHeaderObject("",
"biBusHeader");
```

or

```
SOAPHeaderElement bibus =
getServiceContext().getImplicitHeader("http://developer.cognos.com/schemas/
bibus/3/",
"biBusHeader");
```

The following code retrieves the `biBusHeader` object from a response in an application using the Axis 1.4 toolkit:

```
SOAPHeaderElement temp =
((Stub)cmService).getResponseHeader("http://developer.cognos.com/schemas/bibus/3/",
"biBusHeader");
BiBusHeader bibus =
(BiBusHeader)temp.getValueAsType(new QName(
"http://developer.cognos.com/schemas/bibus/3/", "biBusHeader"));
((Stub)cmService).setHeader("http://developer.cognos.com/schemas/bibus/3/",
"biBusHeader", bibus);
```

In an application using the Axis 1.1 toolkit, the following code clears the header:

```
((Stub)cmService).getServiceContext().clearHeaders();
```

The following code clears the header in an application using the Axis 1.4 toolkit:

```
((Stub)cmService).clearHeaders();
```

Managing Service Headers

Your Java applications must now explicitly manage service headers when making secondary requests. The previous toolkit that used Axis 1.1 had customized methods that are no longer available.

See [“Managing Service Headers in Java Applications”](#) on page 97 for more information.

Changes to the `cam_passport` cookie

The structure of the `cam_passport` cookie has changed and its contents may change during a session. SDK applications that assume that the value of the `cam_passport` cookie is the same as the value of the `bibus » CAMPassport » id` property will no longer work and will have to be modified.

SDK applications that copy the `cam_passport` value into the `bibus » CAMPassport » id` property must be modified. These applications can do one of the following:

- Copy the CAM section from a successful authentication response into a subsequent request.
- Add the `cam_passport` value to the `cookieVars` section of the BI Bus request.
- Submit the `cam_passport` as an HTTP cookie.

Changes to the Microsoft® Excel Output Formats

The `XLS` and `singleXLS` output formats are deprecated in this version.

Change these values in your SDK applications to use `XLWA` or `spreadsheetML` instead. Run requests specifying `XLS` or `singleXLS` will fail, indicating that the output format is not supported. For backwards compatibility, reports specifying these formats in the content store are still viewable.

During upgrade or deployment, the following properties where the `bibus » outputFormatEnum` can be specified are checked for either of these values:

- `bibus » account » format` property
- `bibus » contact » format` property
- `bibus » agentDefinition` class (inherited from `bibus » baseAgentDefinition » options` property)
- `bibus » agentDefinitionView` class (inherited from `bibus » baseAgentDefinition » options` property)
- `bibus » agentTaskDefinition » options` property
- `bibus » reportDataServiceAgentDefinition` class (inherited from `bibus » baseAgentDefinition » options` property)
- `bibus » basePowerPlay8Report » options` property (and derived objects)
- `bibus » baseReport » options` property (and derived objects)
- `bibus » drillPath » options` property
- `bibus » jobDefinition » options` property
- `bibus » jobStepDefinition » options` property
- `bibus » schedule » options` property

If found, the value is automatically changed to `XLWA`.

Multi-Instance IBM Cognos Connection

Support has been added to allow IBM Cognos Connection to reference content in other installations of IBM Cognos Analytics.

This provides support for multi-version coexistence, which allows customers to stage their upgrade process by allowing two versions of IBM Cognos Analytics to be accessed through a single portal, preventing disruptions to end users.

For example, users can access IBM Cognos Analytics Version 8.4 or IBM Cognos Analytics Version 8.3 content from within the IBM Cognos Connection Version 10.1.0 portal.

It is also possible to utilize this feature to provide access to content through IBM Cognos Connection from multiple installations of IBM Cognos Analytics Version 10.1.0 in a large enterprise.

Upgrading to IBM Cognos Analytics Version 8.4

If you are upgrading to IBM Cognos Analytics Version 8.4 or later, you must make the following changes to your SDK applications.

Changes to Report Specification Validation

If your SDK application creates, modifies, or saves report specifications, you must modify your SDK application to comply with the correct version of report specification schema before you upgrade your reports. If you upgrade your reports first, your existing application cannot access the new report specifications until you modify it to comply with the report specification schema for the version of IBM Cognos 8 you are upgrading to. For information about using the new report specification schema, see [Chapter 23, “Using report specifications,” on page 1457](#).

To facilitate upgrading your reports to the schema version supported in IBM Cognos Analytics Version 8.4, the product now contains multiple versions of the report specification schemas and DTDs. The report server uses the schema version specified in the report specification to identify the version with which the report was created.

If the report specification was created with an earlier version, the server automatically upgrades the report to the latest version when you open, run, or save the report.

The location of the schemas and DTDs has changed to *installation_location/webcontent/schemas/rspec/n.o*, where n.o indicates the schema version to use. IBM Cognos Analytics Version 8.4 uses the 6.0 DTD or schema.

Upgrading to IBM Cognos Analytics Version 8.3

If you are upgrading to IBM Cognos Analytics Version 8.3 or later, you must make the following changes to your SDK applications.

Changes to Deployment Options

If your SDK application exports data to or imports data from a deployment archive using the deprecated package deployment option, you may need to modify it to use the [import](#) and [export](#) deployment options.

To export data to an archive that will be imported using IBM Cognos Analytics Version 8.3 or higher, specify the [import](#) and [export](#) options for the [bibus » exportDeployment](#) object.

To export data to an archive that will be imported by an SDK application that still uses the package option, specify the package option for the [bibus » exportDeployment](#) object.

To import data from an archive that was created using IBM Cognos Analytics Version 8.3 or higher, specify the [import](#) and [export](#) options for the [bibus » importDeployment](#) object.

To import data from an archive that was created by an SDK application that still uses the package option, specify the package and [upgradeArchiveOptions](#) options for the [bibus » importDeployment](#) object.

Chapter 11. Schema reference

This chapter describes schemas specific to the "Cognos" namespace.

bibus

The bibus schema defines types used by the IBM Cognos Software Development Kit.

Chapter 12. Services

This chapter describes the Web services defined by IBM Cognos. The IBM Cognos architecture includes a number of services for interacting with and managing runnable objects. Services exist for each of the major components of the product. Each of the services listed in this chapter communicates using the BI Bus API.

Each service shares a set of generic methods for running objects associated with the service. Users send requests directly to the appropriate service, based on the class of the object. For example, requests to run reports can be sent to the report service or the batch report service. Requests to move objects are sent to the Content Manager.

To support this architecture, services execute tasks using a generalized asynchronous protocol. The protocol uses a consistent mechanism for passing parameters to standardize how tasks are executed. Classes and methods exist to support this consistent process of executing tasks. Asynchronous conversations are managed to optimize overall performance. For more information about asynchronous conversations and runnable objects, see [Chapter 6, “Running tasks,” on page 65](#).

A single, generic `bibus » history` class structure is shared by background tasks to record significant events during execution.

The various options specified to run tasks are also unified under a single `bibus » option` class hierarchy.

Services in Java

In Java, the behavior protocols for each of the services are defined by a set of interfaces. For each BI Bus API service, there are two corresponding Java interface definitions: the `<service>_Service` interface and the `<service>_Port` interface.

The implementation of the `<service>_Service` interface is a class named `<service>_ServiceLocator`. This locator class contains implementations for the methods named in the `<service>_Service` interface. Use the methods in the `<service>_ServiceLocator` class to acquire the port information for the associated service.

The implementation of the `<service>_Port` interface is a class named `<service>Stub`. This stub class contains the implementations for the methods named in the `<service>_Port` interface. Use the methods in the `<service>Stub` class to access the functionality provided by the service.

For each service, there is an additional class called `<service>_Type` that is used to manipulate configuration settings for the associated service.

The following table shows examples of the classes that are associated with the `reportService`.

ReportService_ServiceLocator

Implements the `ReportService_Service` interface. Use this class to retrieve the port information for the report service.

ReportServiceStub

Implements the `ReportService_Port` interface. Use this class to call methods provided by the report service.

ReportService_Type

Use this class to manipulate configuration settings for the report service.

Services in C# .NET

In C# .NET, there are two C# .NET classes for each of the BI Bus API services. These classes are named `<service>` and `<service>1`. One class contains the implementation for the service methods and the other is used to manipulate configuration settings for the associated service. Some examples are listed in the following table.

List of Services

The following is a list of available services, with the service reference and configuration class.

Table 10. Services and configuration classes

IBM Cognos service name	Service reference	Configuration class
agent service	agentService	bibus » agentService
annotation service	No published API available	bibus » annotationService
batch report service	batchReportService	bibus » batchReportService
Content Manager service	contentManagerService	bibus » contentManagerService
Content Manager cache service	No published API available	bibus » contentManagerCacheService
N/A	dataIntegrationService	bibus » dataIntegrationService
N/A	dataMovementService	bibus » dataMovementService
delivery service	deliveryService	bibus » deliveryService
N/A	dimensionManagementService	bibus » dimensionManagementService
dispatcher	dispatcher	bibus » dispatcher
N/A	No published API available	bibus » EVService
event management service	eventManagementService	bibus » eventManagementService
graphics service	No published API available	bibus » graphicsService
human task service	No published API available	bibus » humanTaskService
index data service	No published API available	bibus » indexDataService
N/A	indexSearchService	bibus » indexSearchService
N/A	indexUpdateService	bibus » indexUpdateService
job service	jobService	bibus » jobService
log service	No published API available	bibus » logService
N/A	metadataService	bibus » metadataService
N/A	No published API available	bibus » metricsManagerService
N/A	migrationService	bibus » migrationService

Table 10. Services and configuration classes (continued)

IBM Cognos service name	Service reference	Configuration class
N/A	mobileService	bibus » mobileService
monitor service	monitorService	bibus » monitorService
N/A	No published API available	bibus » planningDataService
N/A	planningAdministrationConsoleService	bibus » planningAdministrationConsoleService
N/A	planningRuntimeService	bibus » planningRuntimeService
N/A	planningTaskService	bibus » planningTaskService
N/A	powerPlayService	bibus » powerPlayService
presentation service	No published API available	bibus » presentationService
query service	queryService	bibus » queryService
relational metadata service	relationalMetadataService	bibus » relationalMetadataService
report data service	No published API available	bibus » reportDataService
report service	reportService	bibus » reportService
system service	systemService	bibus » systemService

agentService

Defines the service responsible for the execution of agents ([bibus](#) » [baseAgentDefinition](#) class), stored procedures ([bibus](#) » [storedProcedureTask](#) class), and Web service methods ([bibus](#) » [webServiceTask](#) class).

The agent service uses the [monitorService](#) to execute each task in an [bibus](#) » [agentDefinition](#).

The agent service has a set of configuration parameters defined by the properties of the [bibus](#) » [agentService](#) class.

The agent service creates an [bibus](#) » [agentState](#) object to store data that is required during the execution of the agent. This data includes the [bibus](#) » [output](#) objects containing the agent events and the [bibus](#) » [agentOutputHotList](#).

If the [bibus](#) » [agentState](#) object is deleted the agent service will re-create it, but any events reported by the agent during re-creation are considered to be new.

Example: Connecting to the agent service in Java.

```
private AgentService_ServiceLocator agentServiceLocator = null;
private AgentService_PortType agentService = null;
public static String AS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
agentServiceLocator = new AgentService_ServiceLocator();
...
```

```

try
{
    java.net.URL serverURL = new java.net.URL(AS_URL);
    agentService = agentServiceLocator.getagentService(serverURL);
    ...
}
// catch exceptions

```

Example: Connecting to the agent service in C# .NET.

```

try
{
    agentService c8AS = new agentService();
    c8AS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions

```

References

Runs instances of the following classes:

- [bibus » agentDefinition](#)
- [bibus » agentDefinitionView](#)
- [bibus » humanTask](#)
- [bibus » reportDataServiceAgentDefinition](#)
- [bibus » storedProcedureTask](#)
- [bibus » webServiceTask](#)

Uses the following method sets:

- [agent](#)
- [asynch](#)

What's new

New in Version 10.1.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1846

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/agentService/201101/>.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/agentService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/agentService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/agentService/201404/>.

batchReportService

Defines the service responsible for the execution of reports ([bibus » baseReport](#) class).

The batch report service is dedicated to running non-interactive batch reports. For example, when a report is run as part of an agent or job or is run in the background, the request is sent to the batch report service. A report is run in the background when it is scheduled, emailed, printed, saved, or run in response to an external occurrence (see [event](#) » [trigger\(triggerName\)](#)).

The batch report service has an associated set of configuration parameters that are defined by the properties of the [bibus](#) » [batchReportService](#) class. These parameters can be configured to optimize the batch report service for non-interactive use. This is the only difference between the batch report service and the report service ([reportService](#)).

This service may also make additional requests to the following IBM Cognos Analytics services when processing a request:

- [contentManagerService](#) service
- [deliveryService](#) service
- [eventManagementService](#) service
- [monitorService](#) service

Example: Connecting to the Batch Report Service in Java

```
private BatchReportService_ServiceLocator batchReportServiceLocator = null;
private BatchReportService_PortType batchReportService = null;
public static String BS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
batchReportServiceLocator = new BatchReportService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(BS_URL);
    batchReportService =
batchReportServiceLocator.getbatchReportService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Batch Report Service in C# .NET

```
try
{
    batchReportService1 c8BRS = new batchReportService1();
    c8BRS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus](#) » [analysis](#)
- [bibus](#) » [drillPath](#)
- [bibus](#) » [interactiveReport](#)
- [bibus](#) » [metadataServiceLineageSpecification](#)
- [bibus](#) » [query](#)
- [bibus](#) » [report](#)
- [bibus](#) » [reportServiceAnalysisSpecification](#)

- [bibus » reportServiceDrillThroughSpecification](#)
- [bibus » reportServiceInteractiveReportSpecification](#)
- [bibus » reportServiceMetadataSpecification](#)
- [bibus » reportServiceQuerySpecification](#)
- [bibus » reportServiceReportSpecification](#)
- [bibus » reportTemplate](#)
- [bibus » reportView](#)

Uses the following method sets:

- [asynch](#)
- [dataSource](#)
- [drillThrough](#)
- [paging](#)
- [parameter](#)
- [promptPaging](#)
- [report](#)
- [validate](#)

What's new

New in Version 8.3 – “[Parameter Method Set](#)” on page 1915

This service now implements the [parameter](#) method set.

New in Version 8.4 – “[PowerPlay 8 Integration](#)” on page 1886

This service now implements the [dataSource](#) method set.

New in Version 10.2.0 – “[Changes to IBM Cognos Analytics SOAP action for services](#)” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/batchReportService/201109/>.

New in Version 10.2.1 – “[Changes to IBM Cognos Analytics SOAP action for services](#)” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/batchReportService/201301/>.

New in Version 10.2.2 – “[Changes to IBM Cognos Analytics SOAP action for services](#)” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/>.

contentManagerService

Defines the service responsible for managing the content store. The Content Manager service also performs authentication-related activities.

The Content Manager service has an associated set of configuration parameters defined by the properties of the [bibus » contentManagerService](#) class.

For more information about Content Manager, see [Chapter 5, “Managing content,”](#) on page 55.

Example: Connecting to the Content Manager Service in Java

```
private ContentManagerService_ServiceLocator contentManagerServiceLocator =
null;
private ContentManagerService_PortType contentManagerService = null;
```



```

public static String CMS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
contentManagerServiceLocator = new ContentManagerService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(CMS_URL);
    contentManagerService =
        contentManagerServiceLocator.getContentManagerService(serverURL);
    ...
}
// catch exceptions

```

Example: Connecting to the Content Manager Service in C# .NET

```

try
{
    contentManagerService1 c8CMS = new contentManagerService1();
    c8CMS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions

```

References

Runs instances of the following classes:

- [bibus » contentTask](#)
- [bibus » exportDeployment](#)
- [bibus » importDeployment](#)

Uses the following method sets:

- [asynch](#)
- [authentication](#)
- [content](#)
- [deployment](#)

What's new

New in Version 10.1.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1846

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201101/>.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/>.

dataIntegrationService

Defines the service responsible for the execution of metrics-related tasks. You can define metrics-related tasks using the [bibus » baseDataIntegrationTask](#) class.

The data integration service has an associated set of configuration parameters defined by the properties of the [bibus » dataIntegrationService](#) class.

Example: Connecting to the Data Integration Service in Java

```
private DataIntegrationService_ServiceLocator dataIntegrationServiceLocator =
null;
private DataIntegrationService_PortType dataIntegrationService = null;
public static String DIS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
dataIntegrationServiceLocator = new DataIntegrationService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(DIS_URL);
    dataIntegrationService =
        dataIntegrationServiceLocator.getdataIntegrationService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Data Integration Service in C# .NET

```
try
{
    dataIntegrationService1 c8DIS = new dataIntegrationService1();
    c8DIS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus » dataIntegrationServiceSpecification](#)
- [bibus » metricsDataSourceETLTask](#)
- [bibus » metricsExportTask](#)
- [bibus » metricsFileImportTask](#)
- [bibus » metricsMaintenanceTask](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/`.

dataMovementService

Defines the service responsible for merging and transforming data from multiple sources into coordinated data marts. This data can then be packaged and made available to IBM Cognos applications for reporting and analysis.

For more information, see the *IBM Cognos Data Manager Data Movement Service Developer Guide*.

The data movement service has an associated set of configuration parameters defined by the properties of the [bibus » dataMovementService](#) class.

Example: Connecting to the Data Movement Service in Java

```
private DataMovementService_ServiceLocator dataMovementServiceLocator = null;
private DataMovementService_PortType dataMovementService = null;
public static String DMS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
dataMovementServiceLocator = new DataMovementService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(DMS_URL);
    dataMovementService =
dataMovementServiceLocator.getdataMovementService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Data Movement Service in C# .NET

```
try
{
    dataMovementService1 c8DMS = new dataMovementService1();
    c8DMS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus » dataMovementServiceSpecification](#)
- [bibus » dataMovementTask](#)
- [bibus » dataMovementTaskAlias](#)

Uses the following method sets:

- [asynch](#)
- [validate](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201109/`.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201301/`.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/`.

deliveryService

Defines the service responsible for guaranteed delivery of email to an external email service.

The delivery service is used by the [batchReportService](#) and the [reportService](#) to deliver report outputs.

The [agentService](#) uses the delivery service to execute [bibus](#) » [memo](#) objects that are part of an agent ([bibus](#) » [baseAgentDefinition](#) class).

The delivery service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [deliveryService](#) class.

Example: Connecting to the Delivery Service in Java

```
private DeliveryService_ServiceLocator deliveryServiceLocator = null;
private DeliveryService_PortType deliveryService = null;
public static String DES_URL = "http://localhost:9300/p2pd/servlet/
dispatch";
deliveryServiceLocator = new DeliveryService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(DES_URL);
    deliveryService = deliveryServiceLocator.getdeliveryService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Delivery Service in C# .NET

```
try
{
    deliveryService1 c8DS = new deliveryService1();
    c8DS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus](#) » [deliveryServiceSpecification](#)
- [bibus](#) » [memo](#)
- [bibus](#) » [shortcutAgentRSSTask](#)

- [bibus » shortcutRSSTask](#)
- [bibus » urlRSSTask](#)

Uses the following method sets:

- [asynch](#)
- [delivery](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/deliveryService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/deliveryService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/>.

dimensionManagementService

Defines the service responsible for dimension management.

The dimension management service acts as a gateway to the IBM Cognos Business Viewpoint application.

The dimension management service has a set of configuration parameters defined by the properties of the [bibus » dimensionManagementService](#) class.

Refer to the IBM Cognos Business Viewpoint documentation for more information.

References

Runs instances of the following classes:

- [bibus » dimensionManagementServiceSpecification](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 8.4 – “Dimension Management Service” on page 1896

This service was added.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/>.

dispatcher

Defines the entry point for IBM Cognos Analytics service requests sent by a Web server gateway or other software. The dispatcher manages a set of associated services, performing activities such as routing requests to these services, as well as managing their configuration at run time.

The dispatcher has an associated set of configuration parameters defined by the properties of the [bibus » dispatcher](#) class.

You can also manage and configure dispatchers and their associated services using the server administration tool.

Example: Connecting to the dispatcher in Java.

```
private Dispatcher_ServiceLocator dispatcherLocator = null;
private Dispatcher_PortType dispatcher = null;
public static String DIS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
dispatcherLocator = new Dispatcher_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(DIS_URL);
    dispatcher = dispatcherLocator.getdispatcher(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the dispatcher in C# .NET.

```
try
{
    dispatcher1 c8D = new dispatcher1();
    c8D.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Uses the following method sets:

- [dispatcher](#)

What's new

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201301/>.

New in Version 10.2.2 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1823](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201404/>.

Related information:

- *IBM Cognos Administration and Security Guide*

eventManagementService

Defines the service responsible for the management of events. An event is any task that has run, is currently running, or will be run at a future time.

The event management service manages the event execution queue. Events in the queue can be held for execution at a later time or removed from the queue. Events that are held can later be released for execution.

Events to be run on an irregular basis can also be manually added to the event execution queue. These events are usually on-demand tasks or tasks that are executed in response to an external trigger condition.

The [contentManagerService](#) works in conjunction with the event management service to update the event execution queue whenever a [bibus](#) » [schedule](#) is added to, updated in, or deleted from the content store.

The event management service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [eventManagementService](#) class.

Example: Connecting to the Event Management Service in Java

```
private EventManagementService_ServiceLocator eventManagementServiceLocator =
null;
private EventManagementService_PortType eventManagementService = null;
public static String EMS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
eventManagementServiceLocator = new EventManagementService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(EMS_URL);
    eventManagementService =
        eventManagementServiceLocator.geteventManagementService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Event Management Service in C# .NET

```
try
{
    eventManagementService1 c8EMS = new eventManagementService1();
    c8EMS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus](#) » [eventManagementServiceSpecification](#)

Uses the following method sets:

- [asynch](#)
- [event](#)

- [paging](#)

What's new

New in Version 8.3 – “Schedule Management” on page 1907

This service now implements the [asynch](#) method set and [paging](#) method set.

New in Version 10.1.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1846

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201101/>.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/>.

idVizService

Reserved.

References

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Support for interactive discovery and visualization” on page 1839

This service was added.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/idVizService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/>.

indexSearchService

Reserved.

Example: Connecting to the Index Search Service in Java

```
private IndexSearchService_ServiceLocator indexSearchServiceLocator = null;
private IndexSearchService_PortType indexSearchService = null;
public static String JS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
indexSearchServiceLocator = new IndexSearchService_ServiceLocator();
...
```



```

try
{
    java.net.URL serverURL = new java.net.URL(JS_URL);
    indexSearchService =
indexSearchServiceLocator.getIndexSearchService(serverURL);
    ...
}
// catch exceptions

```

Example: Connecting to the Index Search Service in C# .NET

```

try
{
    indexSearchService1 c8ISS = new indexSearchService1();
    c8ISS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions

```

References

Runs instances of the following classes:

- [bibus » indexSearchServiceSpecification](#)

Uses the following method sets:

- [asynch](#)
- [paging](#)
- [promptPaging](#)

What's new

New in Version 8.3 – [“Search – For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201301/>.

New in Version 10.2.2 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1823](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/>.

indexUpdateService

Reserved.

Example: Connecting to the Index Update Service in Java

```
private IndexUpdateService_ServiceLocator indexUpdateServiceLocator = null;
private IndexUpdateService_PortType indexUpdateService = null;
public static String IUS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
indexUpdateServiceLocator = new IndexUpdateService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(IUS_URL);
    indexUpdateService =
indexUpdateServiceLocator.getindexUpdateService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Index Update Service in C# .NET

```
try
{
    indexUpdateService1 c8IUS = new indexUpdateService1();
    c8IUS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus » indexUpdateServiceSpecification](#)
- [bibus » indexUpdateTask](#)

Uses the following method sets:

- [asynch](#)
- [indexTerm](#)
- [indexUpdate](#)

What's new

New in Version 8.3 – [“Search – For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201301/>.

New in Version 10.2.2 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1823](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/>.

jobService

Defines the service responsible for the execution of jobs ([bibus](#) » [jobDefinition](#) class).

The job service uses the [monitorService](#) to execute each step in a job.

The job service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [jobService](#) class.

Example: Connecting to the Job Service in Java

```
private JobService_ServiceLocator jobServiceLocator = null;
private JobService_PortType jobService = null;
public static String JS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
jobServiceLocator = new JobService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(JS_URL);
    jobService = jobServiceLocator.getjobService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Job Service in C# .NET

```
try
{
    jobService1 c8JS = new jobService1();
    c8JS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus](#) » [jobDefinition](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.1.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1846](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/jobService/201101/>.

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/jobService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/jobService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/jobService/201404/`.

metadataService

Defines the service responsible for querying and updating unpublished IBM Cognos Framework Manager models.

The metadata service has an associated set of configuration parameters defined by the properties of the `bibus » metadataService` class.

Example: Connecting to the Metadata Service in Java

```
private MetadataService_ServiceLocator metadataServiceLocator = null;
private MetadataService_PortType metadataService = null;
public static String MES_URL = "http://localhost:9300/p2pd/servlet/dispatch";
metadataServiceLocator = new MetadataService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(MES_URL);
    metadataService = metadataServiceLocator.getmetadataService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Metadata Service in C# .NET

```
try
{
    metadataService1 c8MDS = new metadataService1();
    c8MDS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- `bibus » metadataServiceLineageSpecification`
- `bibus » metadataServiceModelInformationSpecification`

Uses the following method sets:

- `asynch`
- `dataSource`
- `metadata`

What's new

New in Version 8.4 – “Lineage Metadata” on page 1891

This service now implements the `asynch` method set.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This service now implements the [dataSource](#) method set.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/metadataService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/metadataService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/>.

migrationService

Defines the service that provides for the migration of content to IBM Cognos.

This service can be used to perform the following migrations:

- [bibus » powerPlayReport](#) (IBM Cognos Series 7 PowerPlay) to [bibus » powerPlay8Report](#) (PowerPlay)
- [bibus » powerPlayReport](#) (IBM Cognos Series 7 PowerPlay) to [bibus » analysis](#) (IBM Cognos Analysis Studio) or [bibus » report](#) (IBM Cognos Analytics - Reporting)
- [bibus » powerPlay8Report](#) (IBM Cognos PowerPlay Studio) to [bibus » analysis](#) (IBM Cognos Analysis Studio) or [bibus » report](#) (IBM Cognos Analytics - Reporting)

The migration service has an associated set of configuration parameters defined by the properties of the [bibus » migrationService](#) class.

References

Runs instances of the following classes:

- [bibus » migrationServiceSpecification](#)
- [bibus » migrationTask](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 8.4 – “Migration Service” on page 1883

This service was added.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/migrationService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/migrationService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/>.

mobileService

Defines the service responsible for the delivery of content to mobile devices.

This service is invoked by the [reportService](#) service to deliver Cognos content to wireless devices, such as mobile phones or other wireless devices.

The mobile service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [mobileService](#) class.

Example: Connecting to the Mobile Service in Java

```
private MobileService_ServiceLocator mobileServiceLocator = null;
private MobileService_PortType mobileService = null;
public static String MBS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
mobileServiceLocator = new MobileService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(MBS_URL);
    mobileService = mobileServiceLocator.getmobileService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Mobile Service in C# .NET

```
try
{
    mobileService1 c8MBS = new mobileService1();
    c8MBS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/mobileService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/mobileService/201301/>.

New in Version 10.2.2 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1823](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/>.

monitorService

Defines the service responsible for monitoring non-interactive tasks and managing [bibus » history](#) objects in the content store. The monitoring service updates histories in the event of a catastrophic server failure. The monitoring service is also used to monitor tasks that are run in the background by the user for batch completion.

Unlike many of the other IBM Cognos Analytics services, the monitoring service is not responsible for running a task. Requests to run tasks that are sent to the monitoring service are forwarded to the service that can run the specified task. The monitoring service updates the appropriate [bibus » history](#) object based on the response from the service that is running the task.

The monitoring service is used by the [agentService](#) to run the steps that form an agent ([bibus » baseAgentDefinition](#)). The [jobService](#) uses the monitoring service to run the steps that form a job ([bibus » jobDefinition](#) class).

The monitoring service has an associated set of configuration parameters defined by the properties of the [bibus » monitorService](#) class.

Example: Connecting to the Monitor Service in Java

```
private MonitorService_ServiceLocator monitorServiceLocator = null;
private MonitorService_PortType monitorService = null;
public static String MOS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
monitorServiceLocator = new MonitorService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(MOS_URL);
    monitorService = monitorServiceLocator.getmonitorService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Monitor Service in C# .NET

```
try
{
    monitorService1 c8MS = new monitorService1();
    c8MS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Runs instances of the following classes:

- [bibus » monitorServiceSpecification](#)

Uses the following method sets:

- [asynch](#)
- [monitor](#)
- [paging](#)

What's new

New in Version 8.3 – “Schedule Management” on page 1907

This service now implements the [paging](#) method set.

New in Version 10.1.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1846

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/monitorService/201101/>.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/monitorService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/monitorService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/>.

planningAdministrationConsoleService

Reserved.

References

Runs instances of the following classes:

- [bibus » planningAdministrationConsoleServiceSpecification](#)
- [bibus » planningMacroTask](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/>.

planningRuntimeService

Reserved.

References

Runs instances of the following classes:

- [bibus » planningRuntimeServiceSpecification](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/>.

planningTaskService

Reserved.

References

Runs instances of the following classes:

- [bibus » planningTask](#)
- [bibus » planningTaskServiceSpecification](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/>.

powerPlayService

Defines the service responsible for the execution of PowerPlay reports ([bibus » basePowerPlay8Report](#) class).

The PowerPlay service has an associated set of configuration parameters defined by the properties of the [bibus » powerPlayService](#) class.

Example: Connecting to the PowerPlay Service in Java

```
private PowerPlayService_ServiceLocator powerPlayServiceLocator = null;
```

```

private PowerPlayService_PortType powerPlayService = null;
public static String PPS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
powerPlayServiceLocator = new PowerPlayService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(PPS_URL);
    powerPlayService = powerPlayServiceLocator.getpowerPlayService(serverURL);
    ...
}
// catch exceptions

```

Example: Connecting to the PowerPlay Service in C# .NET

```

try
{
    powerPlayService1 c8PPS = new powerPlayService1();
    c8PPS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions

```

References

Runs instances of the following classes:

- [bibus » powerPlay8Report](#)
- [bibus » powerPlay8ReportView](#)
- [bibus » powerPlayServiceReportSpecification](#)

Uses the following method sets:

- [asynch](#)
- [parameter](#)
- [promptPaging](#)
- [validate](#)

What's new

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This service was added.

New in Version 10.1.0 – [“New PowerPlay Service Methods” on page 1860](#)

This service now implements the [promptPaging](#) and [validate](#) interfaces.

New in Version 10.2.0 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1840](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201109/>.

New in Version 10.2.1 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1831](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201301/>.

New in Version 10.2.2 – [“Changes to IBM Cognos Analytics SOAP action for services” on page 1823](#)

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/>.

queryService

Defines the service responsible for executing dynamic query mode queries.

The query service has an associated set of configuration parameters defined by the properties of the [bibus » queryService](#) class.

References

Runs instances of the following classes:

- [bibus » queryServiceTask](#)

Uses the following method sets:

- [asynch](#)
- [dataSource](#)
- [rolapCubeAdministration](#)

What's new

New in Version 10.1.0 – “Query Modes” on page 1865

This service was added.

New in Version 10.1.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1846

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/queryService/201101/>.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This service now supports the [rolapCubeAdministration](#) method set.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/queryService/201109/>.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/queryService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/queryService/201404/>.

relationalMetadataService

Defines the service responsible for extracting relational metadata from data sources.

The relational metadata service has an associated set of configuration parameters defined by the properties of the [bibus » relationalMetadataService](#) class.

References

Runs instances of the following classes:

- [bibus » relationalMetadataServiceSpecification](#)

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Relational metadata service” on page 1835

This service was added.

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201109/`.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201301/`.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/`.

reportService

Defines the service responsible for the execution of reports ([bibus](#) » [baseReport](#) class).

The report service is dedicated to running interactive reports. For example, when a user runs a report from IBM Cognos Connection, the request is sent to the report service.

The report service has an associated set of configuration parameters defined by the properties of the [bibus](#) » [reportService](#) class. These parameters can be configured to optimize the report service for interactive use. This is the only difference between the report service and the batch report service ([batchReportService](#)).

This service may also make additional requests to the following IBM Cognos Analytics services when processing a request:

- [contentManagerService](#) service
- [deliveryService](#) service
- [eventManagementService](#) service
- [monitorService](#) service

Example: Connecting to the Report Service in Java

```
private ReportService_ServiceLocator reportServiceLocator = null;
private ReportService_PortType reportService = null;
public static String RS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
reportServiceLocator = new ReportService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(RS_URL);
    reportService = reportServiceLocator.getreportService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the Report Service in C# .NET

```
try
{
    reportService1 c8RS = new reportService1();
    c8RS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
```

```
    ...  
  }  
  // catch exceptions
```

References

Runs instances of the following classes:

- [bibus » analysis](#)
- [bibus » drillPath](#)
- [bibus » interactiveReport](#)
- [bibus » metadataServiceLineageSpecification](#)
- [bibus » query](#)
- [bibus » report](#)
- [bibus » reportServiceAnalysisSpecification](#)
- [bibus » reportServiceDrillThroughSpecification](#)
- [bibus » reportServiceInteractiveReportSpecification](#)
- [bibus » reportServiceMetadataSpecification](#)
- [bibus » reportServiceQuerySpecification](#)
- [bibus » reportServiceReportSpecification](#)
- [bibus » reportTemplate](#)
- [bibus » reportView](#)

Uses the following method sets:

- [asynch](#)
- [dataSource](#)
- [drillThrough](#)
- [paging](#)
- [parameter](#)
- [promptPaging](#)
- [report](#)
- [validate](#)

What's new

New in Version 8.3 – “[Parameter Method Set](#)” on page 1915

This service now implements the [parameter](#) method set.

New in Version 8.4 – “[PowerPlay 8 Integration](#)” on page 1886

This service now implements the [dataSource](#) method set.

New in Version 10.2.0 – “[Changes to IBM Cognos Analytics SOAP action for services](#)” on page 1840

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/reportService/201109/>.

New in Version 10.2.1 – “[Changes to IBM Cognos Analytics SOAP action for services](#)” on page 1831

The SOAPAction HTTP header field for this service has changed to <http://www.ibm.com/xmlns/prod/cognos/reportService/201301/>.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/reportService/201404/`.

repositoryService

This service is "Reserved" and is not accessible with the SOAP toolkits. However, a REST API is available that allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

References

Uses the following method sets:

- [asynch](#)

What's new

New in Version 10.2.0 – “Repository service” on page 1840

This service was added.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/repositoryService/201301/`.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/`.

saCAMService

Standalone Cognos Access Manager

Defines the service responsible for providing authentication and authorization functionality to internal IBM Cognos components.

The standalone CAM service has an associated set of configuration parameters defined by the properties of the [saCAMService](#) class.

Note: Do not use the IBM Cognos Software Development Kit to start, stop, or update this service.

Example: Connecting to the standalone CAM Service in Java

```
private SaCAMService_ServiceLocator saCAMServiceLocator = null;
private SaCAMService_PortType saCAMService = null;
public static String SaCAM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
saCAMServiceLocator = new SaCAMService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(SaCAM_URL);
    saCAMService = saCAMServiceLocator.getsaCAMService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the standalone CAM Service in C# .NET

```
try
{
saCAMService1 c8saCAM = new saCAMService1();
c8saCAM.Url = "http://localhost:9300/p2pd/servlet/dispatch";
...
}
// catch exceptions
```

References

Uses the following method sets:

- [asynch](#)
- [standaloneCAM](#)

What's new

New in Version 10.2.0 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1838

This service was added.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/sacamsrv-auth/201301/`.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/sacamsrv-auth/201404/`.

systemService

Defines the service that can be used to obtain application-wide, IBM Cognos Analytics configuration parameters. This service also provides methods that can be used to normalize and validate locale strings, as well as to map locale strings to locales supported by your application.

The system service has an associated set of configuration parameters defined by the properties of the [bibus » systemService](#) class.

Example: Connecting to the system service in Java.

```
private SystemService_ServiceLocator systemServiceLocator = null;
private SystemService_PortType systemService = null;
public static String SS_URL = "http://localhost:9300/p2pd/servlet/dispatch";
systemServiceLocator = new SystemService_ServiceLocator();
...
try
{
    java.net.URL serverURL = new java.net.URL(SS_URL);
    systemService = systemServiceLocator.getsystemService(serverURL);
    ...
}
// catch exceptions
```

Example: Connecting to the system service in C# .NET.

```
try
{
    systemService1 c8SS = new systemService1();
    c8SS.Url = "http://localhost:9300/p2pd/servlet/dispatch";
    ...
}
// catch exceptions
```

References

Uses the following method sets:

- [system](#)

What's new

New in Version 10.2.0 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1840

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/systemService/201109/`.

New in Version 10.2.1 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1831

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/systemService/201301/`.

New in Version 10.2.2 – “Changes to IBM Cognos Analytics SOAP action for services” on page 1823

The SOAPAction HTTP header field for this service has changed to `http://www.ibm.com/xmlns/prod/cognos/systemService/201404/`.

Chapter 13. Method sets

This chapter describes the method sets used to define IBM Cognos Web services. Method sets are groups of related methods. For example, all the report-specific methods are grouped together in a method set. Each service provided by IBM Cognos includes one or more method sets. Some method sets are used by more than one service.

Method sets are not directly exposed as classes or interfaces that are accessible from any toolkit.

For more information about the services and which methods they use, see [Chapter 12, “Services,”](#) on page 129.

agent

Defines the set of methods related to agents.

Methods

- [agent](#) » [deleteHotList\(objectPath\)](#)

References

Implemented by the following services:

- [agentService](#)

asynch

Defines the set of common methods that implement asynchronous processing. All services that run tasks use this method set.

Most of the methods in this method set return responses that support an asynchronous conversation. However, the [asynch](#) » [cancel\(conversation\)](#) method cancels an asynchronous conversation and the [asynch](#) » [release\(conversation\)](#) method removes an inactive request from the service cache.

Use the methods in this method set in combination with other methods to take advantage of features that are specific to a particular service. Any methods that take a parameter of type [bibus](#) » [asynchRequest](#) can be used during an asynchronous conversation. For example, use the methods [asynch](#) » [run\(objectPath, parameterValues, options\)](#) and [paging](#) » [nextPage\(conversation, parameterValues, options\)](#) with [reportService](#) to run a report, and then retrieve the next page of the report output.

Methods

- [asynch](#) » [cancel\(conversation\)](#)
- [asynch](#) » [release\(conversation\)](#)
- [asynch](#) » [run\(objectPath, parameterValues, options\)](#)
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#)
- [asynch](#) » [wait\(conversation, parameterValues, options\)](#)

References

Implemented by the following services:

- [agentService](#)
- [batchReportService](#)

- [contentManagerService](#)
- [dataIntegrationService](#)
- [dataMovementService](#)
- [deliveryService](#)
- [dimensionManagementService](#)
- [eventManagementService](#)
- [idVizService](#)
- [indexSearchService](#)
- [indexUpdateService](#)
- [jobService](#)
- [metadataService](#)
- [migrationService](#)
- [mobileService](#)
- [monitorService](#)
- [planningAdministrationConsoleService](#)
- [planningRuntimeService](#)
- [planningTaskService](#)
- [powerPlayService](#)
- [queryService](#)
- [relationalMetadataService](#)
- [reportService](#)
- [repositoryService](#)
- [saCAMService](#)

authentication

Defines the set of authentication methods.

Methods

- [authentication](#) » [logoff\(\)](#)
- [authentication](#) » [logon\(credentials, roles\)](#)
- [authentication](#) » [selectRoles\(roles\)](#)

References

Implemented by the following services:

- [contentManagerService](#)

content

Defines the set of content management methods. Use these methods to manipulate objects in the content store, for example to move an item to a new location or to delete an item that is no longer needed.

Methods

- [content](#) » [activate\(searchPath\)](#)
- [content](#) » [activateURI\(uri\)](#)

- [content » add\(parentPath, objects, options\)](#)
- [content » addAnnotations\(containerPath, objects, options\)](#)
- [content » copy\(objects, targetPath, options\)](#)
- [content » copyAccount\(sourceAccountPath, targetAccountPath, options\)](#)
- [content » copyRename\(objects, targetPath, newNames, options\)](#)
- [content » delete\(objects, options\)](#)
- [content » deleteAccount\(objectPath, options\)](#)
- [content » deleteTenants\(tenantIDs\)](#)
- [content » determineRouting\(objectPaths\)](#)
- [content » getActiveContentManager\(\)](#)
- [content » listTenants\(options\)](#)
- [content » move\(objects, targetPath, options\)](#)
- [content » moveRename\(objects, targetPath, newNames, options\)](#)
- [content » query\(searchPath, properties, sortBy, options\)](#)
- [content » queryMultiple\(requests\)](#)
- [content » queryMultipleCache\(requests, options\)](#)
- [queryTenantMembership\(tenantIDs\)](#)
- [content » update\(objects, options\)](#)

References

Implemented by the following services:

- [contentManagerService](#)

What's new

New in Version 8.4 — [“Report Output Annotations” on page 1895](#)

Added the [content » addAnnotations\(containerPath, objects, options\)](#) method.

dataSource

Defines the set of methods related to data sources.

Methods

- [dataSource » testDataSourceConnection\(connectionString, credentials\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [metadataService](#)
- [queryService](#)
- [reportService](#)

What's new

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This method set was added.

delivery

Defines the set of methods to use with email alert lists for agents and reports.

Methods

- [delivery](#) » [addNotification\(objectPath\)](#)
- [delivery](#) » [clearNotifications\(objectPath\)](#)
- [delivery](#) » [deleteAllNotifications\(\)](#)
- [delivery](#) » [deleteNotification\(objectPath\)](#)
- [delivery](#) » [queryNotification\(objectPath\)](#)

References

Implemented by the following services:

- [deliveryService](#)

deployment

Defines the set of deployment methods.

Methods

- [deployment](#) » [getDeploymentOptions\(archive, options\)](#)
- [deployment](#) » [listArchives\(\)](#)

References

Implemented by the following services:

- [contentManagerService](#)

dispatcher

Defines the set of dispatcher methods.

Methods

- [dispatcher](#) » [ping\(dispatcherPath\)](#)
- [dispatcher](#) » [startService\(servicePath\)](#)
- [dispatcher](#) » [stopService\(servicePath, immediately\)](#)

References

Implemented by the following services:

- [dispatcher](#)

drillThrough

Defines the set of drill-through methods. Use these methods to gather the information required to perform a drill-through operation.

Methods

- [drillThrough](#) » [addDrillPath\(parentPath, object, options\)](#)

- [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#)
- [drillThrough](#) » [findDrillThroughPaths\(objectPath, parameterValues, options\)](#)
- [drillThrough](#) » [queryDrillPath\(objectPath, parameterValues, options\)](#)
- [drillThrough](#) » [updateDrillPath\(object, options\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [reportService](#)

What's new

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

The [drillThrough](#) » [findDrillThroughPaths\(objectPath, parameterValues, options\)](#) method was added.

New in Version 8.4 – “Supporting New Drill-through Targets” on page 1890

The [drillThrough](#) » [addDrillPath\(parentPath, object, options\)](#), [drillThrough](#) » [queryDrillPath\(objectPath, parameterValues, options\)](#), and [drillThrough](#) » [updateDrillPath\(object, options\)](#) methods were added.

New in Version 8.4 – “Drill-Through Improvements” on page 1889

The [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#) method was added.

New in Version 10.1.0 – “Old Drill-through API Removed” on page 1871

The methods `determineDrillThroughTargetParameterValues(objectPaths, sourceContext, sourceContextValues, targetParameterAssignments, targetParameters, parameterValues, options)`, `getDrillThroughDefaultParameterAssignments(objectPath, parameterValues, options)`, `getDrillThroughPaths(objectPath, parameterValues, options)`, and `getDrillThroughTargetParameterValues(objectPath, sourceContextValues, parameterValues, options)` were removed.

event

Defines the set of event management methods.

Methods

- [event](#) » [cancelEvent\(eventID\)](#)
- [event](#) » [cancelEvents\(eventIDs\)](#)
- [event](#) » [delayEventsFor\(eventIDs, for\)](#)
- [event](#) » [delayEventsUntil\(eventIDs, until\)](#)
- [event](#) » [holdEvent\(eventID\)](#)
- [event](#) » [holdEvents\(eventIDs\)](#)
- [event](#) » [releaseEvent\(eventID\)](#)
- [event](#) » [releaseEvents\(eventIDs\)](#)
- [event](#) » [runAt\(startTime, objectPath, parameterValues, options\)](#)
- [event](#) » [scheduleEvent\(eventID\)](#)
- [event](#) » [scheduleEvents\(eventIDs\)](#)
- [event](#) » [trigger\(triggerName\)](#)

- [event](#) » [updateEvents\(events\)](#)

References

Implemented by the following services:

- [eventManagementService](#)

What's new

New in Version 8.3 — “[Schedule Priority](#)” on page 1912

The [event](#) » [updateEvents\(events\)](#) method was added.

indexTerm

Reserved.

Methods

- [indexTerm](#) » [addTermAssociation\(term, parameterValues, options\)](#)
- [indexTerm](#) » [deleteTermAssociation\(term, parameterValues, options\)](#)

References

Implemented by the following services:

- [indexUpdateService](#)

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

indexUpdate

Reserved.

Methods

- [indexUpdate](#) » [add\(objectPath, parameterValues, options\)](#)
- [indexUpdate](#) » [delete\(objectPath, parameterValues, options\)](#)
- [indexUpdate](#) » [get\(objectPath, parameterValues, options\)](#)

References

Implemented by the following services:

- [indexUpdateService](#)

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

metadata

Defines the set of metadata service methods.

Methods

- [metadata](#) » [queryMetadata\(request\)](#)
- [metadata](#) » [updateMetadata\(request\)](#)

References

Implemented by the following services:

- [metadataService](#)

monitor

Defines the set of monitor methods.

Methods

- [monitor](#) » [background\(conversation\)](#)

References

Implemented by the following services:

- [monitorService](#)

paging

Defines the set of methods related to navigating through result sets.

The methods in this interface return responses that support an asynchronous conversation. Use these methods in conjunction with the methods in the [asynch](#) method set to extend the set of actions that can be performed during an asynchronous conversation.

Methods

- [paging](#) » [currentPage\(conversation, parameterValues, options\)](#)
- [paging](#) » [firstPage\(conversation, parameterValues, options\)](#)
- [paging](#) » [lastPage\(conversation, parameterValues, options\)](#)
- [paging](#) » [nextPage\(conversation, parameterValues, options\)](#)
- [paging](#) » [previousPage\(conversation, parameterValues, options\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [eventManagementService](#)
- [indexSearchService](#)
- [monitorService](#)
- [reportService](#)

parameter

Defines the set of methods related to parameters.

Methods

- [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#)
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#)
- [parameter](#) » [getParameters\(objectPath, parameterValues, options\)](#)
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [powerPlayService](#)
- [reportService](#)

What's new

New in Version 8.3 – “Parameter Method Set” on page 1915

This method set was added.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

Added the [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method and the [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method.

promptPaging

Defines the set of methods related to paging through prompt pages.

The methods in this interface return responses that support an asynchronous conversation. Use these methods in conjunction with the methods in the [async](#) method set to extend the set of actions that can be performed during an asynchronous conversation.

Methods

- [promptPaging](#) » [back\(conversation, parameterValues, options\)](#)
- [promptPaging](#) » [forward\(conversation, parameterValues, options\)](#)
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [indexSearchService](#)
- [powerPlayService](#)
- [reportService](#)

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

Added the [promptPaging » getPromptValues\(conversation, parameterValues, options\)](#) method.

report

Defines the set of methods related to running and manipulating reports.

Many of the methods in this method set return responses that support an asynchronous conversation. In addition, some of these methods, such as [validate » validate\(objectPath, parameterValues, options\)](#) and [parameter » getParameters\(objectPath, parameterValues, options\)](#), start a new asynchronous conversation. Use these methods in conjunction with the methods in the [asynch](#) method set to extend the set of actions that can be performed during an asynchronous conversation.

Some of these methods offer functionality that is more extensive than similar methods provided by the [contentManagerService](#). When adding, querying, or updating objects of type [bibus » authoredReport](#) in the content store, use the methods in this method set rather than the corresponding [contentManagerService](#) methods.

For example, you can retrieve an [bibus » authoredReport](#) object from the content store using either the [content » query\(searchPath, properties, sortBy, options\)](#) method or the [report » query\(objectPath, parameterValues, options\)](#) method. However, only the [report » query\(objectPath, parameterValues, options\)](#) method will update the report specification after retrieving the [bibus » authoredReport](#) object from the content store.

Methods

- [report » add\(parentPath, object, options\)](#)
- [report » collectParameterValues\(objectPath, parameterValues, options\)](#) – obsolete
- [report » collectParameterValuesSpecification\(specification, parameterValues, options\)](#) – obsolete
- [report » deliver\(conversation, parameterValues, options\)](#)
- [report » drill\(conversation, parameterValues, options\)](#)
- [report » getContext\(conversation, parameterValues, options\)](#)
- [report » getObjectContext\(objectPath, parameterValues, options\)](#)
- [report » getOutput\(conversation, parameterValues, options\)](#)
- [report » getPromptValues\(conversation, parameterValues, options\)](#) – obsolete
- [report » lineage\(conversation, parameterValues, options\)](#)
- [report » query\(objectPath, parameterValues, options\)](#)
- [report » render\(conversation, parameterValues, options\)](#)
- [report » testDataSourceConnection\(connectionString, credentials\)](#) – obsolete
- [report » update\(object, options\)](#)
- [report » validate\(objectPath, parameterValues, options\)](#) – obsolete
- [report » validateSpecification\(specification, parameterValues, options\)](#) – obsolete

References

Implemented by the following services:

- [batchReportService](#)
- [reportService](#)

What's new

New in Version 8.4 – “[Report Server deliver\(\) method](#)” on page 1883

Added the `report » deliver(conversation, parameterValues, options)` method.

New in Version 10.1.0 – “[Support for Lineage Requests](#)” on page 1855

This method set was added.

rolapCubeAdministration

Defines the set of methods that implement ROLAP cube administration.

Methods

- [rolapCubeAdministration](#) » [clearCubeWorkloadLog\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [getCubeMessages\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [getCubeMetrics\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [getCubeState\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [incrementallyLoadCubes\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [pauseCubes\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [refreshCubeDataCache\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [refreshCubeMemberCache\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [refreshCubeSecurity\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [restartCubes\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [startCubes\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [stopCubes\(cubeNames, parameterValues, options\)](#)
- [rolapCubeAdministration](#) » [testDataSourceConnectionWithInfo\(connectionString, credentials\)](#)

References

Implemented by the following services:

- [queryService](#)

What's new

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This method set was added.

standaloneCAM

Defines the set of methods related to the IBM Cognos Access Manager (CAM) service.

Methods

- [standaloneCAM](#) » [getIdentity\(\)](#)
- [standaloneCAM](#) » [retrieveCredential\(namespace\)](#)
- [standaloneCAM](#) » [terminateSessions\(search\)](#)

References

Implemented by the following services:

- [saCAMService](#)

system

Defines the set of system methods. Use these methods to verify environment settings.

Methods

- [system](#) » [getConfiguration\(properties\)](#)
- [system](#) » [getFormatSamples\(name\)](#)
- [system](#) » [mapContentLocale\(locale, normalize\)](#)
- [system](#) » [mapProductLocale\(locale, normalize\)](#)
- [system](#) » [normalizeLocale\(locale\)](#)
- [system](#) » [validateContentLocale\(locale\)](#)
- [system](#) » [validateProductLocale\(locale\)](#)

References

Implemented by the following services:

- [systemService](#)

validate

Defines the set of methods related to validating specifications.

Methods

- [validate](#) » [validate\(objectPath, parameterValues, options\)](#)
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#)

References

Implemented by the following services:

- [batchReportService](#)
- [dataMovementService](#)
- [powerPlayService](#)
- [reportService](#)

Chapter 14. Methods

Use the BI Bus API methods to manage the objects in the content store. These methods are defined as operations in the IBM Cognos Web Service Definition Language (WSDL) file.

For information about permissions and capability requirements for methods, see [“Security Prerequisites for Requests”](#) on page 53 and the individual method descriptions provided in this chapter.

Methods in Java

In Java, the methods provided by each BI Bus API service are defined by an interface named `<service>_Port`, which in turn is implemented by a class named `<service>Stub`. You access BI Bus API methods using an instance of this class. For example, methods that belong to the `reportService` service are available through instances of the `reportServiceStub` class. For more information about the Java classes that correspond to the BI Bus API services, see [“IBM Cognos Analytics Services”](#) on page 5.

activate(searchPath)

Use this method to change the active Content Manager instance.

The active Content Manager instance will respond to normal requests and is in the running state. A Content Manager service that is on standby will not respond to normal requests and is in the standby state. A Content Manager service in standby can become active as a result of failure recovery or by a call from this method.

If this method is called and the state of the currently active Content Manager service is suspended, the Content Manager specified by searchPath will start up with a running state of suspended.

This method is executed by the currently active Content Manager service. After the newly specified Content Manager service is active, the service that executed the method call changes to standby.

Calling this method terminates any current sessions and may require users to reauthenticate, depending on the authentication provider used.

Signatures

Java and Apache Axis

```
public void activate(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    searchPath)
```

C# .NET

```
public void activate(SearchPathSingleObject searchPath)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `activate` method, along with the associated SOAP actions, are listed in the following table.

Table 11. Services implementing the activate method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/ contentManagerService/201404/

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseAdministrationPortal](#)

Input parameters

Use the following parameters when calling this method.

searchPath

Specifies the search path of the [bibus](#) » [contentManagerService](#) object you want to put in the [running](#) state.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

None.

activateURI(uri)

Use this method to change the active Content Manager instance.

The active Content Manager instance will respond to normal requests and is in the [running](#) state. A Content Manager service that is on standby will not respond to normal requests and is in the [standby](#) state. A Content Manager service in [standby](#) can become active as a result of failure recovery or by a call from this method.

If this method is called and the state of the currently active Content Manager service is [suspended](#), the Content Manager specified by [uri](#) will start up with a running state of [suspended](#).

This method is executed by the currently active Content Manager service. After the newly specified Content Manager service is active, the service that executed the method call changes to standby.

Calling this method results in the termination of any current sessions and may require users to reauthenticate, depending on the authentication provider used.

Signatures

Java and Apache Axis

```
public void activateURI(java.lang.String uri)
```

C# .NET

```
public void activateURI(string uri)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the activateURI method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseAdministrationPortal](#)

Input parameters

Use the following parameters when calling this method.

uri

Specifies the URI of the Content Manager that you want to put in the [running](#) state.

This argument

- is of type anyURI

is encoded as type `xs:string`

Return values

None.

[add\(parentPath, objects, options\)](#)

Use this method to add objects to the content store.

If you want to add a [bibus](#) » [baseReport](#) object to the content store, use [report](#) » [add\(parentPath, object, options\)](#), which sets the values for some properties to reflect the corresponding values in the report specification.

This method returns properties for each added object.

The [bibus](#) » [uiClass](#) » [viewed](#) property on new objects is set to false for security contexts other than the current security context.

If optimistic concurrency control is used, the [bibus](#) » [baseClass](#) » [version](#) property is compared to the [version](#) of the affected objects in the content store. For more information, see [“Concurrency Control”](#) on [page 55](#).

Use of this method requires:

- write permission for the parent object
- traverse permission for all ancestors of the object
- read permission for the added object to retrieve property values

- setPolicy permission to read or write to the [bibus](#) » [baseClass](#) » [policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] add(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    parentPath, com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.AddOptions options)
```

C# .NET

```
public baseClass[] add(searchPathSingleObject parentPath, baseClass[]
objects, addOptions options)
```

Example: Using the add(parentPath, objects, options) Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [content](#) » [add\(parentPath, objects, options\)](#) method with the Content Manager service.

To see this code in context, view the Java sample HandlersCS/CSHandlers.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
AddOptions ao = new AddOptions();
ao.setUpdateAction(UpdateActionEnum.replace);

return connection.getCMSService().add(
    new SearchPathSingleObject(path), new BaseClass[] { bc }, ao)[0];
```

Example: Using the add(parentPath, objects, options) Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the [content](#) » [add\(parentPath, objects, options\)](#) method with the Content Manager service.

To see this code in context, view the C# sample PrintReport/PrintReport.cs. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
bcNewPrinter = connection.CBICMS.add(printerParentPath, bcAddPrinter,
add_options);
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the add method, along with the associated SOAP actions, are listed in the following table.

Table 13. Services implementing the add method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

parentPath

Specifies a search path that identifies a single object in the content store. This single object path identifies the parent location for the object or group of objects to add.

This parameter can be overridden on a per-object basis. This can be done by specifying the [searchPath](#) for the [objects](#) you are adding. If specifying the target location in this manner, you can set the value of this parameter to NULL.

For more information about specifying a search path, see [Chapter 32, “Search path syntax,” on page 1565](#).

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

objects

Specifies the objects to add to the content store.

If you do not specify a value for the [bibus » baseClass » policies](#) property for a new object, the policies for the object are acquired from the parent of the object.

When you create an object, you do not have to specify a name. If you do not specify a name, Content Manager generates a name using a string that represents the current date and time, in a format that is a subset of the ISO 8601 format. For more information, see the description of the [bibus » baseClass » name](#) property.

If you specify the [bibus » baseClass » searchPath](#) property for an object, it is treated as the search path for the container of the new object. This overrides the value specified in [parentPath](#).

If you specify a value for a read-only property of a new object, such as the [bibus » baseClass » version](#) property, it is ignored.

By default, if the object being added contains a reference to a non-existent object, a fault is generated and the object is not added. For information about changing the default behavior, see the [bibus » addOptions » ignoreInvalidObjectReference](#) property.

This argument

- is an array of type [bibus » baseClass](#)
is encoded as type `tns:baseClassArray`

options

Specifies preferences for the resolution of conflicts and the encoding of data.

This argument

- is of type [bibus » addOptions](#)
is encoded as type `tns:addOptions`

Return values

This method returns the following values.

result

Returns information about the objects that were added. The information that is returned is determined by the value of the [bibus](#) » [addOptions](#) » [returnProperties](#) property.

This result

- is an array of type [bibus](#) » [baseClass](#)
is encoded as type `tns:baseClassArray`

add(objectPath, parameterValues, options)

Reserved.

References

Part of the following method sets:

- [indexUpdate](#)

Implemented by the following services:

The services that implement the add method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
indexUpdateService	<code>http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/</code>

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

add(parentPath, object, options)

Use this method to add an [bibus » authoredReport](#) object to the content store.

This method performs a number of activities before it adds the specified object to the content store by calling the [content » add\(parentPath, objects, options\)](#) method:

- The report specification is validated. If a query is being added, it is converted to a report specification and then validated. If the specification is not valid, this method will fail and an appropriate fault will be returned.
- For [bibus » analysis](#) objects, the following properties are set to reflect the corresponding values in the report specification:
 - [bibus » authoredReport » metadataModel](#) property
 - [bibus » authoredReport » metadataModelPackage](#) property
- For [bibus » interactiveReport](#) objects, the following properties are set to reflect the corresponding values in the report specification:
 - [bibus » authoredReport » canBurst](#) property
 - [bibus » authoredReport » metadataModel](#) property
 - [bibus » authoredReport » metadataModelPackage](#) property
 - [bibus » authoredReport » paths](#) property
- For [bibus » query](#) objects, the following properties are set to reflect the corresponding values in the report specification:
 - [bibus » baseReport » executionPageDefinition](#) property
 - [bibus » authoredReport » metadataModel](#) property
 - [bibus » authoredReport » metadataModelPackage](#) property

- For [bibus » report](#) objects, the following properties are set to reflect the corresponding values in the report specification:
 - [bibus » authoredReport » canBurst](#) property
 - [bibus » authoredReport » metadataModel](#) property
 - [bibus » authoredReport » metadataModelPackage](#) property
 - [bibus » authoredReport » paths](#) property
- For [bibus » reportTemplate](#) objects, the following properties are set to reflect the corresponding values in the report specification:
 - [bibus » authoredReport » canBurst](#) property
 - [bibus » authoredReport » metadataModel](#) property
 - [bibus » authoredReport » metadataModelPackage](#) property
 - [bibus » authoredReport » paths](#) property

To add multiple objects to the content store, you can use [content » add\(parentPath, objects, options\)](#). However, you must set all the required properties for each [bibus » authoredReport](#) you add this way.

Any data stored in the [bibus » reportCache](#) contained by the [bibus » baseReport](#) object is cleared.

Use of this method requires:

- the [canUseReportStudio](#), [canUseQueryStudio](#), [canUseEventStudio](#), [canUseAnalysisStudio](#), or the [canUseCognosInsight](#) capability
- write permission for the parent object
- traverse permission for all ancestors of the object
- read permission for the added object to retrieve property values
- [setPolicy](#) permission to read or write to the [bibus » baseClass » policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AuthoredReport add(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    parentPath, com.cognos.developer.schemas.bibus._3.AuthoredReport
    object, com.cognos.developer.schemas.bibus._3.AddOptions options)
```

C# .NET

```
public authoredReport add(searchPathSingleObject parentPath,
    authoredReport @object, addOptions options)
```

Example: Using the [add\(parentPath, object, options\)](#) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [report » add\(parentPath, object, options\)](#) method with the report service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page [1419](#).

```
AddOptions ao = new AddOptions();
ao.setUpdateAction(UpdateActionEnum.replace);

return connection.getReportService().add(
    new SearchPathSingleObject(path), rpt, ao);
```

Example: Using the add(parentPath, object, options) Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the `report » add(parentPath, object, options)` method with the report service.

To see this code in context, view the C# sample `AddReport/AddReport.cs`. For more information about the samples, see Chapter 18, “Code samples and language-specific coding practices,” on page 1419.

```
cBIRS.add(searchPathSO, newReport, addReportOptions);
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the add method, along with the associated SOAP actions, are listed in the following table.

Table 15. Services implementing the add method.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/</code>

Capability Rules

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 16. Capability rules for the add method.

Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseAnalysisStudio	bibus » analysis
Trusted	bibus » userCapabilityEnum » canUseEventStudio	bibus » report or bibus » reportTemplate
Trusted	bibus » userCapabilityEnum » canUseQueryStudio and bibus » userCapabilityEnum » canUseQueryStudioFileManagement	bibus » query
—	bibus » userCapabilityEnum » canUseCognosInsight	bibus » report , bibus » reportTemplate , or bibus » interactiveReport
—	bibus » userCapabilityEnum » canUseReportStudio and bibus » userCapabilityEnum » canUseReportStudioFileManagement	bibus » report , bibus » reportTemplate , or bibus » interactiveReport

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

What's new

New in Version 10.1.0 – “[IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace](#)” on page 1858

The capability rules were updated to include the rules related to the use of the [bibus » report](#) class with the [userCapabilityEnum » canUseDashboardViewer](#) value user capability.

New in Version 10.1.0 – “[IBM Cognos Active Report](#)” on page 1869

This method can now be used with a [bibus » interactiveReport](#).

The capability rules were updated to include the rules related to the use of the [bibus » interactiveReport](#) class.

Input parameters

Use the following parameters when calling this method.

parentPath

Specifies the parent of the new object.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

object

Specifies the report object to be added.

If you specify the [bibus » baseClass » searchPath](#) property for an object, it is treated as the search path for the container of the new object. This will override the value specified in [parentPath](#).

This argument

- is of type [bibus » authoredReport](#)
is encoded as type `tns:authoredReport`

options

Specifies options for the resolution of conflicts and the encoding of data.

This argument

- is of type [bibus » addOptions](#)
is encoded as type `tns:addOptions`

Return values

This method returns the following values.

result

Returns information about the object that was added. The information that is returned is determined by the value of the `bibus » addOptions » returnProperties` property.

This result

- is of type `bibus » authoredReport`
is encoded as type `tns:authoredReport`

addAnnotations(containerPath, objects, options)

Use this method to add `bibus » annotation` objects to the content store.

This method returns properties for each added object.

The `bibus » uiClass » viewed` property on new objects is set to false for security contexts other than the current security context.

If optimistic concurrency control is used, the `bibus » baseClass » version` property is compared to the version of the affected objects in the content store. For more information, see [“Concurrency Control” on page 55](#).

This method will create an `bibus » annotationFolder` object in the object identified by the `containerPath` parameter if one does not exist. If the object is created, it has the following property values:

Property	Value
<code>owner</code>	<code>owner</code> of the object identified by <code>containerPath</code> parameter
<code>policies</code>	grant “Everyone” on page 1698 read, traverse and write

Annotations created using this method will have the following property values:

Property	Value
<code>policies</code>	grant “Everyone” on page 1698 read, and traverse grant the <code>owner</code> of the object identified by <code>containerPath</code> parameter <code>setPolicy</code>

If a policy is specified in an `bibus » annotation` object in the `objects` parameter it is ignored.

Use of this method requires:

- traverse permission for the parent object
- the value of the `allowAnnotations` property of the object identified by the `containerPath` parameter must be true
- traverse permission for all ancestors of the object
- read permission for the added object to retrieve property values
- `setPolicy` permission to read or write to the `bibus » baseClass » policies` property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[]
```

```
addAnnotations(
  com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
  containerPath, com.cognos.developer.schemas.bibus._3.BaseClass[]
  objects, com.cognos.developer.schemas.bibus._3.AddOptions options)
```

C# .NET

```
public baseClass[] addAnnotations(searchPathSingleObject
  containerPath, baseClass[] objects, addOptions options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the addAnnotations method, along with the associated SOAP actions, are listed in the following table.

<i>Table 19. Services implementing the addAnnotations method.</i>	
Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

What's new

New in Version 8.4 — “[Report Output Annotations](#)” on page 1895

This method was added.

Input parameters

Use the following parameters when calling this method.

containerPath

Specifies a search path that identifies a single [bibus » agentOutputHotList](#) or [bibus » reportVersion](#) in the content store. This single object path identifies the grandparent location for the [bibus » annotation](#) object or group of objects to add.

This parameter can be overridden on a per-object basis. This can be done by specifying the [searchPath](#) for the [objects](#) you are adding. If specifying the target location in this manner, you can set the value of this parameter to NULL.

For more information about specifying a search path, see [Chapter 32, “Search path syntax,”](#) on page 1565.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

objects

Specifies the [bibus » annotation](#) objects to add to the content store.

When you create an object, you do not have to specify a name. If you do not specify a name, Content Manager generates a name using a string that represents the current date and time, in a format that is a subset of the ISO 8601 format. For more information, see the description of the [bibus » baseClass » name](#) property.

If you specify the `bibus » baseClass » searchPath` property for an object, it is treated as the search path for the grandparent of the new object. This overrides the value specified in `containerPath`.

If you specify a value for a read-only property of a new object, such as the `bibus » baseClass » version` property, the value is ignored.

By default, if the object being added contains a reference to a non-existent object, a fault is generated and the object is not added. For information about changing the default behavior, see the `bibus » addOptions » ignoreInvalidObjectReference` property.

This argument

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

options

Specifies preferences for the resolution of conflicts and the encoding of data.

This argument

- is of type `bibus » addOptions`
is encoded as type `tns:addOptions`

Return values

This method returns the following values.

result

Returns information about the objects that were added. The information that is returned is determined by the value of the `bibus » addOptions » returnProperties` property.

This result

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

addDrillPath(parentPath, object, options)

Use this method to add a `bibus » drillPath` object to the content store.

This method performs a number of activities before it adds the specified object to the content store by calling the `content » add(parentPath, objects, options)` method:

- The drill-through specification is validated. If the specification is not valid, this method fails and returns an appropriate error.
- The following properties are set to reflect the corresponding values in the drill-through specification:
 - `bibus » drillPath » deploymentReferences` property

To add multiple objects to the content store, you can use `content » add(parentPath, objects, options)`. However, you must set all the required properties for each `bibus » drillPath` object you add this way.

Use of this method requires:

- write permission for the parent object if this method is being used to add a new object
- traverse permission for all ancestors of the object
- read permission for the added object to retrieve property values
- `setPolicy` permission to read or write to the `bibus » baseClass » policies` property if this method is being used to overwrite an existing object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.DrillPath addDrillPath(  
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
parentPath, com.cognos.developer.schemas.bibus._3.DrillPath object,  
com.cognos.developer.schemas.bibus._3.AddOptions options)
```

C# .NET

```
public drillPath addDrillPath(searchPathSingleObject parentPath,  
drillPath @object, addOptions options)
```

References

Part of the following method sets:

- [drillThrough](#)

Implemented by the following services:

The services that implement the addDrillPath method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/

What's new

New in Version 8.4 — [“Supporting New Drill-through Targets” on page 1890](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

parentPath

Specifies the parent of the new object.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

object

Specifies the [bibus](#) » [drillPath](#) object to be added.

If you specify the [bibus](#) » [baseClass](#) » [searchPath](#) property for an object, it is treated as the search path for the container of the new object. This will override the value specified in [parentPath](#).

This argument

- is of type [bibus » drillPath](#)
is encoded as type `tns:drillPath`

options

Specifies options for conflict resolution and data encoding.

This argument

- is of type [bibus » addOptions](#)
is encoded as type `tns:addOptions`

Return values

This method returns the following values.

result

Returns information about the object that was added. The information that is returned is determined by the value of the [bibus » addOptions » returnProperties](#) property.

This result

- is of type [bibus » drillPath](#)
is encoded as type `tns:drillPath`

addNotification(objectPath)

Use this method to add the current user to an object's alert list. If the user does not have either an email address or a notification email address defined, or if the user is already on the alert list, this method returns a fault.

If the user does not have a notification email address ([notificationEMail](#)) defined, but does have an email address ([email](#)) defined, the [notificationEMail](#) property of the user's account is updated to match the value in the [email](#) property.

This method returns no [bibus » asynchDetail](#) objects in the [bibus » asynchReply » details](#) property.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- read permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply  
addNotification(  
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
objectPath)
```

C# .NET

```
public asyncReply addNotification(searchPathSingleObject objectPath)
```

Example: Using the addNotification(objectPath) Method with the Delivery Service in Java

The following Java code snippet demonstrates how to use the [delivery » addNotification\(objectPath\)](#) method with the delivery service.

To see this code in context, view the Java sample Alerts/ManageAlerts.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asyncReply = connection.getDeliveryService().addNotification(  
    new SearchPathSingleObject(reportPath));
```

References

Part of the following method sets:

- [delivery](#)

Implemented by the following services:

The services that implement the addNotification method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/

What's new

New in Version 8.3 — “[Report Email Alerts](#)” on page 1900

This method can now be used with [bibus » baseReport](#) objects.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asyncReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

addTermAssociation(term, parameterValues, options)

Reserved.

References

Part of the following method sets:

- [indexTerm](#)

Implemented by the following services:

The services that implement the addTermAssociation method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Input parameters

Use the following parameters when calling this method.

term

This argument

- is of type [bibus » indexTerm](#)
is encoded as type `tns:indexTerm`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

back(conversation, parameterValues, options)

Use this method to retrieve the previous prompt page.

This method may return the following values in [bibus » asynchReply » status](#) property:

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply back(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply back(asynchRequest conversation, parameterValue[]
    parameterValues, option[] options)
```

References

Part of the following method sets:

- [promptPaging](#)

Implemented by the following services:

The services that implement the back method, along with the associated SOAP actions, are listed in the following table.

Table 23. Services implementing the back method.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.high
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/ indexSearchService/201404/.high
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/ powerPlayService/201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/.high

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the previous prompt page.

If [complete](#) is returned in the [bibus » asynchReply » status](#) property, this method returns a value of [prompting](#) in the [bibus » asynchDetailReportStatus » status](#) property. An instance of this class will be returned in the [bibus » asynchReply » details](#) property.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

background(conversation)

Use this method to complete execution of a task in the background.

The monitor service creates a [bibus » history](#) object to record information about the conversation and then calls the `asynch » wait(conversation, parameterValues, options)` method against the appropriate service until the conversation is complete. If the service processing the request fails, the monitor service sets the [bibus » runStatusEnum](#) enumeration set to [failed](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply background(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation)
```

C# .NET

```
public asynchReply background(asynchRequest conversation)
```

References

Part of the following method sets:

- [monitor](#)

Implemented by the following services:

The services that implement the background method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
monitorService	<code>http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/</code>

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

cancel(conversation)

Use this method to request that the asynchronous conversation be canceled. No other secondary requests can follow this action.

Signatures

Java and Apache Axis

```
public void cancel(  
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation)
```

C# .NET

```
public void cancel(asynchRequest conversation)
```

Example: Using the cancel(conversation) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [asynch » cancel\(conversation\)](#) method with the report service.

To see this code in context, view the Java sample `CancelExec/Cancel.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
connect.getReportService().cancel(rsr.getPrimaryRequest());
```

Example: Using the cancel(conversation) Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the [asynch » cancel\(conversation\)](#) method with the report service.

To see this code in context, view the C# sample `Cancel/CancelReport.cs`. For more information about the samples, see Chapter 18, “Code samples and language-specific coding practices,” on page 1419.

```
runResponse = cBIRS.run(reportPathSO, paramValueArray, arrReportRunOpts);
cBIRS.cancel(runResponse.primaryRequest);
```

References

Part of the following method sets:

- [asynch](#)

Implemented by the following services:

The services that implement the cancel method, along with the associated SOAP actions, are listed in the following table.

Table 25. Services implementing the cancel method.

Service	SOAP Action
agentService	<code>http://www.ibm.com/xmlns/prod/cognos/agentService/201404/.control</code>
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.control</code>
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/.control</code>
dataIntegrationService	<code>http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/.control</code>
dataMovementService	<code>http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/.control</code>
deliveryService	<code>http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/.control</code>
dimensionManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/.control</code>
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.control</code>
idVizService	<code>http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/.control</code>
indexSearchService	<code>http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.control</code>
indexUpdateService	<code>http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/.control</code>

Table 25. Services implementing the cancel method. (continued)

Service	SOAP Action
jobService	http://www.ibm.com/xmlns/prod/cognos/jobService/201404/.control
metadataService	http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.control
migrationService	http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/.control
mobileService	http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/.control
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.control
planningAdministrationConsoleService	http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/.control
planningRuntimeService	http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/.control
planningTaskService	http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/.control
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.control
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.control
relationalMetadataService	http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/.control
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.control
repositoryService	http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/.control
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/.control

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the `ibis » asynchReply » primaryRequest` property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

Return values

None.

cancelEvent(eventID)

Use this method to cancel an event.

Canceling an event prevents the specified event from completing normally. When canceled, the status of the event changes to [cancelled](#), [purged](#), or [terminated](#), depending on the state of the event when the method is called.

For more information about event status, see the [bibus » runStatusEnum](#) enumeration set and [“Running Jobs”](#) on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the [canUseMonitorActivityTool](#) capability.
This allows administrators to cancel tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to cancel their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to cancel tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void cancelEvent(java.lang.String eventID)
```

C# .NET

```
public void cancelEvent(string eventID)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `cancelEvent` method, along with the associated SOAP actions, are listed in the following table.

Table 26. Services implementing the `cancelEvent` method.

Service	SOAP Action
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – “Administration Capabilities for Background Job Queue Maintenance” on page 1877

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » `cancelEvents(eventIDs)` method

Input parameters

Use the following parameters when calling this method.

eventID

Identifies the event.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

None.

[cancelEvents\(eventIDs\)](#)

Use this method to cancel a set of events.

Canceling an event prevents the specified event from completing normally. When canceled, the status of the event changes to `cancelled`, `purged`, or `terminated`, depending on the state of the event when the method is called.

For more information about event status, see the [bibus](#) » `runStatusEnum` enumeration set and “Running Jobs” on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to cancel tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to cancel their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to cancel tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void cancelEvents(java.lang.String[] eventIDs)
```

C# .NET

```
public void cancelEvents(string[] eventIDs)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the cancelEvents method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance” on page 1877](#)

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » [cancelEvent\(eventID\)](#) method

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

Return values

None.

[clearCubeWorkloadLog\(cubeNames, parameterValues, options\)](#)

Clears aggregate workload log data for the specific ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply  
clearCubeWorkloadLog(java.lang.String[] cubeNames,  
com.cognos.developer.schemas.bibus._3.ParameterValue[]  
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
options)
```

C# .NET

```
public asynchReply clearCubeWorkloadLog(string[] cubeNames,  
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the clearCubeWorkloadLog method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Administration” on page 1836](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the ROLAP cubes to be cleared from the aggregate workload log. If no names are provided, then all cubes configured for the query service receiving the request are cleared.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

clearNotifications(objectPath)

Use this method to clear the object's alert list.

This method returns no [bibus » asynchDetail](#) objects in the [bibus » asynchReply » details](#) property.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- write permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
```



```
clearNotifications(  
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
objectPath)
```

C# .NET

```
public asyncReply clearNotifications(searchPathSingleObject  
objectPath)
```

Example: Using the clearNotifications(objectPath) Method with the Delivery Service in Java

The following Java code snippet demonstrates how to use the [delivery » clearNotifications\(objectPath\)](#) method with the delivery service.

To see this code in context, view the Java sample Alerts/ManageAlerts.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asyncReply = connection.getDeliveryService().clearNotifications(  
new SearchPathSingleObject(reportPath));
```

References

Part of the following method sets:

- [delivery](#)

Implemented by the following services:

The services that implement the clearNotifications method, along with the associated SOAP actions, are listed in the following table.

Table 29. Services implementing the clearNotifications method.

Service	SOAP Action
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/

What's new

New in Version 8.3 — “[Report Email Alerts](#)” on page 1900

This method can now be used with [bibus » baseReport](#) objects.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

collectParameterValues(objectPath, parameterValues, options)

Use this method to collect values for all parameters that must be specified before the report can run.

The values are stored in the [parameters](#) property. An instance of this class is returned in the [details](#) property when the request is complete.

If the request first generates a prompt page, the page is stored in an instance of [bibus » asynchDetailReportOutput](#) which is returned in the [details](#) property.

This method may return the following values in the [status](#) property:

- [asynchReplyStatusEnum » conversationComplete](#) value
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Use of this method requires:

- [traverse](#) permission for all ancestors of the object
- [execute](#) permission for the object
- [read](#) or [execute](#) permission for the metadata

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
collectParameterValues(
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply collectParameterValues(searchPathSingleObject
objectPath, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [parameter](#)

Implemented by the following services:

The services that implement the `collectParameterValues` method, along with the associated SOAP actions, are listed in the following table.

Table 30. Services implementing the `collectParameterValues` method.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added. It replaces the [report » collectParameterValues\(objectPath, parameterValues, options\)](#) – obsolete method.

powerPlayService information

This information is specific to the [powerPlayService](#) service.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

The documentation has been updated to show that this service now supports this method.

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

collectParameterValues(objectPath, parameterValues, options) – obsolete

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is not represented in the WSDL document

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is not represented in the WSDL document

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is not represented in the WSDL document

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
 - is not represented in the WSDL document

collectParameterValuesSpecification(specification, parameterValues, options)

Use this method to collect values for all parameters that must be specified for the report or metadata specification before it can be run.

The values are stored in the [bibus » asynchDetailParameterValues » parameters](#) property. An instance of this class is returned in the [bibus » asynchReply » details](#) property when the request is complete.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You can make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
collectParameterValuesSpecification(
com.cognos.developer.schemas.bibus._3.AsynchSpecification
specification, com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply collectParameterValuesSpecification(
asynchSpecification specification, parameterValue[] parameterValues,
option[] options)
```

References

Part of the following method sets:

- [parameter](#)

Implemented by the following services:

The services that implement the `collectParameterValuesSpecification` method, along with the associated SOAP actions, are listed in the following table.

Table 31. Services implementing the `collectParameterValuesSpecification` method.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

If the request first generates a prompt page, the page is stored in an instance of `bibus » asynchDetailReportOutput`, which is returned in the `bibus » asynchReply » details` property.

Use of this method with this service requires:

- read or execute permission for the metadata

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 32. Capability rules for the `collectParameterValuesSpecification` method.

Specification	Capabilities	Class
Trusted	<code>bibus » userCapabilityEnum » canUseAdaptiveAnalytics</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseAnalysisStudio</code>	<code>bibus » reportServiceAnalysisSpecification, bibus » reportServiceMetadataSpecification, or bibus » reportServiceReportSpecification</code>
—	<code>bibus » userCapabilityEnum » canUseControllerStudio</code>	<code>bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseEventStudio</code>	<code>bibus » reportServiceMetadataSpecification</code>
—	<code>bibus » userCapabilityEnum » canUseEventStudio</code>	<code>bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseMetricsManagerAdministration</code>	<code>bibus » reportServiceMetadataSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseMetricStudio</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUsePlanningContributor</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>

Table 32. Capability rules for the collectParameterValuesSpecification method. (continued)

Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseQueryStudio	bibus » reportServiceMetadataSpecification , bibus » reportServiceQuerySpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseReportStudio	bibus » reportServiceInteractiveReportSpecification , bibus » reportServiceMetadataSpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseSpecifications	bibus » reportServiceAnalysisSpecification , bibus » reportServiceInteractiveReportSpecification , bibus » reportServiceMetadataSpecification , bibus » reportServiceQuerySpecification , or bibus » reportServiceReportSpecification

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

The capability rules were updated to include the rules related to the use of the [bibus » reportServiceReportSpecification](#) class with the [userCapabilityEnum » canUseDashboardViewer](#) value user capability.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added. It replaces the following methods:

- `CognosReportNet » collectParameterValues(spec, options)`
- `report » collectParameterValuesSpecification(specification, parameterValues, options)`

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This method can now be used with a [bibus » reportServiceInteractiveReportSpecification](#).

The capability rules were updated to include the rules related to the use of the [bibus » reportServiceInteractiveReportSpecification](#) class.

powerPlayService information

This information is specific to the [powerPlayService](#) service.

This service does not support the [parameter » collectParameterValuesSpecification\(specification, parameterValues, options\)](#).

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

The documentation has been updated to show that this service now supports this method.

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added.

Input parameters

Use the following parameters when calling this method.

specification

Defines the specification associated with the request.

This argument

- is of type [bibus » asynchSpecification](#)
is encoded as type `tns:asynchSpecification`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

collectParameterValuesSpecification(specification, parameterValues, options) – obsolete

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method is obsolete and was removed. Use the [parameter » collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method instead.

Input parameters

Use the following parameters when calling this method.

specification

Defines the specification associated with the request.

This argument

- is of type [bibus » asynchSpecification](#)
is not represented in the WSDL document

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is not represented in the WSDL document

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is not represented in the WSDL document

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is not represented in the WSDL document

convertDrillThroughContext(inputContext, parameterValues, options)

Use this method to convert a selection context from one format to another.

The converted selection context is stored in the `bibus » asyncDetailSelectionContext » context` property. An instance of this class is returned in the `bibus » asyncReply » details` property when the request is complete.

This method may return the following values in the `bibus » asyncReply » status` property:

- `conversationComplete`
- `working`

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply  
convertDrillThroughContext(  
    com.cognos.developer.schemas.bibus._3.XmlEncodedXML inputContext,  
    com.cognos.developer.schemas.bibus._3.ParameterValue[]  
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
    options)
```

C# .NET

```
public asyncReply convertDrillThroughContext(xmlEncodedXML  
inputContext, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [drillThrough](#)

Implemented by the following services:

The services that implement the `convertDrillThroughContext` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/</code>

What's new

New in Version 8.4 — [“Drill-Through Improvements” on page 1889](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

inputContext

Specifies the selection context to be converted.

This argument

- is of type [bibus](#) » [xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

copy(objects, targetPath, options)

Use this method to copy objects within the content store from one location to another.

This method returns the [searchPath](#), [defaultName](#), and [storeID](#) of the objects copied. An error is returned if no objects are selected.

A copy operation generates a new value for the [bibus](#) » [baseClass](#) » [searchPath](#) property of the copied object and assigns a new, unique [storeID](#). If an object containing an ID-based reference is copied and the referenced object is also copied, the reference contained within the copied object references the copy of the object referenced by the source object. Path-based references are not updated but will continue to work providing the search path is still valid.

To assign new names to the copies of the objects, use the `content » copyRename(objects, targetPath, newNames, options)` method.

Use of this method requires:

- read permission for the source object
- traverse permission for all ancestors of the source object
- traverse permission for the source object if it has child objects and the `bibus » copyOptions » recursive` property is `true`
- write permission for the target container object
- traverse permission for the target container object and all its ancestors
- write permission for the object being overwritten if one already exists in the new location

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] copy(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    targetPath, com.cognos.developer.schemas.bibus._3.CopyOptions options)
```

C# .NET

```
public baseClass[] copy(baseClass[] objects, searchPathSingleObject
    targetPath, copyOptions options)
```

Example: Using the `copy(objects, targetPath, options)` Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the `content » copy(objects, targetPath, options)` method with the Content Manager service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
return connection.getCMSService().copy(
    bc, new SearchPathSingleObject(targetPath), new CopyOptions());
```

Example: Using the `copy(objects, targetPath, options)` Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the `content » copy(objects, targetPath, options)` method with the Content Manager service.

To see this code in context, view the C# sample `Copy/Copy.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
baseClass[] bcaCopyResults = cBICMS.copy(bcaCopy, targetPathS0, cpOptions);
if (bcaCopyResults.GetLength(0) > 0)
{
    //returns the number of successfully copied baseClass objects
    resultMessage =
        "...the report has been successfully copied to : " +
        targetPath +
        ".\n";
    return true;
}
```

```
}
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the copy method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects to be copied.

The [bibus](#) » [baseClass](#) » [searchPath](#) property of each object is used to select the object.

This argument

- is an array of type [bibus](#) » [baseClass](#)
is encoded as type `tns:baseClassArray`

targetPath

Specifies the target location for the copies of the objects. This parameter must select a single container object that must be writable in the current security context.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

options

Specifies the options for the [content](#) » [copy\(objects, targetPath, options\)](#) method.

This argument

- is of type [bibus](#) » [copyOptions](#)
is encoded as type `tns:copyOptions`

Return values

This method returns the following values.

result

Returns the [bibus](#) » [baseClass](#) » [defaultName](#) property, the [bibus](#) » [baseClass](#) » [searchPath](#) property, and the [bibus](#) » [baseClass](#) » [storeID](#) property for each copy of an object.

This result

- is an array of type [bibus » baseClass](#)
is encoded as type `tns:baseClassArray`

copyAccount(sourceAccountPath, targetAccountPath, options)

Use this method to copy the contents of one [bibus » account](#) to another account.

Use the [bibus » copyAccountOptionEnum](#) enumeration set to specify the options for this method.

If necessary, Content Manager will create the target account in the content store, provided that a corresponding user account exists in the security provider. If the target account already exists in the content store, Content Manager copies the contents of the source account to a folder within the target account.

The [owner](#) property of copied objects is set to the target account if the original object was owned by the source account.

This method does not copy any [bibus » credential](#) or [bibus » session](#) objects contained by the source [bibus » account](#).

Use of this method requires:

- read permission for the source account
- traverse permission for all ancestors of the source account
- traverse and read permission for the source account if the [folders](#) or or [pages](#) or options are `true`
- traverse permission for all ancestors of the target account
- write permission for the target account if it already exists in the new location
- write permission for the parent of the target account if it does not already exist in the new location

Signatures

Java and Apache Axis

```
public void copyAccount(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    sourceAccountPath,  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    targetAccountPath, com.cognos.developer.schemas.bibus._3.Option[]  
    options)
```

C# .NET

```
public void copyAccount(searchPathSingleObject sourceAccountPath,  
    searchPathSingleObject targetAccountPath, option[] options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `copyAccount` method, along with the associated SOAP actions, are listed in the following table.

Table 35. Services implementing the copyAccount method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

sourceAccountPath

Identifies the source account object for the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

targetAccountPath

Identifies the target account object for the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

None.

[copyRename\(objects, targetPath, newNames, options\)](#)

Use this method to copy objects to another location in the content store under a different name.

This method returns the [searchPath](#), [defaultName](#), and [storeID](#) of the objects copied. An error is returned if no objects are selected.

A copy operation generates a new value for the [bibus](#) » [baseClass](#) » [searchPath](#) property of the copied object and assigns a new, unique [storeID](#). If an object containing an ID-based reference is copied and the referenced object is also copied, the reference contained within the copied object references the copy of the object referenced by the source object. Path-based references are not updated but will continue to work providing the search path is still valid.

Use of this method requires:

- read permission for the source object
- traverse permission for all ancestors of the source object

- traverse permission for the source object if it has child objects and the `bibus » copyOptions » recursive` property is true
- write permission for the target container object
- traverse permission for the target container object and all its ancestors
- write permission for the object being overwritten if one already exists in the new location

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] copyRename(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    targetPath, java.lang.String[] newNames,
    com.cognos.developer.schemas.bibus._3.CopyOptions options)
```

C# .NET

```
public baseClass[] copyRename(baseClass[] objects,
    searchPathSingleObject targetPath, string[] newNames, copyOptions
    options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `copyRename` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects to be copied.

The `bibus » baseClass » searchPath` property of each object is used to select the object.

This argument

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

targetPath

Specifies the target location for the copied objects. This parameter must select a single container object that is writable in the current security context.

This argument

- is of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObject`

newNames

Specifies the new names for the copies of the objects.

Objects copied recursively are not renamed.

When objects are renamed, the `bibus » baseClass » name` property is replaced with the new name in the language specified by the locale of the request.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

options

Specifies the options for the `content » copyRename(objects, targetPath, newNames, options)` method.

This argument

- is of type `bibus » copyOptions`
is encoded as type `tns:copyOptions`

Return values

This method returns the following values.

result

Returns the `bibus » baseClass » defaultName` property, the `bibus » baseClass » searchPath` property, and the `bibus » baseClass » storeID` property for each copy of an object.

This result

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

currentPage(conversation, parameterValues, options)

Use this method to reretrieve the current page of the result set.

This method may return the following values in the `bibus » asyncReply » status` property:

- `complete`
- `working`

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply currentPage(
    com.cognos.developer.schemas.bibus._3.AsyncRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asyncReply currentPage(asyncRequest conversation,
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [paging](#)

Implemented by the following services:

The services that implement the `currentPage` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high</code>
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.high</code>
indexSearchService	<code>http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.high</code>
monitorService	<code>http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.high</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to reretrieve the current page of report output. The output is stored in an instance of `bibus » asyncDetailReportOutput`, which is returned in the `bibus » asyncReply » details` property when the request is complete.

eventManagementService information

This information is specific to the [eventManagementService](#) service.

This service does not support the `paging » currentPage\(conversation, parameterValues, options\)` method.

New in Version 8.3 — “[Schedule Management](#)” on page 1907

The documentation has been updated to indicate that this service does not support this method.

monitorService information

This information is specific to the [monitorService](#) service.

This service does not support the `paging » currentPage\(conversation, parameterValues, options\)` method.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service does not support this method.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the `bibus » asynchReply » primaryRequest` property.

The value of this parameter should be obtained from the most recent instance of the `bibus » asynchReply » primaryRequest` property returned from a method for the same conversation.

This argument

- is of type `bibus » asynchRequest`
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type `bibus » option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the `bibus » asynchReplyStatusEnum` enumeration set.

Any data requested can be obtained by examining the `bibus » asynchReply » details` property.

This result

- is of type `bibus » asynchReply`
is encoded as type `tns:asynchReply`

[delayEventsFor\(eventIDs, for\)](#)

Use this method to delay a set of events for a specified duration.

Delaying a set of events postpones their execution for the duration specified by the `for` parameter. The status of the events change to `scheduled`.

For more information about event status, see the [bibus » runStatusEnum](#) enumeration set and [“Running Jobs”](#) on page 76.

Use of this method requires:

- traverse permission for the ancestors of the target object that are associated with the event
- one of the following conditions to be met:
 1. The caller has the [canUseMonitorActivityTool](#) capability.
This allows administrators to hold tasks that are initiated by other users.
 2. The passport ID of the caller must match the account that is associated with the task.
This allows users to hold their own tasks that are running in the background.
 3. The account of the caller must match the account credential that was used to schedule the task.
This allows users to hold tasks that they had previously scheduled.

Signatures

Java and Apache Axis

```
public void delayEventsFor(java.lang.String[] eventIDs,  
java.lang.String _for)
```

C# .NET

```
public void delayEventsFor(string[] eventIDs, string @for)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `delayEventsFor` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 10.1.0 – [“Delaying Scheduled Tasks”](#) on page 1852

This method was added.

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

for

Specifies the duration of the delay for the events.

You must specify this parameter as a positive value using the XML Schema type `xs:duration` form.

This argument

- is of type `duration`
is encoded as type `xs:string`

Return values

None.

delayEventsUntil(eventIDs, until)

Use this method to postpone a set of events to a later time.

Delaying a set of events postpones their execution until the time specified by the `event » delayEventsUntil(eventIDs, until) » until` parameter. The status of the events change to `scheduled`.

For more information about event status, see the `bibus » runStatusEnum` enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for the ancestors of the target object that are associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to hold tasks that are initiated by other users.
 - The passport ID of the caller must match the account that is associated with the task.
This allows users to hold their own tasks that are running in the background.
 - The account of the caller must match the account credential that was used to schedule the task.
This allows users to hold tasks that they had previously scheduled.

Signatures

Java and Apache Axis

```
public void delayEventsUntil(java.lang.String[] eventIDs,  
java.util.Calendar until)
```

C# .NET

```
public void delayEventsUntil(string[] eventIDs, System.DateTime until)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `delayEventsUntil` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – “[Delaying Scheduled Tasks](#)” on page 1852

This method was added.

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

until

Specifies the new execution time for the tasks.

The new time must be later than the current execution time, as indicated by `scheduledStartTime` property. An instance of `asynchDetailEventRecord` class is returned in the `details` property for each event when the request is complete.

This argument

- is of type `dateTime`
is encoded as type `xs:dateTime`

Return values

None.

delete(objects, options)

Use this method to delete objects from the content store.

This method returns the number of objects deleted from the content store. An error is not returned if no objects are selected.

When you delete a shortcut, the object it references is not deleted.

If optimistic concurrency control is used, the `bibus » baseClass » version` property is compared to the `version` of the affected objects in the content store. For more information, see [“Concurrency Control” on page 55](#).

Use of this method requires:

- write permission for the object and its parent object
- traverse permission for all ancestors of the object
- traverse permission for the object if the object has child objects and the `bibus » deleteOptions » recursive` property is `true`

If the `bibus » deleteOptions » force` property is `true` and the user has `setPolicy` permission for the object, write and traverse permissions are not required for the object.

Signatures

Java and Apache Axis

```
public int delete(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.DeleteOptions options
)
```

C# .NET

```
public int delete(baseClass[] objects, deleteOptions options)
```

Example: Using the `delete(objects, options)` Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the `content » delete(objects, options)` method with the Content Manager service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,” on page 1419](#).

```
DeleteOptions del = new DeleteOptions();
del.setForce(true);

int i = connection.getCMSservice().delete(new BaseClass[] { bc }, del);
```

Example: Using the `delete(objects, options)` Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the `content » delete(objects, options)` method with the Content Manager service.

To see this code in context, view the C# sample `PrintReport/PrintReport.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,” on page 1419](#).

```
int result = connection.CBICMS.delete(bcaDeletePrinter, delete_options);
if (0 >= result)
{
    returnMessage = "Error occured while deleting the printer.";
    return false;
}
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the delete method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects to be deleted from the content store.

The [bibus](#) » [baseClass](#) » [searchPath](#) property of each object is used to select the object.

By default, deleting an object that is referenced by another object clears the reference. For information about changing the default behavior, see the [bibus](#) » [deleteOptions](#) » [faultIfObjectReferenced](#) property.

This argument

- is an array of type [bibus](#) » [baseClass](#)
is encoded as type `tns:baseClassArray`

options

Specifies the options for the [content](#) » [delete\(objects, options\)](#) method.

This argument

- is of type [bibus](#) » [deleteOptions](#)
is encoded as type `tns:deleteOptions`

Return values

This method returns the following values.

result

Returns the number of objects deleted from the content store.

This result

- is of type `int`
is encoded as type `xs:int`

[delete\(objectPath, parameterValues, options\)](#)

Reserved.

References

Part of the following method sets:

- [indexUpdate](#)

Implemented by the following services:

The services that implement the delete method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
 - is encoded as type `tns:asynchReply`

deleteAccount(objectPath, options)

Use this method to delete an account from the content store. If the user is logged on, the user's passport is invalidated before the account is deleted.

If the user logs on again after the account is deleted and the user account is still valid in the security provider, the account will be recreated in the content store with default properties that are stored in the [“User Profile” on page 1681](#) account.

Use of this method requires:

- write permission for the [bibus » account](#) object and its parent object
 - Note:** A user who has write permission to a [bibus » namespace](#) is able to delete any account in that namespace.
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public void deleteAccount(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    objectPath, com.cognos.developer.schemas.bibus._3.Option[] options)
```

C# .NET

```
public void deleteAccount(searchPathSingleObject objectPath, option[]  
options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the deleteAccount method, along with the associated SOAP actions, are listed in the following table.

Table 42. Services implementing the deleteAccount method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

None.

deleteAllNotifications()

Use this method to delete the current user from all object alert lists.

This method returns no [bibus](#) » [asynchDetail](#) objects in the [bibus](#) » [asynchReply](#) » [details](#) property.

This method may return the following values in the [bibus](#) » [asynchReply](#) » [status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- read permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
deleteAllNotifications()
```

C# .NET

```
public asyncReply deleteAllNotifications()
```

Example: Using the deleteAllNotifications() Method with the Delivery Service in Java

The following Java code snippet demonstrates how to use the [delivery » deleteAllNotifications\(\)](#) method with the delivery service.

To see this code in context, view the Java sample Alerts/ManageAlerts.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asyncReply =  
    connection.getDeliveryService().deleteAllNotifications();
```

References

Part of the following method sets:

- [delivery](#)

Implemented by the following services:

The services that implement the deleteAllNotifications method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/

What's new

New in Version 8.3 — [“Report Email Alerts” on page 1900](#)

This method can now be used with [bibus » baseReport](#) objects.

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asyncReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asyncReply » details](#) property.

This result

- is of type [bibus » asyncReply](#)
is encoded as type `tns:asyncReply`

deleteHotList(objectPath)

Use this method to delete the [bibus » agentOutputHotList](#) objects for the specified [bibus » baseAgentDefinition](#).

Use of this method requires:

- write permission to the bibus » [baseAgentDefinition](#) object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public void deleteHotList(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    objectPath)
```

C# .NET

```
public void deleteHotList(searchPathSingleObject objectPath)
```

References

Part of the following method sets:

- [agent](#)

Implemented by the following services:

The services that implement the deleteHotList method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
agentService	http://www.ibm.com/xmlns/prod/cognos/agentService/201404/

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

None.

deleteNotification(objectPath)

Use this method to delete the current user from an object's alert list. If the user is not on the object's alert list, this method returns a fault.

This method returns no [bibus » asyncDetail](#) objects in the [bibus » asyncReply » details](#) property.

This method may return the following values in the [bibus » asyncReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- read permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
deleteNotification(
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
objectPath)
```

C# .NET

```
public asynchReply deleteNotification(searchPathSingleObject
objectPath)
```

Example: Using the deleteNotification(objectPath) Method with the Delivery Service in Java

The following Java code snippet demonstrates how to use the [delivery » deleteNotification\(objectPath\)](#) method with the delivery service.

To see this code in context, view the Java sample Alerts/ManageAlerts.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asynchReply = connection.getDeliveryService().deleteNotification(
new SearchPathSingleObject(reportPath));
```

References

Part of the following method sets:

- [delivery](#)

Implemented by the following services:

The services that implement the deleteNotification method, along with the associated SOAP actions, are listed in the following table.

<i>Table 45. Services implementing the deleteNotification method.</i>	
Service	SOAP Action
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/

What's new

New in Version 8.3 — “Report Email Alerts” on page 1900

This method can now be used with [bibus](#) » [baseReport](#) objects.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

deleteTenants(tenantIDs)

Use this method to delete content from one or more tenants in the content store, including user accounts and My Folders content. If any user from the tenant list is logged on, the user's passport is invalidated before the account is deleted. If the user logs on again after the account is deleted and the user account is still valid in the security provider, the account will be re-created in the content store with the default properties that are stored in the `User Profile` object account.

Signatures

Java and Apache Axis

```
public void deleteTenants(java.lang.String[] tenantIDs)
```

C# .NET

```
public void deleteTenants(string[] tenantIDs)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `deleteTenants` method, along with the associated SOAP actions, are listed in the following table.

Table 46. Services implementing the deleteTenants method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

What's new

New in Version 10.2.0 – [“Support for multi-tenancy” on page 1838](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

tenantIDs

Identifies the tenants whose contents will be deleted from the content store.

This argument

- is an array of type string
is encoded as type tns:stringArray

Return values

None.

[deleteTermAssociation\(term, parameterValues, options\)](#)

Reserved.

References

Part of the following method sets:

- [indexTerm](#)

Implemented by the following services:

The services that implement the deleteTermAssociation method, along with the associated SOAP actions, are listed in the following table.

Table 47. Services implementing the deleteTermAssociation method.

Service	SOAP Action
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/

What's new

New in Version 8.3 – [“Search – For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Input parameters

Use the following parameters when calling this method.

term

This argument

- is of type [bibus](#) » [indexTerm](#)
is encoded as type `tns:indexTerm`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

[deliver\(conversation, parameterValues, options\)](#)

Use this method to deliver report output to any destination.

This secondary request can use the following run options to specify the destination of the output. Multiple destinations can be specified in the same request.

- [archive](#)

Use this option to save the report output to a file system.

You must set the Archive Location File System Root parameter in IBM Cognos Configuration before using this option. See the *IBM Cognos Analytics Installation and Configuration Guide* for more information.

- [email](#)

Use this option to email the report output.

The request can specify the following additional run options:

- [emailAsAttachment](#)
- [emailAsURL](#)

Use options listed in the [bibus » deliveryOptionEnum](#) enumeration set to specify addressing and subject information for the email, as well as any additional content.

You can use the [emailAsURL](#) run option if the output was saved to the content store using the [saveOutput](#) or the [saveAs](#) run options. If both options are set, the output created by the [saveOutput](#) run option is emailed.

The [emailAsURL](#) option specifies that a link to the stored output should be included in the email. If a specification is run, report output cannot be stored in the content store. Therefore, the [emailAsURL](#) option cannot be used when a specification is run.

- [mobile](#)

Use this option to send the report output to IBM Cognos Analytics Mobile Reports users.

Use the [to mobile](#) option to specify the recipients of the report output.

- [print](#)

Use this option to request that a report be printed.

The request can specify the following additional run options:

- [printer](#)
- [printerAddress](#)

Use of this option requires:

- read permission for the [bibus » printer](#) object if the run option [printer](#) is specified

- [saveAs](#)

Use this option to save the report output as part of a new or existing [bibus » reportView](#) in the content store. If you are saving to an existing [bibus » reportView](#), it must reference the same report as the one being executed.

The [bibus » reportView](#) may be in a location that is the same or different from the original report. If the location is the same, then the name must be different.

Use of this option requires:

- write permission for the parent of the new object
- traverse permission for all ancestors of the new object

- [saveOutput](#)

Use this option to save the report output in the content store. This option may be used only if the report exists in the content store.

Use of this option requires:

- traverse permission for all ancestors of the report in the content store
- write permission for the report in the content store

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)

This method normally terminates the asynchronous conversation. However, the run option [continueConversation](#) may be specified to avoid terminating the conversation.

- [complete](#)

This status can only be returned if the run option [continueConversation](#) is `true`.

- [working](#)

You may be able to make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply deliver(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply deliver(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the `deliver(conversation, parameterValues, options)` Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [report » `deliver\(conversation, parameterValues, options\)`](#) method with the report service.

To see this code in context, view the Java sample `SendEmail/Email.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asynchReply = connection.getReportService().deliver(
    response, new ParameterValue[] {}, emailRunOptions);
```

Example: Using the `deliver(conversation, parameterValues, options)` Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the [report » `deliver\(conversation, parameterValues, options\)`](#) method with the report service.

To see this code in context, view the C# sample `Email/Email.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asynchReply sendEmailResponse = connection.CBIRS.run(
    reportPath,
    new parameterValue[] {},
    emailRunOptions);
connection.CBIRS.deliver(
    sendEmailResponse.primaryRequest, new parameterValue[] { },
    emailOptions);
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the deliver method, along with the associated SOAP actions, are listed in the following table.

Table 48. Services implementing the deliver method.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/.high

What's new

New in Version 8.4 – [“Report Server deliver\(\) method” on page 1883](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

determineRouting(objectPaths)

Use this method to determine the name of the server group that should handle requests that use the specified objects.

If a routing rule is matched this method returns the value of the [bibus » routingTableEntry » serverGroupName](#) property of the matched rule. Otherwise, `nil` is returned.

Use of this method requires:

- read permission for each object specified
- traverse permissions for all ancestors of each object specified

Objects for which the user lacks sufficient permissions will be omitted from the subsequent routing calculation

Signatures

Java and Apache Axis

```
public java.lang.String determineRouting(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject[]  
    objectPaths)
```

C# .NET

```
public string determineRouting(searchPathSingleObject[] objectPaths)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `determineRouting` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Input parameters

Use the following parameters when calling this method.

objectPaths

Specifies the objects associated with the request.

This argument

- is an array of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObjectArray`

Return values

This method returns the following values.

result

Returns the server group name best suited to run the request.

This result

- is of type `string`
is encoded as type `xs:string`

drill(conversation, parameterValues, options)

Use this method to drill in the same report in IBM Cognos Viewer. Drill has to be explicitly enabled in a report.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply drill(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply drill(asynchRequest conversation, parameterValue[]
    parameterValues, option[] options)
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the drill method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.high</code>

Table 50. Services implementing the drill method. (continued)

Service	SOAP Action
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

findDrillThroughPaths(objectPath, parameterValues, options)

Use this method to find the set of drill paths in the package that can be used with the source selection context. Specify the `bibus » package` in the `objectPath` parameter, and provide the selection context in a `selectionContext` option.

The drill paths are returned in instances of the classes `bibus » asynchDetailDrillThroughTarget` and `bibus » asynchDetailDrillThroughTargetURI` in the `bibus » asynchReply » details` property when the request is complete.

This method may return the following values in the `bibus » asynchReply » status` property:

- `conversationComplete`
- `working`

You may be able to make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Use of this method requires:

- read permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
findDrillThroughPaths(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply findDrillThroughPaths(searchPathSingleObject
objectPath, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [drillThrough](#)

Implemented by the following services:

The services that implement the `findDrillThroughPaths` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.session</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.session</code>

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[firstPage\(conversation, parameterValues, options\)](#)

Use this method to retrieve the first page of the result set.

This method may return the following values in the [bibus » asynchReply » status](#) property :

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417 .

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply firstPage(  
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,  
    com.cognos.developer.schemas.bibus._3.ParameterValue[]  
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
    options)
```

C# .NET

```
public asynchReply firstPage(asynchRequest conversation,  
    parameterValue[] parameterValues, option[] options)
```

Example: Using the firstPage(conversation, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [paging » firstPage\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample ExecReports/RunReport.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
rsr =  
    connection.getReportService().firstPage(  
        rsr.getPrimaryRequest(),  
        new ParameterValue[] {},  
        new Option[] {});
```

References

Part of the following method sets:

- [paging](#)

Implemented by the following services:

The services that implement the firstPage method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.high
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/ eventManagementService/201404/.high
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/ indexSearchService/201404/.high
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/ 201404/.high

Table 52. Services implementing the firstPage method. (continued)

Service	SOAP Action
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the first page of report output. The output is stored in an instance of [bibus » asynchDetailReportOutput](#) , which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

eventManagementService information

This information is specific to the [eventManagementService](#) service.

Use this method to retrieve the first page of the result set. An instance of [bibus » asynchDetailEventRecord](#) is returned in the [bibus » asynchReply » details](#) property for each event when the request is complete.

New in Version 8.3 – “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

monitorService information

This information is specific to the [monitorService](#) service.

Use this method to retrieve the first page of the result set. An instance of [bibus » asynchDetailEventRecord](#) is returned in the [bibus » asynchReply » details](#) property for each event when the request is complete.

New in Version 8.3 – “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

forward(conversation, parameterValues, options)

Use this method to retrieve the next prompt page, or the first page of the result set if there are no additional prompt pages.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply forward(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply forward(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [promptPaging](#)

Implemented by the following services:

The services that implement the forward method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.high
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the next prompt page, or the first page of report output if there are no additional prompt pages. The output is stored in an instance of [bibus » asynchDetailReportOutput](#), which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

If [complete](#) is returned in the [bibus » asynchReply » status](#) property and there is an additional prompt page, this method returns a value of [prompting](#) in the [bibus » asynchDetailReportStatus » status](#) property. An instance of this class will be returned in the [bibus » asynchReply » details](#) property.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[get\(objectPath, parameterValues, options\)](#)

Reserved.

References

Part of the following method sets:

- [indexUpdate](#)

Implemented by the following services:

The services that implement the `get` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
indexUpdateService	<code>http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/</code>

What's new

New in Version 8.3 — [“Search — For Internal Use Only”](#) on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

[getActiveContentManager\(\)](#)

Use this method to determine the location of the active Content Manager service.

A Content Manager service in the [running](#), [standby](#), or [suspended](#) state can respond to this request.

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `getActiveContentManager` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
<code>contentManagerService</code>	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Return values

This method returns the following values.

result

Returns the active Content Manager URI.

This result

- is of type `anyURI`

is encoded as type `xs:string`

getConfiguration(properties)

Use this method to retrieve global configuration data.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.ConfigurationData
getConfiguration(
com.cognos.developer.schemas.bibus._3.ConfigurationDataEnum[]
properties)
```

C# .NET

```
public configurationData getConfiguration(configurationDataEnum[]
properties)
```

Example: Using the `getConfiguration(properties)` Method with the System Service in Java

The following Java code snippet demonstrates how to use the `system » getConfiguration(properties)` method with the system service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
data = connection.getSystemService().getConfiguration(config);
locales = data.getServerLocale();
```

Example: Using the `getConfiguration(properties)` Method with the System Service in C# .NET

The following C# code snippet demonstrates how to use the `system » getConfiguration(properties)` method with the system service.

To see this code in context, view the C# sample `AddReport/AddReport.cs`. For more information about the samples, see Chapter 18, “Code samples and language-specific coding practices,” on page 1419.

```
data = cBISS.getConfiguration(config);
locales = data.serverLocale;

if (locales == null)
{
    locales[0] = new locale();
    locales[0].locale1 = "en";
}
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `getConfiguration` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
systemService	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

properties

Specifies the set of properties to be returned.

This argument

- is an array of type [bibus » configurationDataEnum](#)
is encoded as type `tns:configurationDataEnumArray`

Return values

This method returns the following values.

result

Returns the configuration data.

This result

- is of type [bibus » configurationData](#)
is encoded as type `tns:configurationData`

[getContext\(conversation, parameterValues, options\)](#)

Use this method to obtain context information for a running report.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply getContext(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply getContext(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the `getContext` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high</code>

What's new

New in Version 8.3 — [“Improved Context Metadata for Selection” on page 1911](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[getCubeMessages\(cubeNames, parameterValues, options\)](#)

Use this method to retrieve recent error and warning messages for a cube.

Signatures

Java and Apache Axis

```
public RolapCubeMessages[] getCubeMessages(java.lang.String[] cubeNames,  
com.cognos.developer.schemas.bibus._3.ParameterValue[]  
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
options)
```

C# .NET

```
public rolapCubeMessages[] getCubeMessages(string[] cubeNames,  
parameterValue[]  
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the `getCubeMessages` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.2.2 – [Dynamic cube messages](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the dynamic cubes to retrieve messages for.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type `bibus » option`

is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the cube names and associated messages.

This result

- is an array of type `bibus » rolapCubeMessages`
is encoded as type `tns:rolapCubeMessagesArray`

getCubeMetrics(cubeNames, parameterValues, options)

Retrieves metrics for a cube.

Signatures

Java and Apache Axis

```
public RolapCubeMetrics[] getCubeMetrics(java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public RolapCubeMetrics[] getCubeMetrics(string[] cubeNames, parameterValue[]
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the `getCubeMetrics` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

What's new

New in Version 10.2.2 – [Dynamic cube metrics](#)

This method was added.

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus » userCapabilityEnum » canUseDataSourcesTool](#)
- [bibus » userCapabilityEnum » canUseServerAdministrationTool](#)

What's new

New in Version 10.2.2 – [Dynamic cube metrics](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes for which metrics are to be retrieved.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the cube names and associated metrics.

This result

- is an array of type `bibus » rolapCubeMetrics`
is encoded as type `tns:rolapCubeMetricsArray`

[**getCubeState\(cubeNames, parameterValues, options\)**](#)

Use this method to retrieve the current state of ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
getCubeState(java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply getCubeState(string[] cubeNames, parameterValue[]
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the `getCubeState` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the ROLAP cubes to be interrogated.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

getDeploymentOptions(archive, options)

Use this method to retrieve the deployment options from the archive specified as the [archive](#) parameter. The deployment archives location is an IBM Cognos Analytics startup parameter.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.Option[]
getDeploymentOptions(java.lang.String archive,
com.cognos.developer.schemas.bibus._3.Option[] options)
```

C# .NET

```
public option[] getDeploymentOptions(string archive, option[] options)
```

Example: Using the getDeploymentOptions(archive, options) Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [deployment » getDeploymentOptions\(archive, options\)](#) method with the Content Manager service.

To see this code in context, view the Java sample `DeployPackage/Deployment.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
public HashMap getPubFolderContent(String myArchive,
    CRNConnect myConnection) {

    Option[] deployOptEnum = new Option[] {};
    HashMap arrOfPublicFolder = new HashMap();

    try {
        deployOptEnum = myConnection.getCMSService().getDeploymentOptions(
            myArchive, new Option[] {});
    }
}
```

References

Part of the following method sets:

- [deployment](#)

Implemented by the following services:

The services that implement the `getDeploymentOptions` method, along with the associated SOAP actions, are listed in the following table.

<i>Table 61. Services implementing the <code>getDeploymentOptions</code> method.</i>	
Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseContentStoreTool](#)

Input parameters

Use the following parameters when calling this method.

archive

Specifies the name of the archive from which the deployment options are retrieved. The archive name contains the relative path to the deployment location.

This argument

- is of type `string`
is encoded as type `xs:string`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type `bibus` » `option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the set of options stored in the archive.

This result

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

getFormatSamples(name)

Use this method to retrieve sample format specifications.

Possible choices are based on the user's locale preferences. If no format specifications are listed in the user's content locale, the method retrieves the specifications listed in the user's product locale.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.XmlEncodedXML[]
getFormatSamples(java.lang.String name)
```

C# .NET

```
public xmlEncodedXML[] getFormatSamples(string name)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `getFormatSamples` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
systemService	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

name

Specifies the name used to identify a particular set of sample format specifications.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

This method returns the following values.

result

Returns the sample format specifications.

This result

- is an array of type [bibus](#) » [xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLArray`

getIdentity()

Returns the user's identities for the current session.

Signatures

Java and Apache Axis

```
public Identity getIdentity()
```

C# .NET

```
public Identity getIdentity()
```

References

Part of the following method sets:

- [standaloneCAM](#)

Implemented by the following services:

The services that implement the `terminateSessions` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
saCAMService	<code>http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/</code>

What's new

New in Version 10.2.2 – [Full tenant impersonation capability for system administrators](#)

This method was added.

Return values

This method returns the following values.

result

Returns the user's identities for the current session.

This result

- is of type [bibus](#) » [identity](#)
is encoded as type `tns:identity`

getObjectContext(objectPath, parameterValues, options)

Use this method to obtain context information for an [bibus](#) » [output](#) object.

This method may return the following values in the [bibus](#) » [asynchReply](#) » [status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- [traverse](#) permission for all ancestors of the object
- [execute](#) permission for the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
getObjectContext(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply getObjectContext(searchPathSingleObject
objectPath, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the `getObjectContext` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/

What's new

New in Version 8.3 — [“Improved Context Metadata for Selection”](#) on page 1911

This method was added.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[getOutput\(conversation, parameterValues, options\)](#)

Use this method to retrieve the prepared output page from the server. The output is stored in an instance of [bibus » asynchDetailReportOutput](#), which is returned in the [bibus » asynchReply » details](#) property. This method is used to retrieve report output only and should be called only when the value of the [bibus » asynchDetailReportStatus » status](#) property is `responseReady`.

This method always returns the value `complete` in the [bibus » asynchReply » status](#) property.

If the output is a prompt page, this method returns the value `prompting` in the [bibus » asynchDetailReportStatus » status](#) property. An instance of this class will be returned in the [bibus » asynchReply » details](#) property.

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply getOutput(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply getOutput(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the `getOutput(conversation, parameterValues, options)` Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [report » `getOutput\(conversation, parameterValues, options\)`](#) method with the report service.

To see this code in context, view the Java sample `ExecReports/RunReport.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
rsr =
    connection.getReportService().getOutput(
        rsr.getPrimaryRequest(),
        new ParameterValue[] {},
        new Option[] {});
```

Example: Using the `getOutput(conversation, parameterValues, options)` Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the [report » `getOutput\(conversation, parameterValues, options\)`](#) method with the report service.

To see this code in context, view the C# sample `ExecuteReport/ExecuteReport.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
executeReportResponse = cBIConnection.CBIRS.run(cmReportPath, arrParm,
arrRunOpts);
// If response is not immediately complete, call wait until complete
if (!executeReportResponse.status.Equals(asynchReplyStatusEnum.complete))
{
    ...
    if (outputIsReady(executeReportResponse))
    {
        executeReportResponse =
            cBIConnection.CBIRS.getOutput(
                executeReportResponse.primaryRequest,
                new parameterValue[] {},
                new option[] {});
    }
    ...
    asynchDetailReportOutput reportOutput = null;
    for (int i = 0; i < executeReportResponse.details.Length; i++)
```

```

{
  if (executeReportResponse.details[i] is asynchDetailReportOutput)
  {
    reportOutput =
      (asynchDetailReportOutput)executeReportResponse.details[i];
    break;
  }
}

```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the `getOutput` method, along with the associated SOAP actions, are listed in the following table.

<i>Table 65. Services implementing the <code>getOutput</code> method.</i>	
Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.absolute</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/.absolute</code>

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus](#) » [asynchReply](#) » [primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus](#) » [asynchReply](#) » [primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus](#) » [asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[getParameters\(objectPath, parameterValues, options\)](#)

Use this method to return the list of parameters used by a runnable object, including optional parameters. This method also returns parameters from the model and stored procedures that are used by the object.

The parameters are stored in the [bibus » asynchDetailParameters » parameters](#) property. An instance of this class is returned in the [bibus » asynchReply » details](#) property when the request is complete.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
getParameters(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply getParameters(searchPathSingleObject objectPath,
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [parameter](#)

Implemented by the following services:

The services that implement the `getParameters` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/ powerPlayService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

If the request first generates a prompt page for connection information, the page is stored in an instance of the `ibis » asynchDetailReportOutput` class, which is returned in the `ibis » asynchReply » details` property.

Use of this method requires:

- traverse permission for all ancestors of the object
- execute permission for the object
- read or execute permission for the metadata

The following Java code snippet demonstrates how to use the `parameter » getParameters(objectPath, parameterValues, options)` method with the report service.

To see this code in context, view the Java sample `ReportParams/ReportParameters.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
response = connection.getReportService().getParameters(
    reportPath, new ParameterValue[] { }, new Option[] { });
...
for (int i = 0; i < response.getDetails().length; i++)
{
    if (response.getDetails()[i] instanceof AsynchDetailParameters)
    {
        params = ((AsynchDetailParameters)response.getDetails()
[i]).getParameters();
    }
}
```

The following C# code snippet demonstrates how to use the `parameter » getParameters(objectPath, parameterValues, options)` method with the report service.

To see this code in context, view the C# sample `ReportParameters/ReportParameters.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asynchReply gpReply = cBIRS.getParameters(
```

```

        cmReportPath, new parameterValue[] {}, new runOption[]{} );
    ...
    for (int i = 0; i < gpReply.details.Length; i++)
    {
        if (gpReply.details[i] is asynchDetailParameters)
        {
            return ((asynchDetailParameters)gpReply.details[i]).parameters;
        }
    }
}

```

powerPlayService information

This information is specific to the [powerPlayService](#) service.

Use of this method requires:

- traverse permission for all ancestors of the object
- execute permission for the object
- read or execute permission for the metadata

New in Version 10.1.0 – “[New PowerPlay Service Methods](#)” on page 1860

The documentation has been updated to show that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

getParametersSpecification(specification, parameterValues, options)

Use this method to return the list of parameters used by the runnable object, including optional parameters. This method also returns parameters from the model and stored procedures that are used by the object.

The parameters are stored in the [bibus » asynchDetailParameters » parameters](#) property. An instance of this class is returned in the [bibus » asynchReply » details](#) property when the request is complete.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Use of this method requires:

- read or execute permission for the metadata

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
getParametersSpecification(
com.cognos.developer.schemas.bibus._3.AsynchSpecification
specification, com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply getParametersSpecification(asynchSpecification
specification, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [parameter](#)

Implemented by the following services:

The services that implement the `getParametersSpecification` method, along with the associated SOAP actions, are listed in the following table.

Table 67. Services implementing the `getParametersSpecification` method.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

If the request first generates a prompt page for connection information, the page is stored in an instance of the `bibus » asynchDetailReportOutput` class, which is returned in the `bibus » asynchReply » details` property.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 68. Capability rules for the `getParametersSpecification` method.

Specification	Capabilities	Class
Trusted	<code>bibus » userCapabilityEnum » canUseAdaptiveAnalytics</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseAnalysisStudio</code>	<code>bibus » reportServiceAnalysisSpecification, bibus » reportServiceMetadataSpecification, or bibus » reportServiceReportSpecification</code>
—	<code>bibus » userCapabilityEnum » canUseControllerStudio</code>	<code>bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseEventStudio</code>	<code>bibus » reportServiceMetadataSpecification</code>
—	<code>bibus » userCapabilityEnum » canUseEventStudio</code>	<code>bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseMetricsManagerAdministration</code>	<code>bibus » reportServiceMetadataSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseMetricStudio</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUsePlanningContributor</code>	<code>bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification</code>
Trusted	<code>bibus » userCapabilityEnum » canUseQueryStudio</code>	<code>bibus » reportServiceMetadataSpecification, bibus » reportServiceQuerySpecification, or bibus » reportServiceReportSpecification</code>

Table 68. Capability rules for the `getParametersSpecification` method. (continued)

Specification	Capabilities	Class
—	<code>bibus » userCapabilityEnum » canUseReportStudio</code>	<code>bibus » reportServiceInteractiveReportSpecification</code> , <code>bibus » reportServiceMetadataSpecification</code> , or <code>bibus » reportServiceReportSpecification</code>
—	<code>bibus » userCapabilityEnum » canUseSpecifications</code>	<code>bibus » reportServiceAnalysisSpecification</code> , <code>bibus » reportServiceInteractiveReportSpecification</code> , <code>bibus » reportServiceMetadataSpecification</code> , <code>bibus » reportServiceQuerySpecification</code> , or <code>bibus » reportServiceReportSpecification</code>

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch`).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch/ext`).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

The capability rules were updated to include the rules related to the use of the `bibus » reportServiceReportSpecification` class with the `userCapabilityEnum » canUseDashboardViewer` value user capability.

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This method can now be used with a `bibus » reportServiceInteractiveReportSpecification`.

The capability rules were updated to include the rules related to the use of the `bibus » reportServiceInteractiveReportSpecification` class.

powerPlayService information

This information is specific to the `powerPlayService` service.

This service does not support the `getParametersSpecification(specification, parameterValues, options)` method.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

The documentation has been updated to show that this service does not support this method.

Input parameters

Use the following parameters when calling this method.

specification

Defines the specification associated with the request.

This argument

- is of type [bibus » asynchSpecification](#)
is encoded as type `tns:asynchSpecification`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[getPromptValues\(conversation, parameterValues, options\)](#)

Use this method to request candidate prompt values for hierarchical prompts.

The prompt information is returned in the [bibus » asynchDetailReportOutput » outputPages](#) property.

The prompt values to be returned represent the child nodes of the nodes specified in the [promptPaging » getPromptValues\(conversation, parameterValues, options\) » parameterValues](#) parameter.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
getPromptValues(com.cognos.developer.schemas.bibus._3.AsynchRequest
conversation, com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asyncReply getPromptValues(asyncRequest conversation,
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [promptPaging](#)

Implemented by the following services:

The services that implement the `getPromptValues` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high</code>
indexSearchService	<code>http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.high</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.high</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Two run options can be used to limit the number of child nodes returned to the caller, [skipValueCount](#) and [maximumValueCount](#).

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added. It replaces the [report](#) » `getPromptValues(conversation, parameterValues, options)` – obsolete method.

powerPlayService information

This information is specific to the [powerPlayService](#) service.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

The documentation has been updated to show that this service now supports this method.

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method was added.

New in Version 10.1.0 – “Supporting Fail-over for Tree Prompts” on page 1861

The affinity of this method was changed to high to support server fail-over.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

getPromptValues(conversation, parameterValues, options) – obsolete

What's new

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

This method is obsolete and was removed. Use the [promptPaging » getPromptValues\(conversation, parameterValues, options\)](#) method instead.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is not represented in the WSDL document

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is not represented in the WSDL document

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » option](#)
is not represented in the WSDL document

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is not represented in the WSDL document

holdEvent(eventID)

Use this method to hold an event.

Holding an event prevents it from executing. The status of the event changes to [suspended](#) and can be released by calling [event » releaseEvent\(eventID\)](#).

For more information about event status, see the [bibus » runStatusEnum](#) enumeration set and [“Running Jobs”](#) on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the [canUseMonitorActivityTool](#) capability.
This allows administrators to hold tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to hold their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to hold tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void holdEvent(java.lang.String eventID)
```

C# .NET

```
public void holdEvent(string eventID)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the holdEvent method, along with the associated SOAP actions, are listed in the following table.

<i>Table 70. Services implementing the holdEvent method.</i>	
Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 10.1.0 – “Administration Capabilities for Background Job Queue Maintenance” on page 1877

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » [holdEvents\(eventIDs\)](#) method

Input parameters

Use the following parameters when calling this method.

eventID

Identifies the event.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

None.

holdEvents(eventIDs)

Use this method to hold a set of events.

Holding a set of events prevents them from executing. The status of the events change to [suspended](#) and can be released by calling [event](#) » [releaseEvent\(eventID\)](#).

For more information about event status, see the [bibus](#) » [runStatusEnum](#) enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for the ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to hold tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to hold their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to hold tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void holdEvents(java.lang.String[] eventIDs)
```

C# .NET

```
public void holdEvents(string[] eventIDs)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the holdEvents method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance” on page 1877](#)

This method now requires the caller to have the [canUseMonitorActivityTool](#) value capability instead of the [canUseAdministrationPortal](#) value capability when a capability check is required.

Related information:

- [event](#) » [holdEvent\(eventID\)](#) method

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

Return values

None.

[incrementallyLoadCubes\(cubeNames, parameterValues, options\)](#)

Updates the aggregate cache and data cache to reflect newly-added fact rows.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply
```

```
incrementallyLoadCubes(java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply incrementallyLoadCubes(string[] cubeNames,
parameterValue[]
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the incrementallyLoadCubes method, along with the associated SOAP actions, are listed in the following table.

<i>Table 72. Services implementing the incrementallyLoadCubes method.</i>	
Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

What's new

New in Version 10.2.2 – [New options to manage dynamic cubes](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes to be paused.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type string
is encoded as type tns:stringArray

parameterValues

Inlcude the TID (transaction ID) value used as the basis for updating a cube's data values from the underlying fact table as a parameter. The parameter name is transactionID.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type tns:parameterValueArray

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

lastPage(conversation, parameterValues, options)

Use this method to retrieve the last page of the result set.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply lastPage(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply lastPage(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the lastPage(conversation, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [paging » lastPage\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample `ExecReports/RunReport.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
rsr =
    connection.getReportService().lastPage(
        rsr.getPrimaryRequest(),
        new ParameterValue[] { },
        new Option[] { });
```

References

Part of the following method sets:

- [paging](#)

Implemented by the following services:

The services that implement the `lastPage` method, along with the associated SOAP actions, are listed in the following table.

<i>Table 73. Services implementing the lastPage method.</i>	
Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high</code>
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.high</code>
indexSearchService	<code>http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.high</code>
monitorService	<code>http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.high</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the last page of report output. The output is stored in an instance of [bibus » asynchDetailReportOutput](#), which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

If the request causes a query that has not already run to do so, and if that query must prompt the user for new parameter values, this method returns a value of [prompting](#) in the [bibus » asynchDetailReportStatus » status](#) property. An instance of this class will be returned in the [bibus » asynchReply » details](#) property.

eventManagementService information

This information is specific to the [eventManagementService](#) service.

Use this method to retrieve the last page of the result set. An instance of `bibus » asynchDetailEventRecord` is returned in the `bibus » asynchReply » details` property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

monitorService information

This information is specific to the `monitorService` service.

Use this method to retrieve the last page of the result set. An instance of `bibus » asynchDetailEventRecord` is returned in the `bibus » asynchReply » details` property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the `bibus » asynchReply » primaryRequest` property.

The value of this parameter should be obtained from the most recent instance of the `bibus » asynchReply » primaryRequest` property returned from a method for the same conversation.

This argument

- is of type `bibus » asynchRequest`
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

lineage(conversation, parameterValues, options)

Use this method during an asynchronous conversation that is running a report to obtain lineage information for data items and layout expressions that contain data items.

This secondary request should include the context of the selected object and layout expression. The `querySet` response from `reportService` must be sent to the `metadataService`, using the `V5QuerySet` object, for further processing of lineage information.

This method may return the following values in the [bibus » asynchReply » status](#) property:

1. [complete](#)
2. [stillWorking](#)
3. [working](#)

See [Chapter 25, “Using lineage specifications,” on page 1509](#) for more information.

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the lineage method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus » userCapabilityEnum » canUseLineage](#)

What's new

New in Version 10.1.0 – [“Support for Lineage Requests” on page 1855](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

listArchives()

Use this method to retrieve the list of archives in the deployment archives location. The archive location is an IBM Cognos Analytics startup parameter.

Signatures

Java and Apache Axis

```
public java.lang.String[] listArchives()
```

C# .NET

```
public string[] listArchives()
```

Example: Using the listArchives() Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [deployment](#) » [listArchives\(\)](#) method with the Content Manager service.

To see this code in context, view the Java sample `DeployPackage/Deployment.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
public String[] getListOfArchives(CRNConnect connection) {  
    String[] listOfArchives = null;  
    try {  
        listOfArchives = connection.getCMService().listArchives();  
    }  
}
```

References

Part of the following method sets:

- [deployment](#)

Implemented by the following services:

The services that implement the `listArchives` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseContentStoreTool](#)

Return values

This method returns the following values.

result

Returns the archive files from the deployment archives location. The archive location is an IBM Cognos Analytics startup parameter. Each archive name identifies the relative path to the deployment location.

This result

- is an array of type `string`

is encoded as type `tns:stringArray`

listTenants(options)

Use this method to query the tenants information from the content store. When querying a content store that hasn't been enabled for multi-tenancy, only the public tenant information will be returned.

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `listTenants` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

What's new

New in Version 10.2.0 – [“Support for multi-tenancy” on page 1838](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

options

Specifies the options for the request, defined in [bibus](#) » [listTenantsOptions](#) class.

This argument

- is of type [bibus](#) » [listTenantsOptions](#)
is encoded as type `tns:listTenantsOptions`

Return values

This method returns the following values.

result

Returns the tenants information from the content store. When querying a content store that hasn't been enabled for multi-tenancy, only the public tenant information will be returned.

This result

- is an array of type [bibus](#) » [tenantInfo](#)
is encoded as type `tns:tenantInfoArray`

logout()

Use this method to log off through the SDK. If the action is successful, the [bibus](#) » [biBusHeader](#) passport, located in the [bibus](#) » [CAM](#) » [CAMPassport](#) property, is removed.

Signatures

Java and Apache Axis

```
public void logoff()
```

C# .NET

```
public void logoff()
```

Example: Using the logoff() Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [authentication](#) » [logoff\(\)](#) method with the Content Manager service.

To see this code in context, view the Java sample `Security/Logon.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
connection.getCMService().logoff();
```

Example: Using the logoff() Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the [authentication](#) » [logoff\(\)](#) method with the Content Manager service.

To see this code in context, view the C# sample `Security/Security.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
cBIConnection.CBICMS.logoff();
```

References

Part of the following method sets:

- [authentication](#)

Implemented by the following services:

The services that implement the logoff method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Return values

None.

[logon\(credentials, roles\)](#)

Use this method to log on through the SDK. If the user is authenticated by a security provider, the action is successful and a passport is created in the [bibus](#) » [biBusHeader](#).

You can log on to more than one namespace without logging off. However, you cannot log on to the same namespace more than once.

For more information about logging on, see [“Authenticating Users”](#) on page 45.

Signatures

Java and Apache Axis

```
public void logon(com.cognos.developer.schemas.bibus._3.XmlEncodedXML
credentials,
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject[] roles)
```

C# .NET

```
public void logon(xmlEncodedXML credentials, searchPathSingleObject[]
roles)
```

Example: Using the `logon(credentials, roles)` Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [authentication » `logon\(credentials, roles\)`](#) method with the Content Manager service.

To see this code in context, view the Java sample `Security/Logon.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
StringBuffer credentialXML = new StringBuffer();
credentialXML.append("<credential>");
credentialXML.append("<namespace>");
credentialXML.append(namespace);
credentialXML.append("</namespace>");
credentialXML.append("<username>");
credentialXML.append(uid);
credentialXML.append("</username>");
credentialXML.append("<password>");
credentialXML.append(pwd);
credentialXML.append("</password>");
credentialXML.append("</credential>");
String encodedCredentials = credentialXML.toString();
credentialString = encodedCredentials;
connection.getCMService().logon(new XmlEncodedXML(encodedCredentials),
new SearchPathSingleObject[] { });
```

Example: Using the `logon(credentials, roles)` Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the [authentication » `logon\(credentials, roles\)`](#) method with the Content Manager service.

To see this code in context, view the C# sample `SamplesCommon/SamplesLogon.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
System.Text.StringBuilder credentialXML =
new System.Text.StringBuilder("<credential>");
credentialXML.AppendFormat("<namespace>{0}</namespace>", userNamespace);
credentialXML.AppendFormat("<username>{0}</username>", userName);
```

```

credentialXML.AppendFormat( "<password>{0}</password>", userPassword );
credentialXML.Append( "</credential>" );

//The csharp toolkit encodes the credentials
string encodedCredentials = credentialXML.ToString ();
xmlEncodedXML xmlEncodedCredentials = new xmlEncodedXML();
xmlEncodedCredentials.Value = encodedCredentials;
searchPathSingleObject[] emptyRoleSearchPathList = new
searchPathSingleObject[0];
cBICMS.logon(xmlEncodedCredentials, null);

```

References

Part of the following method sets:

- [authentication](#)

Implemented by the following services:

The services that implement the logon method, along with the associated SOAP actions, are listed in the following table.

<i>Table 78. Services implementing the logon method.</i>	
Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

credentials

Specifies the credentials that are used to authenticate the user. This parameter contains the same XML information as the [bibus](#) » [credential](#) » [credentials](#) property.

For more information about the XML structure for credentials, see the [bibus](#) » [credential](#) » [credentials](#) property.

This argument

- is of type [bibus](#) » [xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

roles

Specifies the set of search paths for the roles that will be assigned to the current session.

This argument

- is an array of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

Return values

None.

mapContentLocale(locale, normalize)

Use this method to obtain the supported content locale that corresponds to the specified locale. IBM Cognos determines the supported locale by looking up the specified locale in the content locale mapping table. You can modify these mappings using IBM Cognos Configuration.

If you have not used the [system » normalizeLocale\(locale\)](#) method on the supplied locale, set the `normalize` parameter to `true` when you call this method.

Signatures

Java and Apache Axis

```
public java.lang.String mapContentLocale(java.lang.String locale,
boolean normalize)
```

C# .NET

```
public string mapContentLocale(string locale, bool normalize)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `mapContentLocale` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
systemService	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

locale

Specifies the locale to be mapped to a supported content locale.

This argument

- is of type `string`
is encoded as type `xs:string`

normalize

Specifies whether to normalize the locale before mapping it to a supported content locale.

This argument

- is of type `boolean`
is encoded as type `xs:boolean`

Return values

This method returns the following values.

result

Returns the supported content locale.

This result

- is of type `language`
is encoded as type `xs:string`

mapProductLocale(locale, normalize)

Use this method to obtain the supported product locale that corresponds to the specified locale. IBM Cognos determines the supported locale by looking up the specified locale in the product locale mapping table. You can modify these mappings using IBM Cognos Configuration.

If you have not used the [system » normalizeLocale\(locale\)](#) method on the supplied locale, set the `normalize` parameter to `true` when you call this method.

Signatures

Java and Apache Axis

```
public java.lang.String mapProductLocale(java.lang.String locale,
boolean normalize)
```

C# .NET

```
public string mapProductLocale(string locale, bool normalize)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `mapProductLocale` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
systemService	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

locale

Specifies the locale to be mapped to a supported product locale.

This argument

- is of type `string`

is encoded as type `xs:string`

normalize

Specifies whether to normalize the locale before mapping it to a supported product locale.

This argument

- is of type `boolean`

is encoded as type `xs:boolean`

Return values

This method returns the following values.

result

Returns the supported product locale.

This result

- is of type `language`

is encoded as type `xs:string`

move(objects, targetPath, options)

Use this method to move objects within the content store.

This method returns the [searchPath](#), [defaultName](#), and [storeID](#) of the objects that were moved. An error is returned if no objects are selected.

A move operation updates the [bibus](#) » [baseClass](#) » [searchPath](#) property of the target object.

ID-based references are maintained by Content Manager when moving objects. Path-based references to the object (for example, target objects of shortcuts) are not maintained by Content Manager when an object is moved, resulting in a broken reference.

If you want to assign new names to the moved objects, use the [content](#) » [moveRename\(objects, targetPath, newNames, options\)](#) method.

Use of this method requires:

- read and write permission for the source object
- write permission for the parent of the source object
- traverse permission for all ancestors of the source object
- write permission for the target container object
- traverse permission for the target container object and all its ancestors
- write permission for the object being overwritten if one already exists in the new location

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] move(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    targetPath, com.cognos.developer.schemas.bibus._3.MoveOptions options)
```

C# .NET

```
public baseClass[] move(baseClass[] objects, searchPathSingleObject
targetPath, moveOptions options)
```

Example: Using the move(objects, targetPath, options) Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the `content » move(objects, targetPath, options)` method with the Content Manager service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
return connection.getCMService().move(
    bc, new SearchPathSingleObject(targetPath), new MoveOptions());
```

Example: Using the move(objects, targetPath, options) Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the `content » move(objects, targetPath, options)` method with the Content Manager service.

To see this code in context, view the C# sample `Move/Move.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
baseClass[] bcaMoveResults = cBICMS.move(bcaMove, cmTargetPath, cpOptions);
if (bcaMoveResults.GetLength(0) > 0)
{
    //the number of successfully copied objects
    resultMessage =
        "...the report has been successfully moved to the target
location \" +
        targetPath +
        "\".";
    return true;
}
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the move method, along with the associated SOAP actions, are listed in the following table.

<i>Table 81. Services implementing the move method.</i>	
Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects to be moved to a new location in the content store.

The `bibus » baseClass » searchPath` property of each object is used to select the object.

This argument

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

targetPath

Specifies the target location for the moved objects. This parameter must select a single container object that is writable in the current security context.

This argument

- is of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObject`

options

Specifies the options for the `content » move(objects, targetPath, options)` method.

This argument

- is of type `bibus » moveOptions`
is encoded as type `tns:moveOptions`

Return values

This method returns the following values.

result

Returns the `bibus » baseClass » defaultName` property, the `bibus » baseClass » searchPath` property, and the `bibus » baseClass » storeID` property for each moved object.

This result

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

moveRename(objects, targetPath, newNames, options)

Use this method to move objects to a new location in the content store and rename them.

This method returns the `searchPath`, `defaultName`, and `storeID` of the objects that were moved. An error is returned if no objects are selected.

ID-based references are maintained by Content Manager when moving objects. Path-based references to the object (for example, target objects of shortcuts) are not maintained by Content Manager when an object is moved, resulting in a broken reference.

Use of this method requires:

- read and write permission for the source object
- write permission for the parent of the source object
- traverse permission for all ancestors of the source object
- write permission for the target container object

- traverse permission for the target container object and all its ancestors
- write permission for the object being overwritten if one already exists in the new location

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] moveRename(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    targetPath, java.lang.String[] newNames,
    com.cognos.developer.schemas.bibus._3.MoveOptions options)
```

C# .NET

```
public baseClass[] moveRename(baseClass[] objects,
    searchPathSingleObject targetPath, string[] newNames, moveOptions
    options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the moveRename method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects to be moved.

The [bibus](#) » [baseClass](#) » [searchPath](#) property of each object is used to select the object.

This argument

- is an array of type [bibus](#) » [baseClass](#)
is encoded as type `tns:baseClassArray`

targetPath

Specifies the target location for the moved objects. This parameter must select a single container object.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

newNames

Specifies the new names for the moved objects.

The name property of each object is replaced with the new name in the language specified by the locale of the request.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

options

Specifies the options for the `content » moveRename(objects, targetPath, newNames, options)` method.

This argument

- is of type `bibus » moveOptions`
is encoded as type `tns:moveOptions`

Return values

This method returns the following values.

result

Returns the `bibus » baseClass » defaultName` property, the `bibus » baseClass » searchPath` property, and the `bibus » baseClass » storeID` property for each moved object.

This result

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

nextPage(conversation, parameterValues, options)

Use this method to retrieve the next page of the result set.

This method may return the following values in the `bibus » asyncReply » status` property:

- `complete`
- `working`

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply nextPage(
    com.cognos.developer.schemas.bibus._3.AsyncRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asyncReply nextPage(asyncRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the nextPage(conversation, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [paging » nextPage\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample ExecReports/RunReport.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
rsr =
    connection.getReportService().nextPage(
        rsr.getPrimaryRequest(),
        new ParameterValue[] { },
        new Option[] { });
```

References

Part of the following method sets:

- [paging](#)

Implemented by the following services:

The services that implement the nextPage method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.high
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.high
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the next page of report output. The output is stored in an instance of [bibus » asynchDetailReportOutput](#), which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

If the request causes a query that has not already run to do so, and if that query must prompt the user for new parameter values, this method returns a value of [prompting](#) in the [bibus » asynchDetailReportStatus » status](#) property. An instance of this class will be returned in the [bibus » asynchReply » details](#) property.

eventManagementService information

This information is specific to the [eventManagementService](#) service.

Use this method to retrieve the next page of the result set. An instance of [bibus » asynchDetailEventRecord](#) is returned in the [bibus » asynchReply » details](#) property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

monitorService information

This information is specific to the [monitorService](#) service.

Use this method to retrieve the next page of the result set. An instance of [bibus » asynchDetailEventRecord](#) is returned in the [bibus » asynchReply » details](#) property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

normalizeLocale(locale)

Use this method to standardize the format of the locale information.

The input is processed using the following built-in rules:

- all underscores (`_`) are converted to hyphens (`-`)
- all characters are converted to lower case

The result of the processing of the input locale with the built-in rules is returned.

Signatures

Java and Apache Axis

```
public java.lang.String normalizeLocale(java.lang.String locale)
```

C# .NET

```
public string normalizeLocale(string locale)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `normalizeLocale` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
systemService	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

locale

Specifies the locale to be normalized into the standard format.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

This method returns the following values.

result

Returns the normalized locale.

This result

- is of type `language`
is encoded as type `xs:string`

pauseCubes(cubeNames, parameterValues, options)

Pauses a dynamic cube.

You can now pause a dynamic cube and incrementally update the cube data. Use the `incrementallyLoadCubes(cubeNames, parameterValues, options)` method to update the cubes and the `startCubes(cubeNames, parameterValues, options)` method to re-start the cubes.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply  
pauseCubes(java.lang.String[] cubeNames,  
com.cognos.developer.schemas.bibus._3.ParameterValue[]  
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
options)
```

C# .NET

```
public asyncReply pauseCubes(string[] cubeNames, parameterValue[]  
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the `pauseCubes` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.2.2 – [New options to manage dynamic cubes](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes to be paused.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

ping(dispatcherPath)

Use this method to determine whether a dispatcher is accessible and running.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.PingReply ping(
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
dispatcherPath)
```

C# .NET

```
public PingReply ping(SearchPathSingleObject dispatcherPath)
```

Example: Using the ping (dispatcherPath) Method with the Dispatcher Service in Java

The following Java code snippet demonstrates how to use the [dispatcher » ping\(dispatcherPath\)](#) method with the dispatcher service.

To see this code in context, view the Java sample DispatcherControl\Dispatcher.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
String pingResult = null;
SearchPathSingleObject myDispSearchPath = new SearchPathSingleObject(
dispSearchPathURL);
try {
    PingReply pingReplyResult = Con.getDispatcherService().ping(
myDispSearchPath);

    if (pingReplyResult != null) {
        pingResult = pingReplyResult.getVersion();
    }
}
```

References

Part of the following method sets:

- [dispatcher](#)

Implemented by the following services:

The services that implement the ping method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
dispatcher	http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201404/

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus » userCapabilityEnum » canUseServerAdministrationTool](#)

Input parameters

Use the following parameters when calling this method.

dispatcherPath

Identifies the dispatcher to ping.

Specify the location of the dispatcher that you want to ping. For example, `/configuration/dispatcher[@name='Test Server']`.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

This method returns the following values.

result

Returns the dispatcher state.

This result

- is of type [bibus » pingReply](#)
is encoded as type `tns:pingReply`

[previousPage\(conversation, parameterValues, options\)](#)

Use this method to retrieve the previous page of the result set.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [complete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply previousPage(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply previousPage(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the `previousPage(conversation, parameterValues, options)` Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [paging » previousPage\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample ExecReports/RunReport.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
rsr =
    connection.getReportService().previousPage(
        rsr.getPrimaryRequest(),
        new ParameterValue[] {},
        new Option[] {});
```

References

Part of the following method sets:

- [paging](#)

Implemented by the following services:

The services that implement the previousPage method, along with the associated SOAP actions, are listed in the following table.

<i>Table 87. Services implementing the previousPage method.</i>	
Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/.high
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/ eventManagementService/201404/.high
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/ indexSearchService/201404/.high
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/ 201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/.high

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

Use this method to retrieve the previous page of report output. The output is stored in an instance of [bibus » asyncDetailReportOutput](#), which is returned in the [bibus » asyncReply » details](#) property when the request is complete.

eventManagementService information

This information is specific to the [eventManagementService](#) service.

Use this method to retrieve the previous page of the result set. An instance of [bibus » asyncDetailEventRecord](#) is returned in the [bibus » asyncReply » details](#) property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

monitorService information

This information is specific to the [monitorService](#) service.

Use this method to retrieve the previous page of the result set. An instance of [bibus » asynchDetailEventRecord](#) is returned in the [bibus » asynchReply » details](#) property for each event when the request is complete.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)

is encoded as type `tns:asynchReply`

query(searchPath, properties, sortBy, options)

Use this method to retrieve objects from the content store.

To query an `ibibus » authoredReport` object in the content store, use the `report » query(objectPath, parameterValues, options)` method.

This method returns the specified properties of each selected object. Objects are not repeated within the results of the query.

Use of this method requires:

- read permission for the object and traverse permission for all ancestors of the object if the properties to be returned are not general properties
- traverse permission for all ancestors of the object if the properties to be returned are general properties
- `setPolicy` permission to read or write to the `ibibus » baseClass » policies` property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] query(
    com.cognos.developer.schemas.bibus._3.SearchPathMultipleObject
    searchPath, com.cognos.developer.schemas.bibus._3.PropEnum[]
    properties, com.cognos.developer.schemas.bibus._3.Sort[] sortBy,
    com.cognos.developer.schemas.bibus._3.QueryOptions options)
```

C# .NET

```
public baseClass[] query(searchPathMultipleObject searchPath,
    propEnum[] properties, sort[] sortBy, queryOptions options)
```

Example: Using the query(searchPath, properties, sortBy, options) Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the `content » query(searchPath, properties, sortBy, options)` method with the Content Manager service.

To see this code in context, view the Java sample `QueryCM/cmQuerySample.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
PropEnum props[] =
    new PropEnum[] {PropEnum.searchPath, PropEnum.defaultName,
    PropEnum.policies,
    PropEnum.userName, PropEnum.notificationEMail };
Account myAccount = null;
...
BaseClass bc[] =
    connection.getCMService().query(
        new SearchPathMultipleObject("~"), props, new Sort[] {},
    new QueryOptions());

if ((bc != null) && (bc.length > 0))
{
    for (int i = 0; i < bc.length; i++)
    {
        myAccount = (Account)bc[i];
    }
}
...

```



```

output =
    output.concat(
        "Your alert email address is: "
        + myAccount.getNotificationEMail().getValue());

```

Example: Using the query(searchPath, properties, sortBy, options) Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the `content » query(searchPath, properties, sortBy, options)` method with the Content Manager service.

To see this code in context, view the C# sample `PrintReport/PrintReport.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```

bcaDeletePrinter = connection.CBICMS.query(
    printerToDeleteSearchPath, props, new sort[] {}, new
    queryOptions());

```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the query method, along with the associated SOAP actions, are listed in the following table.

Table 88. Services implementing the query method.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Input parameters

Use the following parameters when calling this method.

searchPath

Specifies the search path string for Content Manager to locate the requested objects. The current user must have read permission for the requested objects.

For more information about specifying a search path, see [Chapter 32, “Search path syntax,”](#) on page 1565.

If you use the `“expandDistributionMembers(object_set)”` on page 1571 or the `“expandSecurityMembers(object_set)”` on page 1572 function in the search path, the query determines the accounts and contacts that are members of each `bibus » group`, `bibus » role`, or `bibus » distributionList` specified in the query. For each account, the query returns the properties of the account and a list of the objects that caused the account or contact to be included in the response.

At the end of the returned objects there is a list of the selected objects that contained no account or contact objects because they were not found or could not be read. A `recipientGroups` element is included in these error objects with a status element and no value element. The status element will contain a status of `notFound` or `noRead`.

If you use the “[expandDistributionMembers\(object_set\)](#)” on page 1571 or the “[expandSecurityMembers\(object_set\)](#)” on page 1572 function in the search path, you cannot specify the `sortBy` parameter in the query request, nor can you specify the following query options:

- [skipObjects](#)
- [maxObjects](#)
- [skipDataBytes](#)
- [maxDataBytes](#)

This argument

- is of type [bibus](#) » [searchPathMultipleObject](#)
is encoded as type `tns:searchPathMultipleObject`

properties

Specifies the set of properties to be returned. By default, the [searchPath](#) property is returned for each object.

If you specify a reference property, properties of the referenced object can also be returned. By default, only the [searchPath](#) property is returned for each referenced object. If the reference property is ID-based, the [storeID](#) property is also returned.

If you specify the [refProps](#) query option, only the [searchPath](#) property is returned for the referenced object. You must then specify additional properties if you want to have them returned for the referenced object.

If the [data](#) property is specified and [dataEncoding](#) is specified as [MIME](#) or [MIMECompressed](#), the data is returned in a MIME attachment. Otherwise, data is returned inline using [base64](#) encoding.

This argument

- is an array of type [bibus](#) » [propEnum](#)
is encoded as type `tns:propEnumArray`

sortBy

Specifies the sort criteria in an array. Each element of the array specifies a property by which the result set will be sorted and the sort direction, either ascending or descending.

The sort order is determined by the locale of the request.

This argument

- is an array of type [bibus](#) » [sort](#)
is encoded as type `tns:sortArray`

options

Specifies the options for the [content](#) » [query\(searchPath, properties, sortBy, options\)](#) method.

This argument

- is of type [bibus](#) » [queryOptions](#)
is encoded as type `tns:queryOptions`

Return values

This method returns the following values.

result

Returns the requested objects from the content store.

This result

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArray`

query(objectPath, parameterValues, options)

Use this method to retrieve a single `bibus » authoredReport` object from the content store.

The `bibus » authoredReport` object is stored in the `bibus » asynchDetailReportObject » report` property. An instance of this class is returned in the `bibus » asynchReply » details` property when the request is complete.

If the request first generates a prompt page for connection information, the page is stored in an instance of `bibus » asynchDetailReportOutput`, which is returned in the `bibus » asynchReply » details` property.

This method may return the following values in the `bibus » asynchReply » status` property:

- `complete`
- `conversationComplete`
- `working`

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

If the status is `complete` and connection information is required, this method returns a value of `prompting` in the `bibus » asynchDetailReportStatus » status` property. An instance of this class will be returned in the `bibus » asynchReply » details` property.

This method performs a number of additional activities after it retrieves the specified object from the content store by calling the `content » query(searchPath, properties, sortBy, options)` method:

- The report specification may be converted to a different format or upgraded to the current version, depending on the parameters provided. If a conversion occurs, the specification is validated. If the specification is not valid, this method will fail and an appropriate fault will be returned.
- For `bibus » analysis` objects, the following analysis specification values are set to reflect the object properties when the `bibus » reportServiceQueryOptionEnum » upgrade` property is `true`:
 - `bibus » baseClass » defaultName` property
 - `bibus » authoredReport » metadataModel` property
- For `bibus » query` objects, the following query specification values are set to reflect the object properties when the `bibus » reportServiceQueryOptionEnum » upgrade` property is `true`:
 - `bibus » baseClass » defaultName` property
 - `bibus » baseReport » executionPageDefinition` property
 - `bibus » baseReport » executionPageOrientation` property
 - `bibus » authoredReport » metadataModel` property
- For `bibus » report` objects, the following report specification values are set to reflect the object properties when the `bibus » reportServiceQueryOptionEnum » upgrade` property is `true`:
 - for Report Specification Version 7.0 and higher, the `<reportName>` element is set to the value of `bibus » baseClass » defaultName` property
 - `bibus » authoredReport » canBurst` property
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » paths` property
- For `bibus » report`, `bibus » interactiveReport`, and `bibus » reportTemplate` objects, the following report specification information is updated based on the specified object properties:

- for report specification Version 7.0 and higher, the <reportName> element is set to the value of [bibus](#) » [baseClass](#) » [defaultName](#) property
- the <modelPath> element is set to the value of [bibus](#) » [authoredReport](#) » [metadataModel](#) property
- all <reportPath> elements are updated based on corresponding entries in the [bibus](#) » [authoredReport](#) » [paths](#) property

If you want to retrieve multiple objects from the content store or do not need to convert the report specification, upgrade the report specification or synchronize the property values, use the [content](#) » [query\(searchPath, properties, sortBy, options\)](#) method.

Use of this method requires:

- read permission for the object and traverse permission for all ancestors of the object if the properties to be returned are not general properties
- traverse permission for all ancestors of the object if the properties to be returned are general properties
- setPolicy permission to read the [bibus](#) » [baseClass](#) » [policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply query(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asyncReply query(searchPathSingleObject objectPath,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the query(objectPath, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [report](#) » [query\(objectPath, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample `ReportSpec/EditReportSpec.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
AsyncReply qResult =
    connect.getReportService().query(
        new SearchPathSingleObject(reportPath),
        new ParameterValue[] {},
        qOpts);
...
// extract the report spec
if (qResult.getDetails() != null)
{
    for (int i = 0; i < qResult.getDetails().length; i++)
    {
        if (qResult.getDetails()[i] instanceof AsyncDetailReportObject)
        {
            reportSpec = ((AsyncDetailReportObject)qResult.getDetails()[i]).
                getReport().getSpecification().getValue();
        }
    }
}
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the query method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/

What's new

New in Version 10.1.0 – [“Use Report Name for Output File Name” on page 1865](#)

The documentation for this method was updated to indicate that the [bibus](#) » [baseClass](#) » [defaultName](#) property is used to populate the `reportName` element of the report specification.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
 - is encoded as type `tns:asynchReply`

queryDrillPath(objectPath, parameterValues, options)

Use this method to retrieve a single [bibus » drillPath](#) object from the content store.

The [bibus » drillPath](#) object is returned in the [bibus » asynchDetailDrillPathObject » drillPath](#) property. An instance of this class is returned in the [bibus » asynchReply » details](#) property when the request is complete.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [complete](#)
- [conversationComplete](#)
- [working](#)

You may be able to make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

This method performs a number of additional activities after it retrieves the specified object from the content store by calling the [content » query\(searchPath, properties, sortBy, options\)](#) method:

- The drill-through specification may be upgraded to the current version, depending on the parameters provided.
- The following drill-through specification values are set to reflect the object properties when the [bibus » queryReportOptions » upgrade](#) property is `true`:
 - [bibus » drillPath » deploymentReferences](#) property

If you want to retrieve multiple objects from the content store, or if you do not need to upgrade the drill-through specification or synchronize the property values, use the [content » query\(searchPath, properties, sortBy, options\)](#) method.

Use of this method requires:

- read permission for the object and traverse permission for all ancestors of the object if the properties to be returned are not general properties
- traverse permission for all ancestors of the object if the properties to be returned are general properties
- `setPolicy` permission to read the [bibus » baseClass » policies](#) property

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
queryDrillPath(
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
```

```
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asyncReply queryDrillPath(searchPathSingleObject objectPath,
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [drillThrough](#)

Implemented by the following services:

The services that implement the queryDrillPath method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/

What's new

New in Version 8.4 — [“Supporting New Drill-through Targets” on page 1890](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

queryMetadata(request)

This method is not supported.

IBM Cognos Framework Manager models are stored in XML files that validate against `BMTModelSpecification.xsd`. A copy of this schema is included in your installation in the location *installation location/templates/bmt/FMSDK*.

To extract metadata from the model files, you can process them locally with XSLT.

References

Part of the following method sets:

- [metadata](#)

Implemented by the following services:

The services that implement the `queryMetadata` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
metadataService	<code>http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/</code>

Input parameters

Use the following parameters when calling this method.

request

Specifies the requested metadata.

This argument

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

Return values

This method returns the following values.

result

Returns the metadata that satisfies the request.

This result

- is of type [bibus](#) » [xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

queryMultiple(requests)

Defines the implementation of the [content](#) » [query\(searchPath, properties, sortBy, options\)](#) method that accepts multiple requests.

Use this method to make multiple requests for objects from the content store.

To make a single request, use the [content](#) » [query\(searchPath, properties, sortBy, options\)](#) method.

Use of this method requires:

- read permission for the object and traverse permission for all ancestors of the object if the properties to be returned are not general properties
- traverse permission for all ancestors of the object if the properties to be returned are general properties
- `setPolicy` permission to read or write to the [bibus](#) » [baseClass](#) » [policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.QueryReply[]
queryMultiple(com.cognos.developer.schemas.bibus._3.QueryRequest[]
requests)
```

C# .NET

```
public queryReply[] queryMultiple(queryRequest[] requests)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the `queryMultiple` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/</code>

Input parameters

Use the following parameters when calling this method.

requests

Defines the set of requests.

Each element defines the set of inputs for a single request.

This argument

- is an array of type [bibus » queryRequest](#)
is encoded as type `tns:queryRequestArray`

Return values

This method returns the following values.

result

Returns the query results.

Each element defines the set of results for a single request.

This result

- is an array of type [bibus » queryReply](#)
is encoded as type `tns:queryReplyArray`

[queryMultipleCache\(requests, options\)](#)

Use this method to improve application performance by determining whether objects in the content store have changed since the previous query. If a set of objects has not changed, an application can take advantage of browser caching if the pages generated by the application are solely dependent on the objects and their property values.

If the [bibus » queryMultipleOptions » cacheValidator](#) property is an empty string, nil, or not specified, this method operates like the [content » queryMultiple\(requests\)](#) method. A cache validator is calculated and returned in the response. You can use this value in subsequent calls to this method to determine if the same response would be generated, allowing you to take advantage of cached browser pages.

If the [bibus » queryMultipleOptions » cacheValidator](#) property is specified, Content Manager will determine whether it is necessary to evaluate the specified queries based on the value of this property. If it is determined that the same results would be generated, the [bibus » queryMultipleResult » replies](#) property contains no entries and the specified cache validator value is returned for the [bibus » queryMultipleOptions » cacheValidator](#) property.

Each call to the [content » queryMultipleCache\(requests, options\)](#) method with a cache validator value must specify a set of queries that is identical to the queries used to determine the cache validator value. Failure to satisfy this requirement will result in undefined behavior.

Use of this method requires:

- read permission for the object and traverse permission for all ancestors of the object if the properties to be returned are not general properties
- traverse permission for all ancestors of the object if the properties to be returned are general properties
- `setPolicy` permission to read or write to the [bibus » baseClass » policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.QueryMultipleResult  
queryMultipleCache(  
    com.cognos.developer.schemas.bibus._3.QueryRequest[] requests,  
    com.cognos.developer.schemas.bibus._3.QueryMultipleOptions options)
```

C# .NET

```
public queryMultipleResult queryMultipleCache(queryRequest[]  
requests, queryMultipleOptions options)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the queryMultipleCache method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/ contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

requests

Specifies the queries to be evaluated.

This argument

- is an array of type [bibus](#) » [queryRequest](#)
is encoded as type tns:queryRequestArray

options

Specifies additional options for the method.

This argument

- is of type [bibus](#) » [queryMultipleOptions](#)
is encoded as type tns:queryMultipleOptions

Return values

This method returns the following values.

result

Returns the results of the queries and the cache validator value.

This result

- is of type [bibus » queryMultipleResult](#)
is encoded as type `tns:queryMultipleResult`

queryNotification(objectPath)

Use this method to determine whether the object supports alerts, and, if so, whether the current user is on the object's alert list.

An instance of the class [bibus » asyncDetailAgentNotificationStatus](#) is returned in the [bibus » asyncReply » details](#) property when the request is complete. If the object supports alerts, the [bibus » asyncDetailAgentNotificationStatus » status](#) property indicates whether the user is on the object's alert list with a value of either [on](#) or [off](#). If the object does not support alerts, this value of this property is [disabled](#).

This method may return the following values in the [bibus » asyncReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Use of this method requires:

- read permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply
queryNotification(
com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
objectPath)
```

C# .NET

```
public asyncReply queryNotification(searchPathSingleObject
objectPath)
```

Example: Using the queryNotification(objectPath) Method with the Delivery Service in Java

The following Java code snippet demonstrates how to use the [delivery » queryNotification\(objectPath\)](#) method with the delivery service.

To see this code in context, view the Java sample `Alerts/ManageAlerts.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asyncReply = connection.getDeliveryService().queryNotification(
new SearchPathSingleObject(reportPath));
```

References

Part of the following method sets:

- [delivery](#)

Implemented by the following services:

The services that implement the queryNotification method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
deliveryService	<code>http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/</code>

What's new

New in Version 8.3 – “[Report Email Alerts](#)” on page 1900

This method can now be used with [bibus](#) » [baseReport](#) objects.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

[queryTenantMembership\(tenantIDs\)](#)

Used to retrieve all tenant objects in Content Manager that have a members property referencing one of the tenant objects represented by the tenantIDs argument.

The method generates a set of tenantIDs by performing the following task for each tenantID referenced in the request argument:

- Find every tenant object under `/directory/tenants` whose members property references the tenant object represented by the tenantID.

Use of this method requires either trusted request or user being System Administrator

Signatures

Java and Apache Axis

```
public java.lang.String[] queryTenantMembership(java.lang.String[] tenantIDs)
```

C# .NET

```
public string[] queryTenantMembership(string[] tenantIDs)
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the deleteTenants method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

What's new

New in Version 10.2.2 – Support for delegated tenant administration

This method was added.

Input parameters

Use the following parameter when calling this method.

tenantIDs

A list of tenantIDs representing the tenant objects in CM.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

Return values

result

A list of tenantIDs.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

[refreshCubeDataCache\(cubeNames, parameterValues, options\)](#)

Use this method to refresh the data cache for ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply  
refreshCubeDataCache(java.lang.String[] cubeNames,  
com.cognos.developer.schemas.bibus._3.ParameterValue[]  
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
options)
```

C# .NET

```
public asynchReply refreshCubeDataCache(string[] cubeNames,  
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the refreshCubeDataCache method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes to be refreshed.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type string
is encoded as type tns:stringArray

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

[refreshCubeMemberCache\(cubeNames, parameterValues, options\)](#)

Use this method to refresh the member cache for ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
refreshCubeMemberCache(java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply refreshCubeMemberCache(string[] cubeNames,
parameterValue[] parameterValues, option[] options)
```


References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the refreshCubeMemberCache method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes to be refreshed.

If no names are provided, then all cubes configured for the query service receiving the request are refreshed.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

refreshCubeSecurity(cubeNames, parameterValues, options)

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
refreshCubeSecurity(java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply refreshCubeSecurity(string[] cubeNames,
parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the `refreshCubeSecurity` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the ROLAP cubes to be refreshed.

If no names are provided, then all cubes configured for the [queryService](#) receiving the request are refreshed.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

release(conversation)

Use this method to remove inactive requests from the service cache earlier than they would be removed automatically by the system. Removing inactive requests makes more resources available for other requests, which can improve performance.

Every asynchronous conversation reserves server-side resources to handle client requests. These reserved resources cannot be used for other purposes while the associated asynchronous conversation exists. Although asynchronous conversations are terminated after a period of inactivity, more effective use of server-side resources can be realized when clients call `asynch » release(conversation)` when they have no further use for an asynchronous conversation. This allows these reserved resources to be used for other purposes.

The `asynch » release(conversation)` method can be called whenever the conversation status is `complete`. If the status is `working` or `stillWorking`, clients can free up server-side resources by calling `asynch » cancel(conversation)`. If the conversation status is `conversationComplete`, the server-side components have already made all resources associated with the asynchronous conversation available for other purposes and no further action is required by the client to terminate the asynchronous conversation.

IBM Cognos Viewer is an example of an SDK client that uses `asynch » release(conversation)` to make the most effective use of server-side resources possible in your IBM Cognos installation.

Signatures

Java and Apache Axis

```
public void release(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation)
```

C# .NET

```
public void release(asynchRequest conversation)
```

Example: Using the release(conversation) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the `asynch » release(conversation)` method with the report service.

To see this code in context, view the Java sample `RenderReport/Render.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
// release the conversation to free resources.
connect.getReportService().release(response.getPrimaryRequest());
```

References

Part of the following method sets:

- [asynch](#)

Implemented by the following services:

The services that implement the release method, along with the associated SOAP actions, are listed in the following table.

Table 99. Services implementing the release method.

Service	SOAP Action
agentService	http://www.ibm.com/xmlns/prod/cognos/agentService/201404/.absolute
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.absolute
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/.absolute
dataIntegrationService	http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/.absolute
dataMovementService	http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/.absolute
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/.absolute
dimensionManagementService	http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/.absolute
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.absolute
idVizService	http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/.absolute
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.absolute
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/.absolute
jobService	http://www.ibm.com/xmlns/prod/cognos/jobService/201404/.absolute
metadataService	http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.absolute
migrationService	http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/.absolute
mobileService	http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/.absolute
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.absolute
planningAdministrationConsoleService	http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/.absolute

Table 99. Services implementing the release method. (continued)

Service	SOAP Action
planningRuntimeService	http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/.absolute
planningTaskService	http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/.absolute
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.absolute
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.absolute
relationalMetadataService	http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/.absolute
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.absolute
repositoryService	http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/.absolute
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/.absolute

powerPlayService information

This information is specific to the [powerPlayService](#) service.

Reserved.

reportService information

This information is specific to the [reportService](#) service.

Used when IBM Cognos Viewer is closed, this method removes the current report request from the report service cache. No other secondary requests can follow this action.

What's new

New in Version 10.1.1 – “Improving the utilization of system resources by using the [asynch » release\(conversation\) method](#)” on page 1850

This topic was updated to remind developers that use of this method can help to improve system resource utilization.

Related information:

- [“Understanding the Asynchronous Conversation” on page 78](#)

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus » asynchReply » primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus » asynchReply » primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus » asynchRequest](#)
is encoded as type `tns:asynchRequest`

Return values

None.

releaseEvent(eventID)

Use this method to release an event that was held.

Releasing an event that was held (status of [suspended](#)) queues the event for execution and changes the status to [pending](#).

For more information about event status, see the [bibus » runStatusEnum](#) enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the [canUseMonitorActivityTool](#) capability.
This allows administrators to release tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to release their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to release tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void releaseEvent(java.lang.String eventID)
```

C# .NET

```
public void releaseEvent(string eventID)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `releaseEvent` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
<code>eventManagementService</code>	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – “Administration Capabilities for Background Job Queue Maintenance” on page 1877

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » [releaseEvents\(eventIDs\)](#) method

Input parameters

Use the following parameters when calling this method.

eventID

Identifies the event.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

None.

releaseEvents(eventIDs)

Use this method to release a set of events that were held.

Releasing an event that was held (status of `suspended`) queues the event for execution and changes the status to `pending`.

For more information about event status, see the [bibus](#) » [runStatusEnum](#) enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to release tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to release their own tasks that are running in the background.

- The account of the caller must match the account credential used to schedule the task.
This allows users to release tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void releaseEvents(java.lang.String[] eventIDs)
```

C# .NET

```
public void releaseEvents(string[] eventIDs)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the releaseEvents method, along with the associated SOAP actions, are listed in the following table.

<i>Table 101. Services implementing the releaseEvents method.</i>	
Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance” on page 1877](#)

This method now requires the caller to have the [canUseMonitorActivityTool](#) value capability instead of the [canUseAdministrationPortal](#) value capability when a capability check is required.

Related information:

- [event](#) » [releaseEvent\(eventID\)](#) method

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

Return values

None.

render(conversation, parameterValues, options)

Use this method to render the report in a different format than that which is specified in the run options.

The request can specify the following run options:

- [cssURL](#)
- [outputFormat](#)
- [outputLocale](#) (only the first locale in the list is processed)
- [outputPageDefinition](#)
- [outputPageOrientation](#)
- [xslURL](#)

You can specify only one [outputFormat](#) option. If you specify more, a fault is returned.

This method may return the following values in the [bibus](#) » [asynchReply](#) » [status](#) property:

- [complete](#)
- [working](#)

If the run options specified require new queries to be run, and if those queries must prompt the user for new parameter values, this method returns a value of [prompting](#) in the [bibus](#) » [asynchDetailReportStatus](#) » [status](#) property. An instance of this class will be returned in the [bibus](#) » [asynchReply](#) » [details](#) property.

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply render(
    com.cognos.developer.schemas.bibus._3.AsynchRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply render(asynchRequest conversation,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the render(conversation, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [report](#) » [render\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample `RenderReport/Render.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
renderedResp =
    connect.getReportService().render(
        response.getPrimaryRequest(),
        new ParameterValue[] {},
        renderOpts);
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the render method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.high
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.high

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the [bibus](#) » [asynchReply](#) » [primaryRequest](#) property.

The value of this parameter should be obtained from the most recent instance of the [bibus](#) » [asynchReply](#) » [primaryRequest](#) property returned from a method for the same conversation.

This argument

- is of type [bibus](#) » [asynchRequest](#)
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

restartCubes(cubeNames, parameterValues, options)

Use this method to restart ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply restartCubes(  
    java.lang.String[] cubeNames,  
    com.cognos.developer.schemas.bibus._3.ParameterValue[]  
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
    options)
```

C# .NET

```
public asynchReply restartCubes(string[] cubeNames, parameterValue[]  
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the restartCubes method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus » userCapabilityEnum » canUseDataSourcesTool](#)
- [bibus » userCapabilityEnum » canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the ROLAP cubes to be restarted.

If no names are provided, then all cubes configured for the query service receiving the request are restarted.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the `bibus » asynchReplyStatusEnum` enumeration set.

Any data requested can be obtained by examining the `bibus » asynchReply » details` property.

This result

- is of type `bibus » asynchReply`
is encoded as type `tns:asynchReply`

retrieveCredential(namespace)

Allows a user who is authenticated in a namespace to retrieve the authentication credentials for that namespace. These credentials can then be used to connect to a datasource that uses the same

authentication mechanism as the namespace. This mechanism allows the user to avoid having to authenticate with the datasource.

The following limitations apply:

- A user can only retrieve their own credentials.
- The credentials are encrypted and can only be accessed with the correct cskey that can decrypt it.

Signatures

Java and Apache Axis

```
public XmlEncodedXML retrieveCredential(java.lang.String namespace)
```

C# .NET

```
public xmlEncodedXML retrieveCredential(string namespace)
```

References

Part of the following method sets:

- [standaloneCAM](#)

Implemented by the following services:

The services that implement the terminateSessions method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/

What's new

New in Version 10.2.2 – [Connecting to a datasource using namespace credentials](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

namespace

Specifies the namespace that the user is authenticated in.

This argument

- is of type string
- is encoded as type `xs:string`

Return values

This method returns the following values.

credential

The credential that will be passed to the datasource.

This result

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXML`

run(objectPath, parameterValues, options)

Use this method to run an object. This method initiates an asynchronous conversation represented by the return value of the method. The conversation state is used by the server to obtain the context in which to perform secondary tasks.

For more information, see [Chapter 6, “Running tasks,”](#) on page 65.

Use of this method requires:

- execute permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply run(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply run(searchPathSingleObject objectPath,
    parameterValue[] parameterValues, option[] options)
```

Example: Using the run(objectPath, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the `asynch » run(objectPath, parameterValues, options)` method with the report service.

To see this code in context, view the Java sample `ExecReports/RunReport.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
//Call run()
rsr =
    connect.getReportService(true, serverGroup).run(
        new SearchPathSingleObject(rSearchPath),
        paramValueArray,
        execReportRunOptions);
```

Example: Using the run(objectPath, parameterValues, options) Method with the Job Service in Java

The following Java code snippet demonstrates how to use the `asynch » run(objectPath, parameterValues, options)` method with the job service.

To see this code in context, view the Java sample `Submit/SubmitReport.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asynchReply =
    connect.getJobService().run(
```

```

        new SearchPathSingleObject(jobPath),
        reportParameters,
        runOpts);
...
for (i = 0; i < asynchReply.getDetails().length; i++)
{
    if (asynchReply.getDetails()[i] instanceof AsynchDetailEventID)
    {
        reportEventID = ((AsynchDetailEventID)asynchReply.getDetails()
[i]).getEventID();
    }
}
}

```

Example: Using the run(objectPath, parameterValues, options) Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the [asynch » run\(objectPath, parameterValues, options\)](#) method with the report service.

To see this code in context, view the C# sample ExecuteReport/ExecuteReport.cs. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```

executeReportResponse = cBIConnection.CBIRS.run(cmReportPath, arrParm,
arrRunOpts);

```

References

Part of the following method sets:

- [asynch](#)

Implemented by the following services:

The services that implement the run method, along with the associated SOAP actions, are listed in the following table.

<i>Table 105. Services implementing the run method.</i>	
Service	SOAP Action
agentService	http://www.ibm.com/xmlns/prod/cognos/agentService/201404/.session
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.session
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/.session
dataIntegrationService	http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/.session
dataMovementService	http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/.session
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/.session

Table 105. Services implementing the run method. (continued)

Service	SOAP Action
dimensionManagementService	http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/.session
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.session
idVizService	http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/.session
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.session
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/.session
jobService	http://www.ibm.com/xmlns/prod/cognos/jobService/201404/.session
metadataService	http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.session
migrationService	http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/.session
mobileService	http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/.session
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.session
planningAdministrationConsoleService	http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/.session
planningRuntimeService	http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/.session
planningTaskService	http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/.session
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.session
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.session
relationalMetadataService	http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/.session
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.session

Table 105. Services implementing the run method. (continued)

Service	SOAP Action
repositoryService	http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/.session
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/.session

agentService information

This information is specific to the [agentService](#) service.

Use of this method requires:

- appropriate permissions for any objects run as part of the agent

Related information:

[bibus » agentOptionEnum](#)

[bibus » rssOptionEnum](#)

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

The type of [bibus » asynchDetail](#) objects returned in the details property of the [bibus » asynchReply](#) object depends on the class of the object in the [objectPath](#) parameter of the initial request. The following table lists the [bibus » asynchDetail](#) objects that can be returned for each class.

Table 106. Class and asynchDetail object cross-reference

Class	Purpose	Details
bibus » analysis bibus » query bibus » report bibus » reportView	Run the object and retrieve output.	bibus » asynchDetailParameterValues bibus » asynchDetailPromptPage bibus » asynchDetailReportOutput bibus » asynchDetailReportStatus
bibus » drillPath	Retrieve information about a target, including translated parameter values, actions, options, and generated information such as the editSpecification option.	bibus » asynchDetailDrillThroughRequest bibus » asynchDetailParameterValues bibus » asynchDetailPromptPage

Not all objects listed are returned for every request.

- An [bibus » asynchDetailDrillThroughRequest](#) object is returned in response to running a [bibus » drillPath](#) object. It contains information about how to run the [target](#) referenced by the [bibus » drillPath](#) object.
- An [bibus » asynchDetailParameterValues](#) object is returned for any request that requires parameter values or was passed parameter values to execute.
- An [bibus » asynchDetailPromptPage](#) object is returned only if the request cannot be completed without additional information.
- An [bibus » asynchDetailReportStatus](#) object is returned if the output is a prompt page.

For report objects that have more than one output format or delivery option, only one output format and delivery option must be valid for the request to succeed. For reports that run interactively in which the

user waits for the output, you can only specify one [bibus » outputFormatEnum](#). If you specify more than one option, the request fails.

Tip: For [bibus » interactiveReport](#) objects, only [HTML](#) can be specified. Specifying any other output format will result in a fault. The resulting output is an MHTML document.

Use of this method requires:

- [traverse](#) permission for the package
- [traverse](#) and [execute](#) permission for the data sources
- [execute](#) permission for the data source connection
- [execute](#) and [traverse](#) permission for the data source connection if you are using a data source signon
- [read](#) or [execute](#) permission for the metadata

New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889

This method can now be used with a [bibus » drillPath](#) object as a replacement for the `determineDrillThroughTargetParameterValues(objectPaths, sourceContext, sourceContextValues, targetParameterAssignments, targetParameters, parameterValues, options)` method.

Related information:

- [bibus » asynchOptionEnum](#) enumeration set
- [bibus » deliveryOptionEnum](#) enumeration set
- [bibus » drillOptionEnum](#) enumeration set
- [bibus » drillThroughOptionEnum](#) enumeration set
- [bibus » pdfOptionEnum](#) enumeration set
- [bibus » runOptionEnum](#) enumeration set
- [bibus » validateOptionEnum](#) enumeration set
- [bibus » runOptionEnum » returnOutputWhenAvailable](#) value
- [bibus » specificationFormatEnum » interactiveReport](#) value
- [bibus » asynchDetailDrillThroughRequest](#) class
- [bibus » asynchDetailParameterValues](#) class
- [bibus » asynchDetailPromptPage](#) class
- [bibus » asynchDetailReportOutput](#) class
- [bibus » asynchDetailReportStatus](#) class

contentManagerService information

This information is specific to the [contentManagerService](#) service.

Use of this method requires:

- for [bibus » exportDeployment](#) objects
 - [read](#) permission for objects being deployed
- for [bibus » importDeployment](#) objects
 - [write](#) permission for the parent of each object being imported
 - [traverse](#) permission for all ancestors of each object being imported

Related information:

- [bibus » contentTaskOptionEnum](#) enumeration set
- [bibus » deploymentOptionEnum](#) enumeration set

dataIntegrationService information

This information is specific to the [dataIntegrationService](#) service.

Related information:

[bibus](#) » [dataIntegrationTaskOptionEnum](#) enumeration set

deliveryService information

This information is specific to the [deliveryService](#) service.

Related information:

[bibus](#) » [deliveryOptionEnum](#) enumeration set

[bibus](#) » [rssOptionEnum](#) enumeration set

dimensionManagementService information

This information is specific to the [dimensionManagementService](#) service.

This service does not support the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

jobService information

This information is specific to the [jobService](#) service.

Use of this method requires:

- appropriate permissions for any objects run as part of the job

metadataService information

This information is specific to the [metadataService](#) service.

This service does not support the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

monitorService information

This information is specific to the [monitorService](#) service.

To retry a task such as an agent or job that failed because of a database or network error

- Specify the task to retry in the [objectPath](#). input parameter.
- Specify the location of the [bibus](#) » [history](#) object for the failed run in a [restartHistoryLocation](#) option.
- Optionally, specify any steps to be skipped during the retry in a [skipTaskHistoryLocations](#) option.

The options and parameter values associated with the original request are used during a retry.

Use of this method requires

- Execute permission for the credential that was used for the original run of the task to be retried

New in Version 8.3 — “Task Retry” on page 1909

Information related to task retry was added to this method.

Related information:

[bibus](#) » [monitorOptionEnum](#) enumeration set

relationalMetadataService information

This information is specific to the [relationalMetadataService](#) service.

New in Version 10.1.0 — “Relational Metadata Service” on page 1858

The documentation has been updated to show that this service does not support this method.

saCAMService information

This information is specific to the [saCAMService](#) service.

This service does not support the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

Related information:

- [bibus](#) » [asynchOptionEnum](#)

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

runAt(startTime, objectPath, parameterValues, options)

Use this method to run a task at a later time.

Permissions and capability requirements are checked when the object is run, not when this method is called. If the credential used to call this method does not have the appropriate requirements, a [bibus » history](#) object is written with a status of [failed](#) for the operation.

Use of this method requires:

- the [canUseScheduling](#) capability
- execute permission for the object
- traverse permission for all ancestors of the object

Depending on the target object, you may also require:

- traverse permission for the package and its ancestors
- traverse and execute permission for the data sources
- execute permission for the data source connection
- execute and traverse permission for the data source connection if you are using a data source signon
- read or execute permission for the metadata

Signatures

Java and Apache Axis

```
public java.lang.String runAt(java.util.Calendar startTime,
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public string runAt(System.DateTime startTime, searchPathSingleObject
    objectPath, parameterValue[] parameterValues, option[] options)
```

Example: Using the runAt(startTime, objectPath, parameterValues, options) Method with the Event Management Service in Java

The following Java code snippet demonstrates how to use the [event » runAt\(startTime, objectPath, parameterValues, options\)](#) method with the event management service.

To see this code in context, view the Java sample `ExecReports/RunReport.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page [1419](#).

```
resultEventID =
    connect.getEventMgmtService().runAt(
        execTime,
        new SearchPathSingleObject(rSearchPath),
        reportParameterValues,
        reportRunOptions);
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the runAt method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

Input parameters

Use the following parameters when calling this method.

startTime

Specifies the time when execution will begin, in Coordinated Universal Time (UTC).

This argument

- is of type [dateTime](#)
is encoded as type `xs:dateTime`

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the identifier of the event.

This result

- is of type `string`
is encoded as type `xs:string`

runSpecification(specification, parameterValues, options)

Use this method to run a specification.

This method initiates an asynchronous conversation with a service. The return value of the method contains information about the conversation. This information can be used by an application to determine the status of the conversation, or which secondary tasks are allowed.

If the request takes longer than the timeout value, a status of `working` may be returned. You should then call `asynch » wait(conversation, parameterValues, options)` until the status is `complete` or `conversationComplete`, at which time the requested data will be returned.

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
runSpecification(
    com.cognos.developer.schemas.bibus._3.AsynchSpecification
    specification, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply runSpecification(asynchSpecification
specification, parameterValue[] parameterValues, option[] options)
```

Example: Using the runSpecification(specification, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the `asynch » runSpecification(specification, parameterValues, options)` method with the report service.

To see this code in context, view the Java sample `ReportSpec/ReportObject.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,” on page 1419](#).

```
runResponse = connection.getReportService().runSpecification(
    metadataSpec, new ParameterValue[] { }, options);
for (int i = 0; i < runResponse.getDetails().length && !foundMetadata;
i++)
{
    if (runResponse.getDetails()[i] instanceof AsynchDetailReportMetadata)
    {
        foundMetadata = true;
        reportMetadata = (AsynchDetailReportMetadata) runResponse.getDetails()
[i];
        sMetaData = reportMetadata.getMetadata().toString();
    }
}
```


References

Part of the following method sets:

- [asynch](#)

Implemented by the following services:

The services that implement the runSpecification method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
agentService	http://www.ibm.com/xmlns/prod/cognos/agentService/201404/.session
batchReportService	http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.session
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/.session
dataIntegrationService	http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/.session
dataMovementService	http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/.session
deliveryService	http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/.session
dimensionManagementService	http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/.session
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.session
idVizService	http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/.session
indexSearchService	http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.session
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/.session
jobService	http://www.ibm.com/xmlns/prod/cognos/jobService/201404/.session
metadataService	http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.session
migrationService	http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/.session

Table 108. Services implementing the runSpecification method. (continued)

Service	SOAP Action
mobileService	http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/.session
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.session
planningAdministrationConsoleService	http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/.session
planningRuntimeService	http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/.session
planningTaskService	http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/.session
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.session
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.session
relationalMetadataService	http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/.session
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.session
repositoryService	http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/.session
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/.session

agentService information

This information is specific to the [agentService](#) service.

This service does not support the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

The type of [bibus » asynchDetail](#) objects returned in the [details](#) property of the [bibus » asynchReply](#) object depends on the class of the object in the [specification](#) parameter of the initial request. The following table lists the [bibus » asynchDetail](#) objects that can be returned for each class.

Table 109. Class and *asynchDetail* object cross-reference

Class	Purpose	Details
bibus » reportServiceAnalysisSpecification bibus » reportServiceInteractiveReportSpecification bibus » reportServiceQuerySpecification bibus » reportServiceReportSpecification	Run the object and retrieve output.	bibus » asynchDetailParameterValues bibus » asynchDetailPromptPage bibus » asynchDetailReportOutput bibus » asynchDetailReportStatus
bibus » reportServiceDrillThroughSpecification	Retrieve information about a target, including translated parameter values, actions, options, and generated information such as the editSpecification option.	bibus » asynchDetailDrillThroughRequest bibus » asynchDetailParameterValues bibus » asynchDetailPromptPage
bibus » reportServiceMetadataSpecification	Retrieve metadata.	bibus » asynchDetailParameterValues bibus » asynchDetailPromptPage bibus » asynchDetailReportMetadata
bibus » metadataServiceLineageSpecification	Retrieve the report layout lineage information and <code>querySet</code> . Users must issue a lineage request against <code>metadataService</code> , for the returned <code>querySet</code> , to obtain further lineage information used to generate the report output.	bibus » asynchDetailMIMEAttachment

Note: For [bibus » reportServiceInteractiveReportSpecification](#) objects, only `HTML` value can be specified. Specifying any other output format will result in a fault. The resulting output is an MHTML document.

Not all objects listed are returned for every request.

- An [bibus » asynchDetailDrillThroughRequest](#) object is returned in response to running a [bibus » reportServiceDrillThroughSpecification](#). It contains information about how to run the target referenced by the specification.
- An [bibus » asynchDetailParameterValues](#) object is returned for any request that requires parameter values or was passed parameter values to execute.
- An [bibus » asynchDetailPromptPage](#) object is returned only if the request cannot be completed without additional information.
- An [bibus » asynchDetailReportStatus](#) object is returned if the output is a prompt page.

Use of this method requires:

- traverse permission for the package
- additional permissions depending on the specification type

For [bibus » reportServiceAnalysisSpecification](#), [bibus » reportServiceInteractiveReportSpecification](#), [bibus » reportServiceQuerySpecification](#), and [bibus » reportServiceReportSpecification](#) requests, the following permissions are required:

- execute permission for the metadata
- traverse and execute permission for the data sources
- execute permission for the data source connection

For [bibus » reportServiceMetadataSpecification](#) requests, the following permissions are required:

- read permission for the metadata
- traverse and execute permission for the data sources if the request queries native metadata
- execute permission for the data source connection if the request queries native metadata
- traverse permission for the data source connection if you are using a data source signon

For more information about metadata specifications, see the [Chapter 35, “Metadata schema reference,”](#) on page 1765.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseAdaptiveAnalytics	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseAnalysisStudio	bibus » reportServiceAnalysisSpecification , bibus » reportServiceMetadataSpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseControllerStudio	bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseEventStudio	bibus » reportServiceMetadataSpecification
—	bibus » userCapabilityEnum » canUseEventStudio	bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseMetricsManagerAdministration	bibus » reportServiceMetadataSpecification
Trusted	bibus » userCapabilityEnum » canUseMetricStudio	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUsePlanningContributor	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseQueryStudio	bibus » reportServiceMetadataSpecification , bibus » reportServiceQuerySpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseReportStudio	bibus » reportServiceInteractiveReportSpecification , bibus » reportServiceMetadataSpecification , or bibus » reportServiceReportSpecification

Table 110. Capability rules for the runSpecification method. (continued)

Specification	Capabilities	Class
—	bibus » userCapabilityEnum » canUseSpecifications	bibus » reportServiceAnalysisSpecification , bibus » reportServiceInteractiveReportSpecification , bibus » reportServiceMetadataSpecification , bibus » reportServiceQuerySpecification , or bibus » reportServiceReportSpecification
Trusted	—	bibus » reportServiceDrillThroughSpecification
Trusted	bibus » userCapabilityEnum » canUseLineage	bibus » metadataServiceLineageSpecification

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

The capability rules were updated to include the rules related to the use of the [bibus » reportServiceReportSpecification](#) class with the [userCapabilityEnum » canUseDashboardViewer](#) value user capability.

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This method can now be used with a [bibus » reportServiceInteractiveReportSpecification](#).

The capability rules were updated to include the rules related to the use of the [bibus » reportServiceInteractiveReportSpecification](#) class.

Related information:

- [bibus » asynchOptionEnum](#) enumeration set
- [bibus » deliveryOptionEnum](#) enumeration set
- [bibus » drillOptionEnum](#) enumeration set
- [bibus » drillThroughOptionEnum](#) enumeration set
- [bibus » pdfOptionEnum](#) enumeration set
- [bibus » runOptionEnum](#) enumeration set
- [bibus » validateOptionEnum](#) enumeration set
- [bibus » specificationFormatEnum » interactiveReport](#) value
- [bibus » asynchDetailDrillThroughRequest](#) class
- [bibus » asynchDetailParameterValues](#) class
- [bibus » asynchDetailPromptPage](#) class
- [bibus » asynchDetailReportOutput](#) class
- [bibus » asynchDetailReportStatus](#) class

contentManagerService information

This information is specific to the [contentManagerService](#) service.

This service does not support the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

dataIntegrationService information

This information is specific to the [dataIntegrationService](#) service.

This service does not support the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

dataMovementService information

This information is specific to the [dataMovementService](#) service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Capabilities	Class
bibus » userCapabilityEnum » canUseDataManager	bibus » dataMovementServiceSpecification

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

Related information:

[bibus » dataMovementServiceSpecification](#) class

[bibus » historyDetailDataMovementService](#) class

deliveryService information

This information is specific to the [deliveryService](#) service.

Use the [asynch » runSpecification\(specification, parameterValues, options\)](#) method with the [delivery](#) service to send email. Provide an empty [bibus » deliveryServiceSpecification](#) to satisfy the [specification](#) parameter. Use the [bibus » deliveryOption](#) family of classes to specify the options for the email.

Related information:

[bibus » deliveryOptionEnum](#) enumeration set

[bibus » rssOptionEnum](#) enumeration set

dimensionManagementService information

This information is specific to the [dimensionManagementService](#) service.

Use the [asynch » runSpecification\(specification, parameterValues, options\)](#) method with the [dimension management](#) service to send requests for IBM Cognos Business Viewpoint data and metadata. Provide a [bibus » dimensionManagementServiceSpecification](#) object for the [specification](#) parameter.

An [bibus » asynchDetailMIMEAttachment](#) object is returned in the [bibus » asynchReply » details](#) property when the asynchronous conversation status is [complete](#).

The requested information is always returned inline, in base64-encoded format.

Refer to the IBM Cognos Business Viewpoint documentation for more information.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Capabilities	Class
<code>bibus » userCapabilityEnum » canUseSpecifications</code>	<code>bibus » dimensionManagementServiceSpecification</code>

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 8.4 — “Dimension Management Service” on page 1896

The documentation has been updated to indicate that this service now supports this method.

eventManagementService information

This information is specific to the `eventManagementService` service.

Use the `asynch » runSpecification(specification, parameterValues, options)` method with the `bibus » eventManagementService` to retrieve information about scheduled future events. You must provide an `bibus » eventManagementServiceSpecification` object for the `specification` parameter.

An `bibus » asynchDetailEventRecord` object is returned in the `bibus » asynchReply » details` property for each event when the asynchronous conversation status is `complete`.

The number of objects returned may be constrained by the value specified in the `maximumObjects` option. If the `maximumObjects` option is specified, `paging » nextPage(conversation, parameterValues, options)` method can be called to get the next set of events.

Use of this method requires one of the following:

- The caller has the `canUseMonitorActivityTool` capability.

An advanced setting `ems.action.requires.permissions.check` can be enabled to force a permissions check. If enabled, a caller with this capability also requires one of the following:

- The account of the caller must match the account credential used to schedule the event.
- The caller must have traverse and execute permissions on the target object.
- The account of the caller must match the account credential used to schedule the event. In this case, the caller must also have:
 - traverse permission for all ancestors of the target object (i.e., parent of the schedule) associated with the event.
 - read permission for the target object associated with the event

This allows users to manage events that they previously scheduled.

The caller must also have read permission on the target object to view detailed information on the task, such as a task's name.

New in Version 8.3 — “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Related information:

`paging` method set

`bibus » runStatusEnum` enumeration set

jobService information

This information is specific to the [jobService](#) service.

This service does not support the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method.

metadataService information

This information is specific to the [metadataService](#) service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseLineage	bibus » metadataServiceLineageSpecification
—	bibus » userCapabilityEnum » canUseSpecifications	bibus » metadataServiceLineageSpecification

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 8.4 — “Lineage Metadata” on page 1891

The documentation has been updated to show that this service now supports this method.

Related information:

[bibus](#) » [asynchDetailMIMEAttachment](#) class

monitorService information

This information is specific to the [monitorService](#) service.

Use the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method with the [bibus](#) » [monitorService](#) to retrieve information about currently running events. You must provide a [bibus](#) » [monitorServiceSpecification](#) for the [specification](#) parameter.

An [bibus](#) » [asynchDetailEventRecord](#) object is returned in the [bibus](#) » [asynchReply](#) » [details](#) property for each task when the asynchronous conversation status is [complete](#).

The number of objects returned may be constrained by the value specified in the [maximumObjects](#) option. If the [maximumObjects](#) option is specified, [paging](#) » [nextPage\(conversation, parameterValues, options\)](#) method can be called to get the next set of events.

Use of this method requires one of the following:

- The caller has the [canUseMonitorActivityTool](#) capability.

The account of the caller must match the account credential used to schedule the event. In this case, the caller must also have:

- traverse permission for all ancestors of the target object (i.e., parent of the schedule) associated with the event.

This allows users to manage events that they previously scheduled.

- The passport ID of the caller must match the account associated with the task run.

This allows users to manage their own events that are running in the background.

The caller must also have read permission on the target object to view detailed information on the task, such as a task's name.

New in Version 8.3 – “Schedule Management” on page 1907

The documentation has been updated to indicate that this service now supports this method.

Related information:

[bibus](#) » [monitorOptionEnum](#) enumeration set

[bibus](#) » [runStatusEnum](#) enumeration set

powerPlayService information

This information is specific to the [powerPlayService](#) service.

Reserved.

queryService information

This information is specific to the [queryService](#) service.

This service does not support the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method.

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

The documentation has been updated to show that this service does not support this method.

relationalMetadataService information

This information is specific to the [relationalMetadataService](#) service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

<i>Table 114. Capability rules for the runSpecification method.</i>		
Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canImportRelationalMetadata	bibus » relationalMetadataServiceSpecification

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

The documentation has been updated to show that this service now supports this method.

Related information:

[bibus](#) » [asynchDetailMIMEAttachment](#) class

saCAMService information

This information is specific to the [saCAMService](#) service.

This service does not support the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method.

Related information:

- [bibus](#) » [asynchOptionEnum](#)

Input parameters

Use the following parameters when calling this method.

specification

Defines the specification associated with the request.

This argument

- is of type [bibus](#) » [asynchSpecification](#)
is encoded as type `tns:asynchSpecification`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type `bibus » asynchReply`
is encoded as type `tns:asynchReply`

scheduleEvent(eventID)

Use this method to reschedule an event that was canceled before being queued for execution.

An event that was canceled (status of `cancelled`) and is then rescheduled is put in the `scheduled` state.

For more information about event status, see `bibus » runStatusEnum` enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for all ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to reschedule tasks on behalf of other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to reschedule their own tasks that are running in the background.
 - The account of the caller must match the account credential used to schedule the task.
This allows users to reschedule tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void scheduleEvent(java.lang.String eventID)
```

C# .NET

```
public void scheduleEvent(string eventID)
```

References

Part of the following method sets:

- `event`

Implemented by the following services:

The services that implement the `scheduleEvent` method, along with the associated SOAP actions, are listed in the following table.

<i>Table 115. Services implementing the scheduleEvent method.</i>	
Service	SOAP Action
<code>eventManagementService</code>	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – “Administration Capabilities for Background Job Queue Maintenance” on page 1877

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » [scheduleEvents\(eventIDs\)](#) method

Input parameters

Use the following parameters when calling this method.

eventID

Identifies the event.

This argument

- is of type `string`
is encoded as type `xs:string`

Return values

None.

scheduleEvents(eventIDs)

Use this method to reschedule events that were canceled before being queued for execution.

An event that was previously canceled (status of `cancelled`) and is then rescheduled is put in the `scheduled` state.

For more information about event status, see the [bibus](#) » [runStatusEnum](#) enumeration set and “[Running Jobs](#)” on page 76.

Use of this method requires:

- traverse permission for the ancestors of the target object associated with the event
- one of the following conditions to be met:
 - The caller has the `canUseMonitorActivityTool` capability.
This allows administrators to reschedule tasks initiated by other users.
 - The passport ID of the caller must match the account associated with the task.
This allows users to reschedule their own tasks that are running in the background.
 - The account of the caller must match the account credential used to originally schedule the task.
This allows users to reschedule tasks that they previously scheduled.

Signatures

Java and Apache Axis

```
public void scheduleEvents(java.lang.String[] eventIDs)
```

C# .NET

```
public void scheduleEvents(string[] eventIDs)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `scheduleEvents` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance” on page 1877](#)

This method now requires the caller to have the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability when a capability check is required.

Related information:

- [event](#) » [scheduleEvent\(eventID\)](#) method

Input parameters

Use the following parameters when calling this method.

eventIDs

Identifies the events.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

Return values

None.

[selectRoles\(roles\)](#)

Use this method to select roles for the current session. If the user is authenticated by a security provider, the action is successful and the [bibus](#) » [biBusHeader](#) passport, located in the [bibus](#) » [CAM](#) » [CAMPassport](#) property, is updated.

For more information, refer to [“Using the selectRoles Method has no Effect” on page 117](#).

Signatures

Java and Apache Axis

```
public void selectRoles(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject[] roles)
```

C# .NET

```
public void selectRoles(searchPathSingleObject[] roles)
```

References

Part of the following method sets:

- [authentication](#)

Implemented by the following services:

The services that implement the selectRoles method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/ contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

roles

Specifies the set of search paths for the roles that will be assigned to the current session.

This argument

- is an array of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

Return values

None.

startCubes(cubeNames, parameterValues, options)

Use this method to start ROLAP cubes on a particular [queryService](#).

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply startCubes(  
    java.lang.String[] cubeNames,  
    com.cognos.developer.schemas.bibus._3.ParameterValue[]  
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]  
    options)
```

C# .NET

```
public asyncReply startCubes(string[] cubeNames, parameterValue[]  
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the startCubes method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the cubes to be started.

If no names are provided, then all cubes configured for the query service receiving the request are started.

This argument

- is an array of type `string`
is encoded as type `tns:stringArray`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

startService(servicePath)

Use this method to signal the dispatcher to start a service.

If the service is already running, no action is taken and the operation is successful.

If the service is disabled, the service is not started and a fault is returned.

Use of this method requires:

- write permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public void startService(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    servicePath)
```

C# .NET

```
public void startService(searchPathSingleObject servicePath)
```

Example: Using the startService (servicePath) Method with the Dispatcher Service in Java

The following Java code snippet demonstrates how to use the [dispatcher » startService\(servicePath\)](#) method with the dispatcher service.

To see this code in context, view the Java sample `DispatcherControl\Dispatcher.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
SearchPathSingleObject myDispSearchPath = new SearchPathSingleObject(  

```



```

        dispSearchPathURL);
    try {
        Con.getDispatcherService().startService(myDispSearchPath);
    }
}

```

References

Part of the following method sets:

- [dispatcher](#)

Implemented by the following services:

The services that implement the startService method, along with the associated SOAP actions, are listed in the following table.

Table 119. Services implementing the startService method.

Service	SOAP Action
dispatcher	http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201404/

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

Input parameters

Use the following parameters when calling this method.

servicePath

Identifies the service to start.

Specify the location of the service that you want to start. For example, configuration/dispatcher[@name='Test Server']/reportService[@name='Report Service']

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type tns:searchPathSingleObject

Return values

None.

stopCubes(cubeNames, parameterValues, options)

Use this method to stop ROLAP cubes on a particular [queryService](#).

Stopping a cube restricts further requests to access a cube and waits until all current requests have finished accessing the cube.

Signatures

Java and Apache Axis

```

public com.cognos.developer.schemas.bibus._3.AsyncReply stopCubes(

```

```
java.lang.String[] cubeNames,
com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply stopCubes(string[] cubeNames, parameterValue[]
parameterValues, option[] options)
```

References

Part of the following method sets:

- [rolapCubeAdministration](#)

Implemented by the following services:

The services that implement the stopCubes method, along with the associated SOAP actions, are listed in the following table.

<i>Table 120. Services implementing the stopCubes method.</i>	
Service	SOAP Action
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#)
- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

cubeNames

Specifies the names of the ROLAP cubes to be stopped.

If no names are provided, then all cubes configured for the query service receiving the request are stopped.

This argument

- is an array of type string
is encoded as type tns:stringArray

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

stopService(servicePath, immediately)

Use this method to signal the dispatcher to stop a service.

If the service is already stopped or disabled, no action is taken and the operation is successful.

Use of this method requires:

- write permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public void stopService(  
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject  
    servicePath, boolean immediately)
```

C# .NET

```
public void stopService(searchPathSingleObject servicePath, bool  
    immediately)
```

Example: Using the stopService(servicePath, immediately) Method with the Dispatcher Service in Java

The following Java code snippet demonstrates how to use the [dispatcher » stopService\(servicePath, immediately\)](#) method with the dispatcher service.

To see this code in context, view the Java sample DispatcherControl\Dispatcher.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
SearchPathSingleObject myDispSearchPath = new SearchPathSingleObject(
    dispSearchPathURL);

try {
    Con.getDispatcherService().stopService(myDispSearchPath, true);
}
```

References

Part of the following method sets:

- [dispatcher](#)

Implemented by the following services:

The services that implement the stopService method, along with the associated SOAP actions, are listed in the following table.

<i>Table 121. Services implementing the stopService method.</i>	
Service	SOAP Action
dispatcher	http://www.ibm.com/xmlns/prod/cognos/dispatcherService/201404/

Capability Rules

Use of this method requires one of the following sets of capabilities:

- [bibus](#) » [userCapabilityEnum](#) » [canUseServerAdministrationTool](#)

Input parameters

Use the following parameters when calling this method.

servicePath

Identifies the service to stop.

Specify the location of the service that you want to stop. For example, configuration/dispatcher[@name='Test Server']/reportService[@name='Report Service']

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type tns:searchPathSingleObject

immediately

Specifies whether the service must be stopped immediately.

If set to true, the service is stopped immediately. This causes running requests to fail and all requests to be removed from the queue. The service will attempt to generate a reasonable error page or Simple Object Access Protocol (SOAP) fault for each affected request.

If set to false, the service is stopped after all running and queued requests are completed.

This argument

- is of type boolean

is encoded as type `xs:boolean`

Return values

None.

terminateSessions(search)

Use this method to terminate existing user sessions identified by the search path argument. This method can only be performed by a member of the System Administrators role.

Signatures

Java and Apache Axis

```
public int terminateSessions(SearchPathMultipleObject search)
```

C# .NET

```
public int terminateSessions(SearchPathMultipleObject search)
```

References

Part of the following method sets:

- [standaloneCAM](#)

Implemented by the following services:

The services that implement the terminateSessions method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
saCAMService	<code>http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/</code>

Input parameters

Use the following parameters when calling this method.

search

Specifies the search path for the sessions to be terminated.

This argument

- is of type [bibus » searchPathMultipleObject](#)
is encoded as type `tns:searchPathMultipleObject`

Return values

This method returns the following values.

count

Returns the number of sessions that were terminated.

This result

- is of type `int`
is encoded as type `xs:int`

testDataSourceConnection(connectionString, credentials)

Use this method to test whether the data source connection string can be used to attach to the data source. If it cannot attach, information about the failure is returned as a fault. No secondary requests can follow this method.

To test your connection on a specific server, set the `bibus » conversationContext » nodeID` property to the `dispatcherID` of the dispatcher you want to test.

The `bibus » conversationContext » nodeID` property is contained by the `bibus » tracking » conversationContext` property which is part of the `bibus » biBusHeader`.

If the `bibus » conversationContext » nodeID` property is not set, the request is forwarded to any available dispatcher.

On success, this method returns an empty response.

Java and Apache Axis

```
public void testDataSourceConnection(java.lang.String
connectionString, com.cognos.developer.schemas.bibus._3.XmlEncodedXML
credentials)
```

C# .NET

```
public void testDataSourceConnection(string connectionString,
xmlEncodedXML credentials)
```

References

Part of the following method sets:

- [dataSource](#)

Implemented by the following services:

The services that implement the `testDataSourceConnection` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.server</code>
metadataService	<code>http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.server</code>
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.server</code>

Capability Rules

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 124. Capability rules for the `testDataSourceConnection` method.

Specification	Capabilities	Class
Trusted	<code>bibus » userCapabilityEnum » canUseAnalysisStudio</code>	—
—	<code>bibus » userCapabilityEnum » canUseDataSourcesTool</code>	—
Trusted	<code>bibus » userCapabilityEnum » canUseEventStudio</code>	—
Trusted	<code>bibus » userCapabilityEnum » canUseMetricStudio</code>	—
Trusted	<code>bibus » userCapabilityEnum » canUseQueryStudio</code>	—
Trusted	<code>bibus » userCapabilityEnum » canUseReportStudio</code>	—
Trusted	<code>bibus » userCapabilityEnum » canUsePowerPlay</code>	—

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch`).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch/ext`).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

batchReportService and reportService information

This information is specific to the `batchReportService` and `reportService` services.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This method was added. It replaces the `report » testDataSourceConnection(connectionString, credentials)` – obsolete method.

metadataService information

This information is specific to the `metadataService` service.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

The documentation has been updated to show that this service now supports this method.

queryService information

This information is specific to the `queryService` service.

New in Version 10.1.0 – “Query Modes” on page 1865

The documentation has been updated to show that this service now supports this method.

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

The capability rules were updated to include the rules related to the [canUsePowerPlay](#) capability.

Input parameters

Use the following parameters when calling this method.

connectionString

Specifies the connection string.

This argument

- is of type `string`
is encoded as type `xs:string`

credentials

Specifies the credentials that are used to authenticate the user.

This argument

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXML`

Return values

None.

testDataSourceConnection(connectionString, credentials) – obsolete

What's new

New in Version 8.3 – “Capabilities Refinements” on page 1918

The capability rules were updated to reflect the introduction of the [canUseDataSourcesTool](#) capability to replace the obsolete [canUseDirectoryTool](#) capability.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This method is obsolete and was removed. Use the [dataSource » testDataSourceConnection\(connectionString, credentials\)](#) method instead.

Input parameters

Use the following parameters when calling this method.

connectionString

This argument

- is of type `string`
is not represented in the WSDL document

credentials

This argument

- is of type [bibus » xmlEncodedXML](#)
is not represented in the WSDL document

Return values

None.

testDataSourceConnectionWithInfo(connectionString, credentials)

Use this method to test a JDBC data source connection string and return information about the data source. The information returned is the information that is displayed when you click **Succeeded** after testing a data source connection in IBM Cognos Administration.

To test your connection on a specific server, set the [bibus » conversationContext » nodeID](#) property to the [dispatcherID](#) of the dispatcher you want to test.

The [bibus » conversationContext » nodeID](#) property is contained by the [bibus » tracking » conversationContext](#) property which is part of the [bibus » biBusHeader](#).

If the [bibus » conversationContext » nodeID](#) property is not set, the request is forwarded to any available dispatcher.

On success, this method returns an empty response.

Java and Apache Axis

```
public java.lang.String[] testDataSourceConnectionWithInfo(java.lang.String
connectionString, com.cognos.developer.schemas.bibus._3.XmlEncodedXML
credentials)
```

C# .NET

```
public string[] testDataSourceConnectionWithInfo(string connectionString,
xmlEncodedXML credentials)
```

References

Part of the following method sets:

- [dataSource](#)

Implemented by the following service:

The services that implement the `testDataSourceConnectionWithInfo` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
queryService	<code>http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.server</code>

What's new

New in Version 10.2.2 – Data source connection information

This method was added.

Input parameters

Use the following parameters when calling this method.

connectionString

Specifies the connection string.

This argument

- is of type `string`
is encoded as type `xs:string`

credentials

Specifies the credentials that are used to authenticate the user.

This argument

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXML`

Return values

result

Returns an array of messages that provide information about the data source connection.

This argument

- is an array of type `string`
is encoded as type `xs:stringArray`

trigger(triggerName)

Use this method to initiate a trigger-based schedule in response to an external occurrence.

For example, you may want to ensure that reports are run every Wednesday after the reporting database has been backed up. After the backup is completed, a script using this method triggers the report to run.

To schedule execution of an object based on a trigger, do the following:

1. Create an instance of the `bibus » schedule` class for the object.
2. Set the `type` property to `trigger`.
3. Set the `triggerName` property to the name of the external occurrence.
4. Save the schedule.

The event management service:

- will return a fault if the `triggerName` sent in on a `trigger` request is null or zero length
- executes only reports that the user can see in Content Manager
- asks Content Manager for a list of schedules that have a matching `triggerName` and are active
- returns the number of schedules that were activated by this method
- runs the task associated with the `schedule` under the user specified by the `bibus » schedule » credential` property

Use of this method requires:

- read and traverse permission for the `bibus` » `schedule` object
- traverse permission for all ancestors of the `bibus` » `schedule` object
- the `canUseMonitorActivityTool` capability

Signatures

Java and Apache Axis

```
public int trigger(java.lang.String triggerName)
```

C# .NET

```
public int trigger(string triggerName)
```

Example: Using the `trigger(triggerName)` Method with the Event Management Service in Java

The following Java code snippet demonstrates how to use the `event` » `trigger(triggerName)` method with the event management service.

To see this code in context, view the Java sample `EventTrigger/Trigger.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
return eventService.trigger(triggerName);
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the `trigger` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/</code>

What's new

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance”](#) on page 1877

This method now checks for the `canUseMonitorActivityTool` value capability instead of the `canUseAdministrationPortal` value capability.

Input parameters

Use the following parameters when calling this method.

triggerName

Specifies the name of the external occurrence. Tasks associated with schedules that have a matching value for the `bibus » schedule » triggerName` property are queued for execution.

This parameter can also be substituted with a string similar to a command URI:

```
<trigger-name>?f=<folder-search-path>&f=<folder-search-path>& . . .  
&f=<folder-search-path>
```

For example:

```
"TDAILY?f=/content/folder[@name='Customer A']&f=/content/folder[@name='Customer B']"
```

The scheduler interprets the arguments from left to right, obtains the list of runnable objects associated with each folder and `triggerName`, and queues the objects for execution.

An SDK application cannot control the order of execution within a folder, only the order that each folder is executed in for a particular trigger name.

Each trigger requires its own SDK call, but each `triggerName` string may contain one or more folders.

For example, an SDK application can call this method with a partial list of folders when it detects that the data refresh is late, and submit additional folders for execution as the appropriate content is delivered. The SDK application would have the option to:

- Call the trigger method without a list of folder paths.
 - All schedules associated with this trigger would be executed immediately.
- Call the trigger with a prioritized list of folder paths using the trigger execution URI string.
 - Only the objects residing under the folder list would be executed in the specified order. The application must ensure that all folders are specified using single or multiple calls to the method.

There is no context kept by the call. For example, if a call is made with a list of selected folders and a second call only specifies a trigger by name, all runnable objects associated with the trigger are executed, even those that were executed in the previous prioritized method call.

Do not use the question mark character "?" unless you are specifying a list of folder search path strings. Doing so may cause unpredictable results as an attempt will be made to interpret anything after the question mark as a search path.

This argument

- is of type `string`
 - is encoded as type `xs:string`

Return values

This method returns the following values.

result

Returns the number of schedules that were activated.

This result

- is of type `int`
 - is encoded as type `xs:int`

update(objects, options)

Use this method to modify existing objects in the content store.

By default, this method returns the [searchPath](#) property for each selected object. An error is returned if no objects are selected.

If optimistic concurrency control is used, the [bibus](#) » [baseClass](#) » [version](#) property is compared to the [version](#) of the affected objects in the content store. These properties are not updated. For more information, see [“Concurrency Control”](#) on page 55.

Use of this method requires:

- write permission for the object
- traverse permission for all ancestors of the object
- read permission for the updated object to retrieve property values
- `setPolicy` permission to read or write to the [bibus](#) » [baseClass](#) » [policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.BaseClass[] update(
    com.cognos.developer.schemas.bibus._3.BaseClass[] objects,
    com.cognos.developer.schemas.bibus._3.UpdateOptions options)
```

C# .NET

```
public baseClass[] update(baseClass[] objects, updateOptions options)
```

Example: Using the `update(objects, options)` Method with the Content Manager Service in Java

The following Java code snippet demonstrates how to use the [content](#) » [update\(objects, options\)](#) method with the Content Manager service.

To see this code in context, view the Java sample `HandlersCS/CSHandlers.java`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
return connection.getCMSService().update(bc, new UpdateOptions());
```

Example: Using the `update(objects, options)` Method with the Content Manager Service in C# .NET

The following C# code snippet demonstrates how to use the [content](#) » [update\(objects, options\)](#) method with the Content Manager service.

To see this code in context, view the C# sample `PrintReport/PrintReport.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
bcNewPrinterName = connection.CBICMS.update(bcAddPrinter, updateOpts);
```

References

Part of the following method sets:

- [content](#)

Implemented by the following services:

The services that implement the update method, along with the associated SOAP actions, are listed in the following table.

Table 127. Services implementing the update method.

Service	SOAP Action
contentManagerService	http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/

Input parameters

Use the following parameters when calling this method.

objects

Specifies the objects and properties to be updated.

The following conditions should be observed:

Table 128. objects conditions and results

Condition	Result
Do not include a property for an object	The property is not modified
Include a property for an object but do not specify a value	The value of that property is deleted
The value of an acquired property is deleted from an object, such as the policies property	The value will be acquired from an ancestor of the object
The specified value of a property contains a reference to a non-existent object	A fault is generated and the operation fails. For information about changing the default behavior, see the bibus » updateOptions » ignoreInvalidObjectReference property.

The [bibus » baseClass » searchPath](#) property of each object is used to select the object.

This argument

- is an array of type [bibus » baseClass](#)
is encoded as type `tns:baseClassArray`

options

Specifies additional update options for the method.

This argument

- is of type [bibus » updateOptions](#)
is encoded as type `tns:updateOptions`

Return values

This method returns the following values.

result

Returns information about the objects that were updated. The information that is returned is determined by the value of the [bibus » updateOptions » returnProperties](#) property.

This result

- is an array of type [bibus » baseClass](#)

is encoded as type `tns:baseClassArray`

update(object, options)

Use this method to modify an existing `bibus » authoredReport` object in the content store.

This method performs a number of activities before it updates the specified object in the content store by calling the `content » update(objects, options)` method:

- The report specification is validated. If a query is being added, it is converted to a report specification and then validated. If the specification is not valid, this method will fail and an appropriate fault will be returned.
- For `bibus » analysis` objects, the following properties are set to reflect the corresponding values in the report specification:
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » metadataModelPackage` property
- For `bibus » interactiveReport` objects, the following properties are set to reflect the corresponding values in the report specification:
 - `bibus » authoredReport » canBurst` property
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » metadataModelPackage` property
 - `bibus » authoredReport » paths` property
- For `bibus » query` objects, the following properties are set to reflect the corresponding values in the report specification:
 - `bibus » baseReport » executionPageDefinition` property
 - `bibus » baseReport » executionPageOrientation` property
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » metadataModelPackage` property
- For `bibus » report` objects, the following properties are set to reflect the corresponding values in the report specification:
 - `bibus » authoredReport » canBurst` property
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » metadataModelPackage` property
 - `bibus » authoredReport » paths` property
- For `bibus » reportTemplate` objects, the following properties are set to reflect the corresponding values in the report specification:
 - `bibus » authoredReport » canBurst` property
 - `bibus » authoredReport » metadataModel` property
 - `bibus » authoredReport » metadataModelPackage` property
 - `bibus » authoredReport » paths` property

Any data stored in the `bibus » reportCache` contained by the `bibus » baseReport` object is cleared.

Use of this method requires:

- the `canUseReportStudio`, `canUseQueryStudio`, `canUseEventStudio`, `canUseAnalysisStudio`, or the `canUseCognosInsight` capability
- write permission for the object
- traverse permission for all ancestors of the object
- read permission for the updated object to retrieve property values

- setPolicy permission to read or write to the [bibus](#) » [baseClass](#) » [policies](#) property

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AuthoredReport update(
    com.cognos.developer.schemas.bibus._3.AuthoredReport object,
    com.cognos.developer.schemas.bibus._3.UpdateOptions options)
```

C# .NET

```
public authoredReport update(authoredReport @object, updateOptions
options)
```

References

Part of the following method sets:

- [report](#)

Implemented by the following services:

The services that implement the update method, along with the associated SOAP actions, are listed in the following table.

<i>Table 129. Services implementing the update method.</i>	
Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/

Capability Rules

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

<i>Table 130. Capability rules for the update method.</i>		
Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseAnalysisStudio	bibus » analysis
Trusted	bibus » userCapabilityEnum » canUseEventStudio	bibus » report or bibus » reportTemplate
Trusted	bibus » userCapabilityEnum » canUseQueryStudio and bibus » userCapabilityEnum » canUseQueryStudioFileManagement	bibus » query
—	bibus » userCapabilityEnum » canUseCognosInsight	bibus » report , bibus » reportTemplate , or bibus » interactiveReport

Table 130. Capability rules for the update method. (continued)

Specification	Capabilities	Class
—	bibus » userCapabilityEnum » canUseReportStudio and bibus » userCapabilityEnum » canUseReportStudioFileManagement	bibus » report , bibus » reportTemplate , or bibus » interactiveReport

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (http://server_name:9300/p2pd/servlet/dispatch/ext).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

What's new

New in Version 10.1.0 — “[IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace](#)” on page 1858

The capability rules were updated to include the rules related to the use of the [bibus » report](#) class with the [userCapabilityEnum » canUseDashboardViewer](#) value user capability.

New in Version 10.1.0 — “[IBM Cognos Active Report](#)” on page 1869

This method can now be used with a [bibus » interactiveReport](#).

The capability rules were updated to include the rules related to the use of the [bibus » interactiveReport](#) class.

Input parameters

Use the following parameters when calling this method.

object

Specifies the object to be updated.

The [bibus » baseClass » searchPath](#) property is used to select the object.

This argument

- is of type [bibus » authoredReport](#)
is encoded as type `tns:authoredReport`

options

Specifies additional options for the method.

This argument

- is of type [bibus » updateOptions](#)
is encoded as type `tns:updateOptions`

Return values

This method returns the following values.

result

Returns information about the object that was updated. The information that is returned is determined by the value of the `bibus » updateOptions » returnProperties` property.

This result

- is of type `bibus » authoredReport`
is encoded as type `tns:authoredReport`

updateDrillPath(object, options)

Use this method to modify an existing `bibus » drillPath` object in the content store.

This method performs a number of activities before it updates the specified object in the content store by calling the `content » update(objects, options)` method:

- The drill-through specification is validated. If the specification is not valid, this method fails and returns an appropriate error.
- The following properties are set to reflect the corresponding values in the drill-through specification:
 - `bibus » drillPath » deploymentReferences` property

Use of this method requires:

- write permission for the object
- traverse permission for all ancestors of the object
- read permission for the updated object to retrieve property values
- `setPolicy` permission to read or write to the `bibus » baseClass » policies` property

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.DrillPath
updateDrillPath(com.cognos.developer.schemas.bibus._3.DrillPath
object, com.cognos.developer.schemas.bibus._3.UpdateOptions options)
```

C# .NET

```
public drillPath updateDrillPath(drillPath @object, updateOptions
options)
```

References

Part of the following method sets:

- `drillThrough`

Implemented by the following services:

The services that implement the `updateDrillPath` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
<code>batchReportService</code>	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/</code>

Table 131. Services implementing the updateDrillPath method. (continued)

Service	SOAP Action
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/

What's new

New in Version 8.4 – [“Supporting New Drill-through Targets” on page 1890](#)

This method was added.

Input parameters

Use the following parameters when calling this method.

object

Specifies the object to be updated.

The [bibus](#) » [baseClass](#) » [searchPath](#) property is used to select the object.

This argument

- is of type [bibus](#) » [drillPath](#)
is encoded as type `tns:drillPath`

options

Specifies additional options for the method.

This argument

- is of type [bibus](#) » [updateOptions](#)
is encoded as type `tns:updateOptions`

Return values

This method returns the following values.

result

Returns information about the object that was updated. The information that is returned is determined by the value of the [bibus](#) » [updateOptions](#) » [returnProperties](#) property.

This result

- is of type [bibus](#) » [drillPath](#)
is encoded as type `tns:drillPath`

updateEvents(events)

Use this method to update future scheduled events. A recurring schedule generates multiple events.

Use of this method requires:

- `traverse` permission for the ancestors of the target object associated with the event.
- That the caller has the [canUseSchedulingPriority](#) capability.

This allows them to change the task priority.

- That the caller meet one of the following conditions:
 - The caller has the [canUseMonitorActivityTool](#) capability.

- This allows administrators to prioritize scheduled tasks initiated by other users.
- The caller's passport ID matches the account associated with the task.
- This allows users to prioritize their own scheduled tasks.
- The caller's account matches the account credential used to schedule the task.
- This allows users to prioritize previously scheduled tasks.

Signatures

Java and Apache Axis

```
public void updateEvents(
    com.cognos.developer.schemas.bibus._3.EventRecord[] events)
```

C# .NET

```
public void updateEvents(eventRecord[] events)
```

References

Part of the following method sets:

- [event](#)

Implemented by the following services:

The services that implement the updateEvents method, along with the associated SOAP actions, are listed in the following table.

<i>Table 132. Services implementing the updateEvents method.</i>	
Service	SOAP Action
eventManagementService	http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/

What's new

New in Version 8.3 – [“Schedule Priority” on page 1912](#)

This method was added.

New in Version 10.1.0 – [“Administration Capabilities for Background Job Queue Maintenance” on page 1877](#)

This method now requires the caller to have the [canUseMonitorActivityTool](#) value capability instead of the [canUseAdministrationPortal](#) value capability when a capability check is required.

Input parameters

Use the following parameters when calling this method.

events

Specifies the requested events that the service will update.

This argument

- is an array of type [bibus » eventRecord](#)
is encoded as type `tns:eventRecordArray`

Return values

None.

updateMetadata(request)

Use this method to send `mdprovider` requests to the metadata service to update or publish the metadata in an unpublished model. Send `generic` requests to open, save, and close the model. Send `action` requests to update the model or to publish a package to the content store.

Action requests include action elements like the ones in the action logs that are created when you make changes through Framework Manager. You can use the actions in these action logs as a basis for building your own.

Updating an Unpublished Model

1. Open the model:
 - If the model does not already exist, send a `createModel` request.
 - If the model already exists, send an `openModel` request.
2. Send action requests. Each action request can include only one transaction, but the transaction may include several actions.
3. Send a `saveModel` request.
4. When you are finished modifying the model, send a `closeModel` request.

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.XmlEncodedXML
updateMetadata(com.cognos.developer.schemas.bibus._3.XmlEncodedXML
request)
```

C# .NET

```
public xmlEncodedXML updateMetadata(xmlEncodedXML request)
```

References

Part of the following method sets:

- [metadata](#)

Implemented by the following services:

The services that implement the `updateMetadata` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
metadataService	<code>http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/</code>

Input parameters

Use the following parameters when calling this method.

request

Specifies the `mdprovider` request. This can be an action request or a generic request.

For more information about the structure of these requests, see the Metadata Provider (Wrapper) Reference in the *IBM Cognos Framework Manager Developer Guide*.

This argument

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

Return values

This method returns the following values.

result

Returns the output for the requested action or actions.

This result

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

validate(objectPath, parameterValues, options) – obsolete

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus » searchPathSingleObject](#)
is not represented in the WSDL document

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » parameterValue](#)
is not represented in the WSDL document

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus » option](#)
is not represented in the WSDL document

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
- is not represented in the WSDL document

validate(objectPath, parameterValues, options)

Use this method to validate an object that contains a specification.

To control the severity of reported defects, specify a validate option of the [bibus » validateOptionValidateSeverity](#) class.

The results are stored in an instance of the [bibus » asynchDetailReportValidation](#) class, which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,” on page 1417](#).

Use of this method requires:

- execute permission for the object
- traverse permission for all ancestors of the object

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply validate(
    com.cognos.developer.schemas.bibus._3.SearchPathSingleObject
    objectPath, com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asynchReply validate(searchPathSingleObject objectPath,
    parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [validate](#)

Implemented by the following services:

The services that implement the validate method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/
dataMovementService	http://www.ibm.com/xmlns/prod/cognos/ dataMovementService/201404/
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/ powerPlayService/201404/
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

The results of calling this method for a [bibus » baseReport](#) object are stored in an instance of the [bibus » asynchDetailReportValidation](#) class, which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

If the request first generates a prompt page for connection information, the page is stored in an instance of the [bibus » asynchDetailReportOutput](#) class, which is returned in the [bibus » asynchReply » details](#) property.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Capabilities	Class
bibus » userCapabilityEnum » canUseAnalysisStudio	bibus » analysis
bibus » userCapabilityEnum » canUseEventStudio	bibus » report
bibus » userCapabilityEnum » canUseQueryStudio	bibus » query
bibus » userCapabilityEnum » canUseReportStudio	bibus » interactiveReport , bibus » report , or bibus » reportTemplate

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This method can now be used with a [bibus » interactiveReport](#).

The capability rules were updated to include the rules related to the use of the [bibus » interactiveReport](#) class.

dataMovementService information

This information is specific to the [dataMovementService](#) service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Capabilities	Class
bibus » userCapabilityEnum » canUseDataManager	bibus » dataMovementTask

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

deliveryService information

This information is specific to the [deliveryService](#) service.

powerPlayService information

This information is specific to the [powerPlayService](#) service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Capabilities	Class
bibus » userCapabilityEnum » canUsePowerPlay	bibus » powerPlayServiceReportSpecification

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “New PowerPlay Service Methods” on page 1860

The documentation has been updated to indicate that this service now supports this method.

Input parameters

Use the following parameters when calling this method.

objectPath

Specifies the object associated with the request.

This argument

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is encoded as type `tns:asynchReply`

validateContentLocale(locale)

Use this method to confirm that the specified locale is a supported content locale.

Signatures

Java and Apache Axis

```
public boolean validateContentLocale(java.lang.String locale)
```

C# .NET

```
public bool validateContentLocale(string locale)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `validateContentLocale` method, along with the associated SOAP actions, are listed in the following table.

Table 138. Services implementing the validateContentLocale method.

Service	SOAP Action
systemService	http://www.ibm.com/xmlns/prod/cognos/systemService/201404/

Input parameters

Use the following parameters when calling this method.

locale

Specifies the locale to be validated as a supported content locale.

This argument

- is of type `language`
is encoded as type `xs:string`

Return values

This method returns the following values.

result

Returns `true` if the supplied locale is a supported content locale, and `false` otherwise.

This result

- is of type `boolean`
is encoded as type `xs:boolean`

validateProductLocale(locale)

Use this method to confirm that the specified locale is a supported product locale.

Signatures

Java and Apache Axis

```
public boolean validateProductLocale(java.lang.String locale)
```

C# .NET

```
public bool validateProductLocale(string locale)
```

References

Part of the following method sets:

- [system](#)

Implemented by the following services:

The services that implement the `validateProductLocale` method, along with the associated SOAP actions, are listed in the following table.

Table 139. Services implementing the `validateProductLocale` method.

Service	SOAP Action
<code>systemService</code>	<code>http://www.ibm.com/xmlns/prod/cognos/systemService/201404/</code>

Input parameters

Use the following parameters when calling this method.

locale

Specifies the locale to be validated as a supported product locale.

This argument

- is of type `language`
is encoded as type `xs:string`

Return values

This method returns the following values.

result

Returns `true` if the supplied locale is a supported product locale, and `false` otherwise.

This result

- is of type `boolean`
is encoded as type `xs:boolean`

validateSpecification(specification, parameterValues, options) – obsolete

Input parameters

Use the following parameters when calling this method.

specification

This argument

- is of type `bibus » reportServiceSpecification`
is not represented in the WSDL document

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type `bibus » parameterValue`
is not represented in the WSDL document

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This argument

- is an array of type [bibus » option](#)
is not represented in the WSDL document

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus » asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus » asynchReply » details](#) property.

This result

- is of type [bibus » asynchReply](#)
is not represented in the WSDL document

[validateSpecification\(specification, parameterValues, options\)](#)

Use this method to validate a specification.

To control the severity of reported defects, specify a validate option of the [bibus » validateOptionValidateSeverity](#) class.

This method may return the following values in the [bibus » asynchReply » status](#) property:

- [conversationComplete](#)
- [working](#)

You may make secondary requests after using this method, depending on the server response. For more information about the asynchronous conversation status and secondary requests in asynchronous conversations, see [Chapter 17, “Secondary requests,”](#) on page 1417.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsynchReply
validateSpecification(
com.cognos.developer.schemas.bibus._3.AsynchSpecification
specification, com.cognos.developer.schemas.bibus._3.ParameterValue[]
parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
options)
```

C# .NET

```
public asynchReply validateSpecification(asynchSpecification
specification, parameterValue[] parameterValues, option[] options)
```

References

Part of the following method sets:

- [validate](#)

Implemented by the following services:

The services that implement the `validateSpecification` method, along with the associated SOAP actions, are listed in the following table.

Service	SOAP Action
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/ batchReportService/201404/</code>
dataMovementService	<code>http://www.ibm.com/xmlns/prod/cognos/ dataMovementService/201404/</code>
powerPlayService	<code>http://www.ibm.com/xmlns/prod/cognos/ powerPlayService/201404/</code>
reportService	<code>http://www.ibm.com/xmlns/prod/cognos/reportService/ 201404/</code>

batchReportService and reportService information

This information is specific to the [batchReportService](#) and [reportService](#) services.

For a [bibus » baseReport](#) object, the results are stored in an instance of the [bibus » asynchDetailReportValidation](#) class, which is returned in the [bibus » asynchReply » details](#) property when the request is complete.

If the request first generates a prompt page for connection information, the page is stored in an instance of the [bibus » asynchDetailReportOutput](#) class, which is returned in the [bibus » asynchReply » details](#) property.

The following Java code snippet demonstrates how to use the [report » validateSpecification\(specification, parameterValues, options\)](#) – obsolete method with the report service.

To see this code in context, view the Java sample `ReportAdd/AddReport.java`. For more information about the samples, see Chapter 18, “Code samples and language-specific coding practices,” on page 1419.

```
asynchReply =
    connection.getReportService().validateSpecification(
        reportSpec,
        new ParameterValue[] {},
        new Option[] {});
...
for (int i = 0; i < asynchReply.getDetails().length; i++)
{
    if (asynchReply.getDetails()[i] instanceof AsynchDetailReportValidation)
    {
        ValidationDefects =
            ((AsynchDetailReportValidation)asynchReply.getDetails()[i]).getDefects();
    }
}
```

The following C# code snippet demonstrates how to use the [report » validateSpecification\(specification, parameterValues, options\)](#) – obsolete method with the report service.

To see this code in context, view the C# sample `AddReport/AddReport.cs`. For more information about the samples, see Chapter 18, “Code samples and language-specific coding practices,” on page 1419.

```
validationResults = cBIRS.validateSpecification(
```

```

        reportSvcSpec, new parameterValue[]{}}, validateOptions);
    ...
    for (int i = 0; i < validationResults.details.Length; i++)
    {
        if (validationResults.details[i] is asynchDetailReportValidation)
        {
            if (((asynchDetailReportValidation)
                validationResults.details[i]).defects.Value.Length > 0 )
            {
                defects =
                ((asynchDetailReportValidation)
                validationResults.details[i]).defects.Value;
            }
        }
    }
}

```

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 141. Capability rules for the validateSpecification method.

Specification	Capabilities	Class
Trusted	bibus » userCapabilityEnum » canUseAdaptiveAnalytics	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseAnalysisStudio	bibus » reportServiceAnalysisSpecification , bibus » reportServiceMetadataSpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseControllerStudio	bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseEventStudio	bibus » reportServiceMetadataSpecification
—	bibus » userCapabilityEnum » canUseEventStudio	bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseMetricsManagerAdministration	bibus » reportServiceMetadataSpecification
Trusted	bibus » userCapabilityEnum » canUseMetricStudio	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUsePlanningContributor	bibus » reportServiceMetadataSpecification or bibus » reportServiceReportSpecification
Trusted	bibus » userCapabilityEnum » canUseQueryStudio	bibus » reportServiceMetadataSpecification , bibus » reportServiceQuerySpecification , or bibus » reportServiceReportSpecification
—	bibus » userCapabilityEnum » canUseReportStudio	bibus » reportServiceInteractiveReportSpecification , bibus » reportServiceMetadataSpecification , or bibus » reportServiceReportSpecification

Table 141. Capability rules for the `validateSpecification` method. (continued)

Specification	Capabilities	Class
—	<code>bibus » userCapabilityEnum » canUseSpecifications</code>	<code>bibus » reportServiceAnalysisSpecification</code> , <code>bibus » reportServiceInteractiveReportSpecification</code> , <code>bibus » reportServiceMetadataSpecification</code> , <code>bibus » reportServiceQuerySpecification</code> , or <code>bibus » reportServiceReportSpecification</code>

To use this method, all conditions specified by any row must be satisfied:

- A specification is *Trusted* if it is digitally signed or was received via the secure dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch`).
- A specification is *Not Trusted* if it is not digitally signed and was received via the external dispatcher entry point (`http://server_name:9300/p2pd/servlet/dispatch/ext`).
- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

The capability rules were updated to include the rules related to the use of the `bibus » reportServiceReportSpecification` class with the `userCapabilityEnum » canUseDashboardViewer` value user capability.

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This method can now be used with a `bibus » reportServiceInteractiveReportSpecification`.

The capability rules were updated to include the rules related to the use of the `bibus » reportServiceInteractiveReportSpecification` class.

dataMovementService information

This information is specific to the `dataMovementService` service.

The following capability rules are enforced for this method.

The capability rules, along with the associated specifications and classes, are listed in the following table.

Table 142. Capability rules for the `validateSpecification` method.

Capabilities	Class
<code>bibus » userCapabilityEnum » canUseDataManager</code>	<code>bibus » dataMovementServiceSpecification</code>

To use this method, all conditions specified by any row must be satisfied:

- The object must be a member of one of the specified classes. If a class is not specified, then the class of the object is not used to determine whether the user can execute the method.
- The user must have all specified capabilities.

deliveryService information

This information is specific to the `deliveryService` service.

powerPlayService information

This information is specific to the [powerPlayService](#) service.

New in Version 10.1.0 – [“New PowerPlay Service Methods” on page 1860](#)

Reserved.

Input parameters

Use the following parameters when calling this method.

specification

Defines the specification associated with the request.

This argument

- is of type [bibus](#) » [asynchSpecification](#)
is encoded as type `tns:asynchSpecification`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This argument

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the [bibus](#) » [asynchReplyStatusEnum](#) enumeration set.

Any data requested can be obtained by examining the [bibus](#) » [asynchReply](#) » [details](#) property.

This result

- is of type [bibus](#) » [asynchReply](#)
is encoded as type `tns:asynchReply`

[wait\(conversation, parameterValues, options\)](#)

Use this method to notify the server that the issuer of the request is still waiting for the request to complete, and to request that the processing be continued.

Signatures

Java and Apache Axis

```
public com.cognos.developer.schemas.bibus._3.AsyncReply wait(
    com.cognos.developer.schemas.bibus._3.AsyncRequest conversation,
    com.cognos.developer.schemas.bibus._3.ParameterValue[]
    parameterValues, com.cognos.developer.schemas.bibus._3.Option[]
    options)
```

C# .NET

```
public asyncReply wait(asyncRequest conversation, parameterValue[]
    parameterValues, option[] options)
```

Example: Using the wait(conversation, parameterValues, options) Method with the Report Service in Java

The following Java code snippet demonstrates how to use the [async » wait\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the Java sample ExecReports/RunReport.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
while (rsr.getStatus() != AsyncReplyStatusEnum.complete)
{
    rsr =
        connection.getReportService().wait(
            rsr.getPrimaryRequest(),
            new ParameterValue[] {},
            new Option[] {});
}
```

Example: Using the wait(conversation, parameterValues, options) Method with the Job Service in Java

The following Java code snippet demonstrates how to use the [async » wait\(conversation, parameterValues, options\)](#) method with the job service.

To see this code in context, view the Java sample Submit/SubmitReport.java. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
asyncReply =
    connect.getJobService().wait(
        asyncReply.getPrimaryRequest(),
        new ParameterValue[] {},
        new Option[] {});
```

Example: Using the wait(conversation, parameterValues, options) Method with the Report Service in C# .NET

The following C# code snippet demonstrates how to use the [async » wait\(conversation, parameterValues, options\)](#) method with the report service.

To see this code in context, view the C# sample `PrintReport/PrintReport.cs`. For more information about the samples, see [Chapter 18, “Code samples and language-specific coding practices,”](#) on page 1419.

```
while (reportResponse.status != asynchReplyStatusEnum.complete)
{
    reportResponse = connection.CBIRS.wait(
        reportResponse.primaryRequest, new parameterValue[] {}, new
option[] {} );
}
```

References

Part of the following method sets:

- [asynch](#)

Implemented by the following services:

The services that implement the wait method, along with the associated SOAP actions, are listed in the following table.

<i>Table 143. Services implementing the wait method.</i>	
Service	SOAP Action
agentService	<code>http://www.ibm.com/xmlns/prod/cognos/agentService/201404/.absolute</code>
batchReportService	<code>http://www.ibm.com/xmlns/prod/cognos/batchReportService/201404/.absolute</code>
contentManagerService	<code>http://www.ibm.com/xmlns/prod/cognos/contentManagerService/201404/.absolute</code>
dataIntegrationService	<code>http://www.ibm.com/xmlns/prod/cognos/dataIntegrationService/201404/.absolute</code>
dataMovementService	<code>http://www.ibm.com/xmlns/prod/cognos/dataMovementService/201404/.absolute</code>
deliveryService	<code>http://www.ibm.com/xmlns/prod/cognos/deliveryService/201404/.absolute</code>
dimensionManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/dimensionManagementService/201404/.absolute</code>
eventManagementService	<code>http://www.ibm.com/xmlns/prod/cognos/eventManagementService/201404/.absolute</code>
idVizService	<code>http://www.ibm.com/xmlns/prod/cognos/idVizService/201404/.absolute</code>
indexSearchService	<code>http://www.ibm.com/xmlns/prod/cognos/indexSearchService/201404/.absolute</code>

Table 143. Services implementing the wait method. (continued)

Service	SOAP Action
indexUpdateService	http://www.ibm.com/xmlns/prod/cognos/indexUpdateService/201404/.absolute
jobService	http://www.ibm.com/xmlns/prod/cognos/jobService/201404/.absolute
metadataService	http://www.ibm.com/xmlns/prod/cognos/metadataService/201404/.absolute
migrationService	http://www.ibm.com/xmlns/prod/cognos/migrationService/201404/.absolute
mobileService	http://www.ibm.com/xmlns/prod/cognos/mobileService/201404/.absolute
monitorService	http://www.ibm.com/xmlns/prod/cognos/monitorService/201404/.absolute
planningAdministrationConsoleService	http://www.ibm.com/xmlns/prod/cognos/planningAdministrationConsoleService/201404/.absolute
planningRuntimeService	http://www.ibm.com/xmlns/prod/cognos/planningRuntimeService/201404/.absolute
planningTaskService	http://www.ibm.com/xmlns/prod/cognos/planningTaskService/201404/.absolute
powerPlayService	http://www.ibm.com/xmlns/prod/cognos/powerPlayService/201404/.absolute
queryService	http://www.ibm.com/xmlns/prod/cognos/queryService/201404/.absolute
relationalMetadataService	http://www.ibm.com/xmlns/prod/cognos/relationalMetadataService/201404/.absolute
reportService	http://www.ibm.com/xmlns/prod/cognos/reportService/201404/.absolute
repositoryService	http://www.ibm.com/xmlns/prod/cognos/repositoryService/201404/.absolute
saCAMService	http://www.ibm.com/xmlns/prod/cognos/sacamsrvc-auth/201404/.absolute

Input parameters

Use the following parameters when calling this method.

conversation

Specifies the current state of the asynchronous conversation. All methods that begin an asynchronous conversation, and most methods used within an asynchronous conversation, return this information as the value of the `bibus » asynchReply » primaryRequest` property.

The value of this parameter should be obtained from the most recent instance of the `bibus » asynchReply » primaryRequest` property returned from a method for the same conversation.

This argument

- is of type `bibus » asynchRequest`
is encoded as type `tns:asynchRequest`

parameterValues

Specifies the parameter values for the request.

Parameter values can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArray`

options

Specifies the options for the request.

Options can also be specified in other locations. For more information, see “[Specifying Options and Parameters](#)” on page 66.

This argument

- is an array of type `bibus » option`
is encoded as type `tns:optionArray`

Return values

This method returns the following values.

result

Returns the state of the asynchronous conversation. Possible asynchronous conversation states are defined in the `bibus » asynchReplyStatusEnum` enumeration set.

Any data requested can be obtained by examining the `bibus » asynchReply » details` property.

This result

- is of type `bibus » asynchReply`
is encoded as type `tns:asynchReply`

Chapter 15. Classes

The content store is a hierarchical collection of objects. A single instance of the root class, represented in a search path by a slash (/), is at the top of the hierarchy.

The classes of these objects are described in this section of the document.

Classes in Java

In the Java toolkit, you cannot manipulate the properties of the classes listed in this section by dereferencing instances of these classes. Instead you use accessor methods. The names of these accessor methods are derived from the corresponding property names.

For example, the `bibus » contact` class has the properties `businessPhone` and `email`, among others. In Java, these properties are manipulated using the methods `getBusinessPhone()`, `setBusinessPhone(StringProp businessPhone)`, `getEmail()`, and `setEmail(StringProp email)`, respectively, of the Java `contact` class.

account

Contains the information used to define a user account.

Most instances of this class represent user accounts defined in a security system. IBM Cognos Analytics can process the user account information stored in a security system by changing the startup configuration.

The only `account` instance that does not represent a user account in a security system is the `Anonymous account user` in the Cognos namespace.

Because most `account` instances are based on information in a security system, IBM Cognos Analytics may constrain your ability to manipulate these objects. For example, you may not be able to change the value of a property, or the property may not return a value when queried.

The name of an instance may match the name of a sibling. In this case, you must use a different search strategy to distinguish between objects.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- `bibus » asynchDetailEventRecord » cancelledBy`
- `bibus » dataSourceSignon » consumers`
- `bibus » adminFolder » contact`
- `bibus » aliasRoot » contact`
- `bibus » annotation » contact`
- `bibus » annotationFolder » contact`
- `bibus » archiveLocation » contact`
- `bibus » baseAgentDefinition » contact`
- `bibus » baseDataIntegrationTask » contact`

- [bibus » baseDataMovementTask » contact](#)
- [bibus » basePowerPlay8Report » contact](#)
- [bibus » basePowerPlayClass » contact](#)
- [bibus » baseReport » contact](#)
- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)

- [bibus » URL » contact](#)
- [bibus » baseROLAPDataSource » dataSourceAccessAccount](#)
- [bibus » group » distributionMembers](#)
- [bibus » role » distributionMembers](#)
- [bibus » session » identity](#)
- [bibus » distributionList » members](#)
- [bibus » group » members](#)
- [bibus » role » members](#)
- [bibus » baseAgentDefinition » notificationList](#)
- [bibus » baseReport » notificationList](#)
- [bibus » baseClass » owner](#)
- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)
- [bibus » group » securityMembers](#)
- [bibus » role » securityMembers](#)
- [bibus » asynchDetailEventRecord » suspendedBy](#)
- [bibus » asynchDetailEventRecord » user](#)
- [bibus » history » user](#)

Container Information

Contains instances of the following classes

- [bibus » credential](#)
- [bibus » dataSetFolder](#)
- [bibus » dataSourceCredential](#)
- [bibus » favoritesFolder](#)
- [bibus » folder](#)
- [bibus » mruFolder](#)
- [bibus » personalizationFolder](#)
- [bibus » session](#)
- [bibus » subscriptionFolder](#)

Contained by instances of the following classes

- [bibus » configuration](#)
- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Properties

This class has the following properties.

businessPhone

Specifies the business telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `teLephonenumbeR` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

contentLocale

Specifies the user's preferred locale, which determines the language and data format of the returned content. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

This property

- is of type `language`
 - is encoded as type `tns:languageProp`
- can contain at least 255 characters

credentials

Contains the credentials for the user, typically an account name and password.

This property

- must have at most 1 item

dataSourceCredentials

Contains the personal data source credential objects for this account.

A data source credential object is created when a user logs into a data source, if the user chooses to have their credentials persisted. The `canUsePersonalDataSourceCredentials` capability is used in IBM Cognos Analytics to determine if the option to save a user's credentials is displayed.

New in Version 10.1.0 – “[Personal Data Source Credentials](#)” on page 1851

This property was added.

desktop

Contains the items associated with the user's personal desktop.

This property

- must have exactly 1 item

email

Specifies the email address. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `mail` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 128 characters

favorites

Contains the favourites folder for this account.

This property

- must have at least 1 item

New in Version 10.1.0 – “[Content Pane Improvements](#)” on page 1862

This property was added.

faxPhone

Specifies the fax number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `facsimiletelephonenumber` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

format

Specifies the rendering format.

When used in the `bibus » account` class or the `bibus » contact` class, this property specifies the preferred output format of reports for the account or contact.

When used in the `bibus » documentContent` class or `bibus » output` class this property specifies the format of the data contained in the object

This property

- is of type `bibus » outputFormatEnum`
 - is encoded as type `tns:nmtokenProp`
- must contain no more than 10 characters
- is searchable

givenName

Specifies the given name. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `givenname` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

homePhone

Specifies the home telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `homephone` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

horizontalElementsRenderingLimit

Specifies the number of elements to be rendered horizontally on a canvas of a predefined size, such as an HTML page. Provides the default value for the runOption `horizontalElements`.

This property

- is of type `int`

is encoded as type `tns:intProp`

mobileDeviceID

Identifies the user's mobile device. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

mobilePhone

Specifies the mobile telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `mobile` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

mostRecentlyUsedList

Contains the most recently used folder for the account.

This property

- must have exactly 1 item

notificationEMail

Specifies the alert email address for the account.

This property

- is of type `bibus » addressSMTP`
 - is encoded as type `tns:addressSMTPProp`
- can contain at least 255 characters

New in Version 8.3 — “[Report Email Alerts](#)” on page 1900

This property was added.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type `bibus » option`
 - is encoded as type `tns:optionArrayProp`

page

Refers to the user's preferred page settings, such as the height and width of the page.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » pageDefinition`

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

pageOrientation

Specifies the user's preferred page orientation.

This property

- is of type [bibus » pageOrientationEnum](#)
 - is encoded as type `tns:nmtokenProp`
- can contain at least 255 characters

pagerPhone

Specifies the pager telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `pager` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`
- is read-only

personalizationFolder

Contains the personalization folder for the account.

This property

- must have at most 1 item

New in Version 8.4 – [“Dashboards”](#) on page 1894

This property was added.

portalPage

Specifies the XML definition of the user's My Page.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`

portalPages

Contains the "Public Pages" used by the account.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » content`, `bibus » folder`, or `bibus » pagelet`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

portalPreferences – obsolete

postalAddress

Specifies the postal address. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `postalAddress` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 4095 characters

productLocale

Specifies the user's preferred locale for the product, which determines the language and data format for the user interface. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

This property

- is of type `language`
 - is encoded as type `tns:languageProp`
- can contain at least 255 characters

repositoryRules

This property

- is an array of type `bibus » repositoryRule`
 - is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The `bibus » account`, `bibus » namespace`, and `bibus » namespaceFolder` classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

New in Version 8.3 – “[Package Hierarchies](#)” on page 1913

This property can now be acquired.

sessions

Contains the list of session objects. Each session name matches the security passport ID in effect when the object was created.

subscriptionFolder

Contains the subscription folder for the account.

This property

- must have exactly 1 item

New in Version 8.3 – “[Conditional Subscriptions](#)” on page 1899

This property was added.

surname

Specifies the surname. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `sn` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 32 characters

timeZoneID

Specifies the time zone for the object. The specification for the ID string follows the International Components for Unicode (ICU) time zone format.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters
- is searchable

useAccessibilityFeatures – deprecated

Specifies whether accessibility features should be used for this account.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`

New in Version 10.1.0 – “[Accessibility](#)” on page 1859

This property is deprecated. Use the following properties instead:

- `bibus` » `genericOptionBoolean` class

- [bibus » account » options property](#)
- [bibus » systemOptionEnum » accessibilityFeatures value](#)

userName

Specifies the user name. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `uid` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 64 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

verticalElementsRenderingLimit

Specifies the number of elements to be rendered vertically on a canvas of a predefined size, such as an HTML page. Provides the default value for the runOption `verticalElements`.

This property

- is of type `int`
 - is encoded as type `tns:intProp`

adaptiveAnalyticsService

Defines run-time configuration parameters for the adaptive analytics service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Adaptive Analytics Service” on page 1856](#)

This class was added.

Related information:

- [“Adaptive Analytics service” on page 5](#)

Properties

This class has the following properties.

aasAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasAuditLevel

Specifies the auditing level for the adaptive analytics service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the adaptive analytics service, low affinity requests are used by client applications of IBM® Cognos® Analytic Applications to retrieve metadata and query information.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `4`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

addOptions

Defines the options you can specify for the `content » add(parentPath, objects, options)` method.

References

Used by the following method parameters:

- `content » add(parentPath, objects, options) » options`
- `content » addAnnotations(containerPath, objects, options) » options`
- `drillThrough » addDrillPath(parentPath, object, options) » options`
- `report » add(parentPath, object, options) » options`

Properties

This class has the following properties.

dataEncoding

Specifies the encoding used for the added data.

This property

- is of type `bibus » encodingEnum`
 - is encoded as type `tns:encodingEnum`

faultIfObjectReferenced

Specifies how references to objects being deleted or replaced are processed by Content Manager.

If this property is set to `false`, any reference to the objects being deleted are automatically deleted. If this property is set to `true`, Content Manager generates a fault if any object in the content store continues to reference any of the deleted objects.

During recursive delete, a fault is not generated if one deleted object refers to another object that is also deleted.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

ignoreInvalidObjectReference

Specifies how invalid object references in ID-based reference properties are processed by Content Manager. A reference is invalid if it refers to a non-existent object.

If this property is set to `false`, Content Manager generates a fault if an object contains an invalid object reference in an ID-based reference property.

If this property is set to `true`, any invalid object reference in an ID-based reference property is automatically ignored by Content Manager when updating the content store.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

returnProperties

Specifies the list of properties returned for each added object.

If no properties are specified, the values for the [defaultName](#), [searchPath](#), and [storeID](#) properties are returned.

If properties are specified, only the values for the specified properties are returned. For example, if you specify [creationTime](#), only the value for this property is returned. If you also want the value for the [searchPath](#) property, you must specify it as well.

This property

- is an array of type [bibus » propEnum](#)
 - is encoded as type `tns:propEnumArray`

updateAction

Specifies what happens when an object already exists at the new object's location in the content store.

This property

- is of type [bibus » updateActionEnum](#)
 - is encoded as type `tns:updateActionEnum`

addressSMTP

Defines the type to represent SMTP addresses, such as `name@company.com`.

This class

- inherits properties from the `string`

References

Used by the following properties:

- [bibus » account » notificationEMail](#)

- [bibus » addressSMTPArrayProp » value](#)
- [bibus » addressSMTPProp » value](#)
- [bibus » deliveryOptionAddressSMTPArray » value](#)

addressSMTPArrayProp

Defines the array property class for the [bibus » addressSMTP](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » addressSMTP](#)
is encoded as type `tns:addressSMTPArray`

addressSMTPProp

Defines the simple property class for the [bibus » addressSMTP](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » account » notificationEMail](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » addressSMTP](#)
is encoded as type `tns:addressSMTP`

adminFolder

Contains the set of content objects used by administrators to automate repetitive tasks.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [adminFolder](#)
- [bibus](#) » [contentTask](#)
- [bibus](#) » [exportDeployment](#)
- [bibus](#) » [importDeployment](#)
- [bibus](#) » [indexUpdateTask](#)
- [bibus](#) » [jobDefinition](#)
- [bibus](#) » [migrationTask](#)
- [bibus](#) » [planningMacroTask](#)
- [bibus](#) » [queryServiceTask](#)

Contained by instances of the following classes

- [bibus](#) » [adminFolder](#)
- [bibus](#) » [root](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)

has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

items

Contains a collection of child objects.

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was extended to allow instances of the [bibus](#) » [migrationTask](#) class to be contained by instances of this property.

New in Version 10.1.0 – [“Query Service Administration Task” on page 1869](#)

This property was extended to allow instances of the [bibus](#) » [queryServiceTask](#) class to be contained by instances of this property.

agentDefinition

Defines an agent as a series of tasks and defines parameters for those tasks.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredAgentDefinition](#) class

References

Used by the following properties:

- [bibus » agentDefinitionView » base](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the agentDefinition class.

Action	Mode	Service	Method
Run	All	agentService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » report](#)

What's new

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This class is now derived from the [bibus » authoredAgentDefinition](#) class.

Properties

This class has the following properties.

eventKey – obsolete

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property is obsolete and was removed. Use the [bibus » authoredAgentDefinition » eventKey](#) property instead.

items – obsolete

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property is obsolete and was removed. Use the [bibus » authoredAgentDefinition » items](#) property instead.

report

Specifies the report used to determine the agent's events.

sequencing – obsolete

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property is obsolete and was removed. Use the [bibus » authoredAgentDefinition » sequencing](#) property instead.

tasks – obsolete

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property is obsolete and was removed. Use the [bibus » authoredAgentDefinition » tasks](#) property instead.

agentDefinitionView

Defines a customized view of an agent definition.

Important: The [bibus » baseAgentDefinition » runAsOwner](#) property is ignored for this class.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseAgentDefinition](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `agentDefinitionView` class.

Action	Mode	Service	Method
Run	All	agentService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » reportView](#)

Properties

This class has the following properties.

base

Refers to the object on which this agent definition view is based.

An agent definition view can still exist, even when the object it is based on is deleted or moved.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » agentDefinition](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

reportView

Specifies the report view used to determine the agent view's events.

agentOption

Defines the abstract base class for all agent option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » agentOptionBoolean](#)

Properties

This class has the following properties.

name

Identifies the agent option.

This property

- is of type [bibus » agentOptionEnum](#)
is encoded as type `tns:agentOptionEnum`

agentOptionBoolean

Defines boolean values for the agent options.

This class

- inherits properties from the [bibus » agentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the agent option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

agentOutputEnumProp

Defines the simple property class for the [bibus » agentOutputEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » shortcutAgentRSSTask » link](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » agentOutputEnum](#)
is encoded as type `tns:agentOutputEnum`

agentOutputHotList

Defines output created by the comparison of two instances of report output.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseClass](#) class

References

Used by the following properties:

- [bibus » historyDetailAgentService » output](#)

Container Information

Contains instances of the following classes

- [bibus » annotationFolder](#)
- [bibus » output](#)

Contained by instances of the following classes

- [bibus » agentState](#)

Properties

This class has the following properties.

allowAnnotations

Specifies whether the outputs in this agent hot list can be annotated by report consumers.

The value of this property is normally determined by the value of the [allowAnnotations](#) option when the output was created.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`

New in Version 8.4 — “Report Output Annotations” on page 1895

This property was added.

annotationFolder

Specifies the annotation folder for this agent hot list.

This property

- must have at most 1 item

New in Version 8.4 — “Report Output Annotations” on page 1895

This property was added.

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable

New in Version 8.4 — “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

output

Contains the output for this agent hot list.

agentService

Defines run-time configuration parameters for the [agentService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

asAuditLevel

Specifies the auditing level for the agent service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

asConnections

Specifies the maximum number of connections that a process of the agent service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

asMaximumEmailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the agent service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – [“New Email Configuration Parameters” on page 1868](#)

This property was added.

asPeakConnections

Specifies the number of connections that a agent service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

agentState

Defines the transient state maintained during the execution of an `bibus » agentDefinition`.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » baseClass` class

Container Information

Contains instances of the following classes

- `bibus » agentOutputHotList`
- `bibus » agentTaskState`
- `bibus » reportView`

Contained by instances of the following classes

- `bibus » transientStateFolder`

Properties

This class has the following properties.

definition

Specifies the agent definition that manages this state.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » baseAgentDefinition](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

items

Contains the set of child hot list objects for the agent.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » agentOutputHotList](#)

reportView

Specifies the report view used to store output that defines the agent's events.

This property

- must have exactly 1 item

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » agentOutputHotList class		2	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

taskStates

Contains the set of task state objects for this agent state.

agentTaskDefinition

Specifies an object to be executed as part of an [bibus » agentDefinition](#).

This class also specifies options and parameter values that are used when executing the object.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » agentTaskState » definition](#)

Container Information

Contained by instances of the following classes

- [bibus » authoredAgentDefinition](#)

Related information:

- [“Running Agents” on page 76](#)

Properties

This class has the following properties.

eventKey

Specifies additional query items in the report contained by the containing agent that can be checked when a change event occurs.

This property

- is an array of type `string`
is encoded as type `tns:stringArrayProp`

eventTypes

Specifies the event types that are of interest to this task.

This property

- is an array of type [bibus » eventTypeEnum](#)
is encoded as type `tns:eventTypeEnumArrayProp`

filterDataItemName

Specifies the name of a data item that must be defined in the agent's condition report (the [bibus » report](#) object contained by the [bibus » agentDefinition](#) object). Event conditions that evaluate to `true` for the named data item are processed by this agent task.

If the [bibus » agentTaskDefinition » filterDataItemName](#) property is not specified or references a data item from the agent's condition report that does not produce boolean values, all events are passed to the agent task.

This property

- is of type [bibus » metadataModelItemName](#)
is encoded as type `tns:metadataModelItemNameProp`

New in Version 8.3 — “Agent Task Event Filters” on page 1901

This property was added.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

parameterAssignments

Contains the parameter assignments for the object.

This property

- is an array of type [bibus » baseParameterAssignment](#)
is encoded as type `tns:baseParameterAssignmentArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

runCondition

Specifies when the agent task should be run based on the [bibus » runStatusEnum](#) of other tasks.

This property

- is of type [bibus » runConditionEnum](#)
is encoded as type `tns:runConditionEnumProp`

New in Version 8.3 — “Run Agent Tasks on Failure” on page 1912

This property was added.

taskObject

Refers to the object that runs as this agent task.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay8Report](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » exportDeployment](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » queryServiceTask](#), [bibus » storedProcedureTask](#), or [bibus » webServiceTask](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property was extended to allow instances of the [bibus » dataMovementTaskAlias](#) class to be referenced by instances of this property.

New in Version 8.4 – “Migration Service” on page 1883

This property was extended to allow instances of the [bibus » migrationTask](#) class to be referenced by instances of this property.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

This property was extended to allow instances of the [bibus » queryServiceTask](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Human Task” on page 1862

This property was extended to allow instances of the [bibus » humanTask](#) class to be referenced by instances of this property.

agentTaskState

Defines the transient state maintained during the execution of an [bibus » agentTaskDefinition](#).

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » agentState](#)

Properties

This class has the following properties.

definition

Specifies the definition that manages this state.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » agentTaskDefinition](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

related

References the objects tracked by the agent task.

This property

- is an array of type [bibus » baseClass](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

aliasLocation – obsolete

Reserved.

What's new

New in Version 8.4 – [“Package Data Sources” on page 1895](#)

This class was added.

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Administration” on page 1836](#)

This class is deprecated.

Properties

This class has the following properties.

aliasRoot

Reserved.

contact

Refers to a person or group of people who are responsible for the object.

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

uri

Reserved.

aliasRoot

Reserved.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » configuration](#)

What's new

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Administration” on page 1836

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

unixURI

Reserved.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

windowsURI

Reserved.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

analysis

Contains information used to define an IBM Cognos Business Intelligence Analysis specification.

Instances of this class are usually created in IBM Cognos Analysis Studio. For example, you would create an analysis to analyze sales data.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredReport](#) class

References

Used by the following properties:

- [bibus](#) » [packageConfiguration](#) » [defaultAnalysis](#)
- [bibus](#) » [drillPath](#) » [target](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the analysis class.

Table 147. Services and methods for the analysis class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus](#) » [session](#)

ancestorInfo

Defines the structure used to return information about the ancestors of an object.

References

Used by the following properties:

- [bibus](#) » [baseClass](#) » [ancestors](#)
- [bibus](#) » [deploymentObjectInformation](#) » [ancestors](#)
- [bibus](#) » [ancestorInfoArrayProp](#) » [value](#)
- [bibus](#) » [ancestorInfoProp](#) » [value](#)

Properties

This class has the following properties.

objectClass

Specifies the class of the object.

This property

- is of type [bibus](#) » [classEnum](#)
is encoded as type `tns:classEnum`

searchPath

Specifies the location of the object in the content store.

This property

- is of type `string`

is encoded as type `xs:string`

storeID

Identifies the object in the content store.

This property

- is of type [bibus » guid](#)

is encoded as type `tns:guid`

title

Specifies the object name closest to the `contentLocale` value.

This property

- is of type `string`

is encoded as type `xs:string`

ancestorInfoArrayProp

Defines the array property class for the [bibus » ancestorInfo](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » baseClass » ancestors](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » ancestorInfo](#)

is encoded as type `tns:ancestorInfoArray`

ancestorInfoProp

Defines the simple property class for the [bibus » ancestorInfo](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » ancestorInfo](#)

is encoded as type `tns:ancestorInfo`

annotation

Defines the class for annotations.

Annotations contain additional information that questions, clarifies, or explains content in an object that is stored in the content store.

Custom applications can use the `content » addAnnotations(containerPath, objects, options)` method to create annotations.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contained by instances of the following classes

- `bibus » annotationFolder`

What's new

New in Version 8.4 — “Report Output Annotations” on page 1895

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type `bibus » baseClass`

has items that must be of class `bibus » account`, `bibus » contact`, `bibus » distributionList`, `bibus » group`, or `bibus » role`

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

outputs

Refers to the outputs associated with the annotation.

This property

- is an array of type `bibus » baseClass`

has items that must be of class [bibus » output](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path

selectionContext

Specifies the data referred to by the annotation.

This property

- is of type [bibus » xmlEncodedXML](#)

is encoded as type `tns:xmlEncodedXMLProp`

annotationFolder

Defines the class that contains annotations.

An instance of this class is used to override the security policy of the containing object in order to allow consumers to store their annotations.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » annotation](#)

Contained by instances of the following classes

- [bibus » agentOutputHotList](#)
- [bibus » reportVersion](#)

What's new

New in Version 8.4 — “[Report Output Annotations](#)” on page 1895

This class was added.

Properties

This class has the following properties.

annotations

Contains the annotations.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path

- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

annotationService

Defines run-time configuration parameters for the annotation service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Annotation Service” on page 1864](#)

This class was added.

Related information:

- [“Annotation service” on page 6](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

- can be acquired from a containing object

ansAnnotationLifetime

Specifies the lifetime of the annotation in XML Schema 1.0 type `xs:duration` form.

The lifetime is the length of time in days after the entry associated with the annotation is deleted. For example, if the lifetime for an annotation is set to 60 days, the annotation is deleted 60 days after the associated report is deleted.

For an `annotation` object associated with a `reportVersion` object, the annotations are deleted when the report is deleted.

The default is 180 days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of `P180D`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

ansAuditLevel

Specifies the auditing level for the annotation service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

anyTypeMIMEProp

Defines the simple MIME property class for the anyType.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [model](#) » [model](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

The value may be placed in a MIME attachment.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set toMIME. The encoding parameter varies from method to method; see the [bibus](#) » [addOptions](#) » [dataEncoding](#) propertyand the [bibus](#) » [queryOptions](#) » [dataEncoding](#) propertyfor additional details.

This property

- is of type anyType
- is encoded as type `xs:string`

anyTypeProp

Defines the simple property class for the anyType.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [adaptiveAnalyticsService](#) » [advancedSettings](#)
- [bibus](#) » [agentService](#) » [advancedSettings](#)
- [bibus](#) » [annotationService](#) » [advancedSettings](#)
- [bibus](#) » [batchReportService](#) » [advancedSettings](#)
- [bibus](#) » [configuration](#) » [advancedSettings](#)
- [bibus](#) » [configurationFolder](#) » [advancedSettings](#)
- [bibus](#) » [contentManagerCacheService](#) » [advancedSettings](#)
- [bibus](#) » [contentManagerService](#) » [advancedSettings](#)
- [bibus](#) » [dataAdvisorService](#) » [advancedSettings](#)
- [bibus](#) » [dataIntegrationService](#) » [advancedSettings](#)
- [bibus](#) » [dataMovementService](#) » [advancedSettings](#)
- [bibus](#) » [deliveryService](#) » [advancedSettings](#)
- [bibus](#) » [dimensionManagementService](#) » [advancedSettings](#)

- [bibus » dispatcher » advancedSettings](#)
- [bibus » eventManagementService » advancedSettings](#)
- [bibus » EVService » advancedSettings](#)
- [bibus » graphicsService » advancedSettings](#)
- [bibus » humanTaskService » advancedSettings](#)
- [bibus » idVizService » advancedSettings](#)
- [bibus » indexDataService » advancedSettings](#)
- [bibus » indexSearchService » advancedSettings](#)
- [bibus » indexUpdateService » advancedSettings](#)
- [bibus » jobService » advancedSettings](#)
- [bibus » logService » advancedSettings](#)
- [bibus » metadataService » advancedSettings](#)
- [bibus » metricsManagerService » advancedSettings](#)
- [bibus » migrationService » advancedSettings](#)
- [bibus » mobileService » advancedSettings](#)
- [bibus » monitorService » advancedSettings](#)
- [bibus » planningAdministrationConsoleService » advancedSettings](#)
- [bibus » planningDataService » advancedSettings](#)
- [bibus » planningRuntimeService » advancedSettings](#)
- [bibus » planningTaskService » advancedSettings](#)
- [bibus » powerPlayService » advancedSettings](#)
- [bibus » presentationService » advancedSettings](#)
- [bibus » queryService » advancedSettings](#)
- [bibus » relationalMetadataService » advancedSettings](#)
- [bibus » reportDataService » advancedSettings](#)
- [bibus » reportService » advancedSettings](#)
- [bibus » repositoryService » advancedSettings](#)
- [bibus » saCAMService » advancedSettings](#)
- [bibus » systemService » advancedSettings](#)
- [bibus » credential » credentials](#)
- [bibus » dataSourceCredential » credentials](#)
- [bibus » dataSourceSignon » credentials](#)
- [bibus » baseAgentDefinition » ownerPassport](#)
- [bibus » baseDataIntegrationTask » ownerPassport](#)
- [bibus » baseDataMovementTask » ownerPassport](#)
- [bibus » baseReport » ownerPassport](#)
- [bibus » account » portalPage](#)
- [bibus » authoredReport » specification](#)
- [bibus » reportCache » specification](#)
- [bibus » reportVersion » specification](#)
- [bibus » runTimeState » state](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `anyType`
is encoded as type `xs:string`

anyURIArrayProp

Defines the array property class for the anyURI.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [baseROLAPDataSource](#) » [capabilities](#)
- [bibus](#) » [dataSource](#) » [capabilities](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type `anyURI`
is encoded as type `tns:anyURIArray`

anyURIProp

Defines the simple property class for the anyURI.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [activeJMXProxyURI](#)
- [bibus](#) » [planningApplication](#) » [applicationURL](#)
- [bibus](#) » [configuration](#) » [bpmRestURI](#)
- [bibus](#) » [configuration](#) » [collaborationDiscoveryURI](#)
- [bibus](#) » [graphic](#) » [dataPathInfo](#)
- [bibus](#) » [output](#) » [dataPathInfo](#)
- [bibus](#) » [visualization](#) » [dataPathInfo](#)
- [bibus](#) » [dispatcher](#) » [dispatcherPath](#)

- [bibus » basePowerPlayClass » gateway](#)
- [bibus » configuration » glossaryURI](#)
- [bibus » uiClass » iconURI](#)
- [bibus » urlRSSTask » link](#)
- [bibus » configuration » metadataInformationURI](#)
- [bibus » namespace » namespaceFormat](#)
- [bibus » configuration » qsGCPolicy](#)
- [bibus » configurationFolder » qsGCPolicy](#)
- [bibus » dispatcher » qsGCPolicy](#)
- [bibus » queryService » qsGCPolicy](#)
- [bibus » authoredReport » queryMode](#)
- [bibus » systemMetricThresholds » systemMetric](#)
- [bibus » configuration » temporaryObjectLocation](#)
- [bibus » configuration » unixRepositoryURI](#)
- [bibus » aliasRoot » unixURI](#)
- [bibus » archiveLocation » uri](#)
- [bibus » humanTask » uri](#)
- [bibus » URL » uri](#)
- [bibus » webServiceTask » uri](#)
- [bibus » configuration » windowsRepositoryURI](#)
- [bibus » aliasRoot » windowsURI](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `anyURI`

is encoded as type `xs:string`

archiveDescriptor

Specifies additional information about the object that will be stored in the archive location.

References

Used by the following properties:

- [bibus » archiveOptionDescriptor » value](#)

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

asOfTime

Specifies the "as of" time of the archived object.

See the [bibus » reportVersion » asOfTime](#) property for additional details.

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

burstKey

Specifies the burst key value for the archived object.

See the [bibus » output » burstKey](#) property for additional details.

This property

- is of type `string`
is encoded as type `xs:string`

defaultDescription

Specifies the default description of the archived object.

See the [bibus » uiClass » defaultDescription](#) property for additional details.

This property

- is of type `string`
is encoded as type `xs:string`

defaultName

Specifies the default name of the archived object.

See the [bibus » baseClass » defaultName](#) property for additional details.

This property

- is of type `string`
is encoded as type `xs:string`

executionTime

Specifies the time when the object that created the archived object was run.

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

filename

Specifies the URI of the object in the archive location.

This property

- is of type `anyURI`
is encoded as type `xs:string`

options

Specifies the options used to created the archived object.

See the [bibus » reportVersion » options](#) property for additional details.

This property

- is an array of type [bibus » option](#)

is encoded as type `tns:optionArray`

parameters

Specifies the parameter values used to create the archived object.

See the [bibus » reportVersion » parameters](#) property for additional details.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

reportSearchPath

Specifies the search path of the report that created the archived object.

See the [bibus » baseClass » searchPath](#) property for additional details.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

reportStoreID

Specifies the storeID of the report that created the archived object.

See the [bibus » baseClass » storeID](#) property for additional details.

This property

- is of type [bibus » guid](#)
is encoded as type `tns:guid`

reportViewSearchPath

Specifies the search path of the report view that created the archived object.

See the [bibus » baseClass » searchPath](#) property for additional details.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

reportViewStoreID

Specifies the storeID of the report view that created the archived object.

See the [bibus » baseClass » storeID](#) property for additional details.

This property

- is of type [bibus » guid](#)
is encoded as type `tns:guid`

archiveLocation

Defines an archive location.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » configuration](#)

What's new

New in Version 8.3 — “[Saving Report Output to File System](#)” on page 1901

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

uri

Specifies the URI of the archive location.

URIs using the `file` scheme must refer to a location within the location established by the administrator in IBM Cognos Configuration.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

archiveOption

Defines the abstract base class for all archive option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » archiveOptionConflictResolution](#)
- [bibus » archiveOptionDescriptor](#)

- [bibus » archiveOptionSearchPathSingleObject](#)
- [bibus » archiveOptionString](#)

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

name

Identifies the archive option.

This property

- is of type [bibus » archiveOptionEnum](#)
is encoded as type `tns:archiveOptionEnum`

archiveOptionConflictResolution

Defines [bibus » archiveConflictResolutionEnum](#) values for the archive options.

This class

- inherits properties from the [bibus » archiveOption](#) class

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the archive option.

This property

- is of type [bibus » archiveConflictResolutionEnum](#)
is encoded as type `tns:archiveConflictResolutionEnum`

archiveOptionDescriptor

Defines [bibus » archiveDescriptor](#) values for the archive options.

This class

- inherits properties from the [bibus » archiveOption](#) class

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the archive option.

This property

- is of type [bibus » archiveDescriptor](#)
is encoded as type `tns:archiveDescriptor`

archiveOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the archive options.

This class

- inherits properties from the [bibus » archiveOption](#) class

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the archive option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

archiveOptionString

Defines string values for the archive options.

This class

- inherits properties from the [bibus » archiveOption](#) class

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the archive option.

This property

- is of type `string`

is encoded as type `xs:string`

asynchDetail

Defines the base class for all classes used to support asynchronous response details.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » asynchDetailAgentNotificationStatus](#)
- [bibus » asynchDetailAsynchSpecification](#)
- [bibus » asynchDetailContext](#)
- [bibus » asynchDetailCount](#)
- [bibus » asynchDetailDrillPathObject](#)
- [bibus » asynchDetailDrillThroughRequest](#)
- [bibus » asynchDetailDrillThroughTarget](#)
- [bibus » asynchDetailDrillThroughTargetURI](#)
- [bibus » asynchDetailEventID](#)
- [bibus » asynchDetailEventRecord](#)
- [bibus » asynchDetailIndexData](#)
- [bibus » asynchDetailIndexTerms](#)
- [bibus » asynchDetailMessages](#)
- [bibus » asynchDetailMIMEAttachment](#)
- [bibus » asynchDetailParameters](#)
- [bibus » asynchDetailParameterValue](#)
- [bibus » asynchDetailPromptPage](#)
- [bibus » asynchDetailReportMetadata](#)
- [bibus » asynchDetailReportObject](#)
- [bibus » asynchDetailReportOutput](#)
- [bibus » asynchDetailReportStatus](#)
- [bibus » asynchDetailReportValidation](#)
- [bibus » asynchDetailROLAPDataSourceState](#)
- [bibus » asynchDetailSelectionContext](#)
- [bibus » asynchDetailUnstructuredData](#)

References

Used by the following properties:

- [bibus » asynchReply » details](#)

asynchDetailAgentNotificationStatus

Defines the class that contains alert status information.

This class

- inherits properties from the [bibus » asynchDetail](#) class

What's new

New in Version 8.3 — “Report Email Alerts” on page 1900

This enumeration set is now used to indicate the alert status of [bibus](#) » [baseReport](#) objects.

Properties

This class has the following properties.

status

Specifies whether the object supports alerts, and if so, whether the user is on the alert list.

This property

- is of type [bibus](#) » [agentNotificationStatusEnum](#)
 - is encoded as type `tns:agentNotificationStatusEnum`
- is read-only

[asynchDetailAsynchSpecification](#)

Defines the class that contains an [bibus](#) » [asynchSpecification](#).

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 10.1.0 — “Support for Lineage Requests” on page 1855

This class was added.

New in Version 10.1.0 — “Batch Report Service/Report Service Optimizations” on page 1870

This class was added.

Properties

This class has the following properties.

specification

Contains the asynchronous specification.

This property

- is of type [bibus](#) » [asynchSpecification](#)
 - is encoded as type `tns:asynchSpecification`
- is read-only

[asynchDetailContext](#)

Defines the class that contains context data.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This class was added.

Properties

This class has the following properties.

context

Contains the context data.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchDetailCount

Reserved.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

count

Specifies the count.

This property

- is of type `int`
 - is encoded as type `xs:int`

asynchDetailDrillPathObject

Defines the class returned by the [drillThrough](#) » [queryDrillPath\(objectPath, parameterValues, options\)](#) method, in the [bibus](#) » [asynchReply](#) » [details](#) property.

It contains a [bibus](#) » [drillPath](#) object, as well as information indicating whether the drill-through specification was upgraded.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.4 — “[Supporting New Drill-through Targets](#)” on page 1890

This class was added.

Properties

This class has the following properties.

drillPath

Contains the [bibus](#) » [drillPath](#) object.

This property

- is of type [bibus](#) » [drillPath](#)
 - is encoded as type `tns:drillPath`
- is read-only

upgraded

Specifies whether the drill-through specification was upgraded to the current version.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- is read-only

asyncDetailDrillThroughRequest

Defines the class used to contain the information necessary to invoke a drill-through target.

Use this information to construct a new [async](#) » [run\(objectPath, parameterValues, options\)](#) request to perform a drill-through operation.

This class

- inherits properties from the [bibus](#) » [asyncDetail](#) class

What's new

New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889

This class was added.

Properties

This class has the following properties.

action

Specifies the action to be performed by the [recipient](#).

This property

- is of type [bibus](#) » [drillThroughActionEnum](#)
 - is encoded as type `xs:string`
- is read-only

bookmarkText

Specifies the bookmark text to be used with the drill-through request.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`
- is read-only

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`
- is read-only

recipient

Specifies the recipient of the drill-through request.

This property

- is of type [bibus » drillThroughRecipientEnum](#)
is encoded as type `xs:string`
- is read-only

target

Specifies the object to be used as the target of the drill-through request.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`
- is read-only

targetURI

Specifies the external URI to be used with the drill-through request.

This property

- is of type `anyURI`
is encoded as type `xs:string`
- is read-only

asynchDetailDrillThroughTarget

Defines the class used to describe a [bibus » drillPath](#) and its target.

This class

- inherits properties from the [bibus » asynchDetail](#) class

What's new

New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889

This class was added.

Properties

This class has the following properties.

action

Specifies the action to be performed by the [recipients](#).

This property

- is of type [bibus » drillThroughActionEnum](#)
 - is encoded as type `xs:string`
- is read-only

drillPathName

Specifies the name of the drill path.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

drillPathScreenTip

Specifies the screen tip for the drill path.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

drillPathSearchPath

Specifies the drill path's search path.

This property

- is of type [bibus » searchPathSingleObject](#)
 - is encoded as type `tns:searchPathSingleObject`
- is read-only

recipients

Specifies the tools that may receive the drill-through request based on the [bibus » drillPath](#).

This property

- is an array of type [bibus » drillThroughRecipientEnum](#)
 - is encoded as type `tns:anyURIArray`
- is read-only

target

Specifies the drill path target's search path.

This property

- is of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObject`
- is read-only

targetAncestorNames

Specifies the names of the ancestors of the `target`.

If the `hidden` property of the `target` is `true`, the `targetAncestorNames` property will not contain any names.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`
- is read-only

targetHidden

Specifies whether the `target` is visible in the portal.

This property

- is of type `boolean`
is encoded as type `xs:boolean`
- is read-only

targetIconURI

Specifies the location of the icon for the `target`.

This property

- is of type `anyURI`
is encoded as type `xs:string`
- is read-only

targetName

Specifies the name of the `target`.

This property

- is of type `string`
is encoded as type `xs:string`
- is read-only

asynchDetailDrillThroughTargetURI

Defines the class used to describe a `bibus » drillPath` and its target URI.

This class

- inherits properties from the `bibus » asynchDetail` class

What's new

New in Version 8.4 — “Dynamic Filtering of Report Data” on page 1889

This class was added.

Properties

This class has the following properties.

drillPathName

Specifies the name of the drill path.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

targetURI

Specifies the drill path target URI.

This property

- is of type `anyURI`
 - is encoded as type `xs:string`
- is read-only

asyncDetailEventID

Defines the class that returns an eventID.

This class

- inherits properties from the [bibus » asyncDetail](#) class

Properties

This class has the following properties.

eventID

Specifies a unique ID for the task.

This property

- is of type `string`
 - is encoded as type `xs:string`
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase

asyncDetailEventRecord

Defines the class that contains details about a current or future event.

This class

- inherits properties from the [bibus » asyncDetail](#) class

What's new

New in Version 8.3 — [“Schedule Management” on page 1907](#)

This class was added.

Properties

This class has the following properties.

cancelledBy

Specifies the account used to cancel the task.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 10.1.0 – “Scheduling – Better Administrative Oversight” on page 1852

This property was added.

duration

Specifies the duration of the task execution. If the task is not running, this property will be NIL.

This property

- is of type `duration`
 - is encoded as type `xs:string`
- is read-only

eventID

Specifies a unique task ID.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase

history

Provides information about the task execution.

Properties inherited from [bibus » baseClass](#) class and [bibus » uiClass](#) class will not have values. The history object returned here is not available from the content store.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » history](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- must have at most 1 item

historyDetails

Contains additional information about running the task. The information varies based on the type of task.

Properties inherited from `bibus » baseClass` class and `bibus » uiClass` class will not have values. The objects returned here are not available from the content store.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » baseHistoryDetail`
 - is encoded as type `tns:baseClassArray`
- is read-only

priority

Specifies the schedule priority. Higher priority tasks execute before lower priority tasks. Priority values range from 1 to 5. 1 is the lowest priority; 5 is the highest.

This property

- is of type `int`
 - is encoded as type `xs:int`
- is read-only
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 5

New in Version 8.3 – “Schedule Priority” on page 1912

This property was added.

processID

Identifies the process executing the task.

This property

- is of type `int`
 - is encoded as type `xs:int`

New in Version 10.1.0 – “Scheduling – Better Administrative Oversight” on page 1852

This property was added.

requestedStartTime

Specifies the requested event start time, in Coordinated Universal Time (UTC).

If the scheduled task is delayed by calling either the `event » delayEventsFor(eventIDs, for)` method or the `event » delayEventsUntil(eventIDs, until)` method, then the value of this property is updated with the new calculated start time.

This property

- is of type `dateTime`
 - is encoded as type `xs:dateTime`
- is read-only

runnable

Specifies information about the task.

Only the `bibus » baseClass » name` property and `bibus » baseClass » searchPath` property properties will contain values.

This property

- is an array of type `bibus » baseClass`

has items that must be of class [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » exportDeployment](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » jobDefinition](#), [bibus » memo](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » queryServiceTask](#), [bibus » storedProcedureTask](#), or [bibus » webServiceTask](#)

is encoded as type `tns:baseClassArray`

- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

This property was extended to allow instances of the [bibus » queryServiceTask](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Human Task” on page 1862

This property was extended to allow instances of the [bibus » humanTask](#) class to be referenced by instances of this property.

scheduledStartTime

Specifies the originally scheduled start time for the event, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `xs:dateTime`
- is read-only

New in Version 10.1.0 – “Delaying Scheduled Tasks” on page 1852

This property was added.

Related information:

[delayEventsFor\(eventIDs, for\)](#) method

[delayEventsUntil\(eventIDs, until\)](#) method

[requestedStartTime](#) property

scheduleTriggerName

Specifies the value from [bibus » schedule » triggerName](#) property when the task was scheduled.

If the task was not run as a scheduled task, or the value of [bibus » schedule » type](#) property was not [trigger](#) then this property will be NIL.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

scheduleType

Specifies the value from [bibus » schedule » type](#) property when the task was scheduled.

If the task was not run as a scheduled task, then this property will be NIL.

This property

- is of type [bibus » scheduleTypeEnum](#)
 - is encoded as type `xs:string`
- is read-only

status

Specifies the event's run status.

This property

- is of type [bibus](#) » [runStatusEnum](#)
 - is encoded as type `xs:string`
- is read-only

suspendedBy

Specifies the account used to suspend the task.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 10.1.0 – “Scheduling – Better Administrative Oversight” on page 1852

This property was added.

tenantID

Supports multi-tenancy view in Cognos Administration portal.

New in Version 10.2.2 – Support for delegated tenant administration

This property was added.

user

Specifies information about the account used to run the task.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

asynchDetailIndexData

Reserved.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.3 – “Search – For Internal Use Only” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

creationTime

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

displayName

This property

- is of type `string`
is encoded as type `tns:multilingualStringArray`
- is multilingual

documentClass

This property

- is of type `bibus » classEnum`
is encoded as type `tns:classEnum`

externalGUID

This property

- is of type `bibus » guid`
is encoded as type `tns:guid`

locales

This property

- is an array of type `language`
is encoded as type `tns:languageArray`

modificationTime

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

sourceDocument

This property

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXML`

terms

This property

- is an array of type `bibus » indexTerm`
is encoded as type `tns:indexTermArray`

url

This property

- is of type `anyURI`

is encoded as type `xs:string`

asynchDetailIndexTerms

Reserved.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.3 – “[Search – For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

childTerms

This property

- is an array of type [bibus](#) » [indexTerm](#)
is encoded as type `tns:indexTermArray`

exampleTerms

This property

- is an array of type [bibus](#) » [indexTerm](#)
is encoded as type `tns:indexTermArray`

parentTerms

This property

- is an array of type [bibus](#) » [indexTerm](#)
is encoded as type `tns:indexTermArray`

siblingTerms

This property

- is an array of type [bibus](#) » [indexTerm](#)
is encoded as type `tns:indexTermArray`

asynchDetailMessages

Defines the class used to return messages with the response.

Note that information related to the failure of the request is returned using a SOAP fault. This class is used to return informational and warning messages related to the execution of the request.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

What's new

New in Version 8.4 — “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

messages

Contains the messages generated during the successful execution of the request.

This property

- is an array of type [bibus » faultDetail](#)
 - is encoded as type `tns: faultDetailArray`
- is read-only

asynchDetailMIMEAttachment

Defines the class used to return large volumes of data in an asynchronous conversation.

Data is returned inline (base64-encoded), or in a MIME attachment based on the value of the [attachmentEncoding](#) option. Data returned in a MIME attachment may also be compressed.

This class

- inherits properties from the [bibus » asynchDetail](#) class

What's new

New in Version 8.4 — “Lineage Metadata” on page 1891

This class was added.

Related information:

- [bibus » asynchOptionEnum » attachmentEncoding](#) value

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
 - is encoded as type `xs:base64Binary`

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`

is encoded as type `xs:integer`

- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `xs:string`
- can contain at least 255 characters

asynchDetailParameters

Defines the class that returns parameter definitions.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

Properties

This class has the following properties.

parameters

Contains the set of execution parameter definitions for this object.

This property

- is an array of type [bibus](#) » [baseParameter](#)
 - is encoded as type `tns:baseParameterArray`

asynchDetailParameterValues

Defines the class that returns parameter values in an asynchronous conversation.

This class

- inherits properties from the [bibus](#) » [asynchDetail](#) class

Properties

This class has the following properties.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)
 - is encoded as type `tns:parameterValueArray`
- is read-only

asynchDetailPromptPage

Defines the class used to return a prompt page.

Information about the parameters represented on the prompt page is also returned.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

parameters

Returns information about the parameters represented on the prompt page.

This property

- is an array of type [bibus » baseParameter](#)
 - is encoded as type `tns:baseParameterArray`
- is read-only
- must have at least 1 item

promptPage

Returns the prompt page.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchDetailReportMetadata

Defines the class that contains metadata responses from the report server.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

metadata

Contains the metadata response.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchDetailReportObject

Defines the class that contains a report object returned from the report service.

Information indicating whether the report specification was upgraded is also returned.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

report

Contains the report object.

This property

- is of type [bibus » authoredReport](#)
 - is encoded as type `tns:authoredReport`
- is read-only

upgraded

Specifies whether the report specification was upgraded to the current version. If the value specified for option [specificationFormat](#) caused a conversion to occur, the report specification was upgraded automatically.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- is read-only

asynchDetailReportOutput

Defines the class used to return requested report output.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

outputObjects

References the objects created in the content store to persist the report output.

The number and classes of objects referenced depends on the run options specified when the report is executed. The structure matches that returned by Content Manager in response to the [content » query\(searchPath, properties, sortBy, options\)](#) method, and includes the following:

- [bibus » baseClass » defaultName](#) property
- [bibus » baseClass » objectClass](#) property
- [bibus » baseClass » searchPath](#) property
- [bibus » baseClass » storeID](#) property

If the report output is saved in Content Manager by setting run option [saveAs](#) or [saveOutput](#), this property contains references to the [bibus » reportVersion](#) objects that were created to store the report output.

If the output is returned in the [bibus » asynchDetailReportOutput » outputPages](#) property, meaning none of the run options [email](#), [print](#), [saveAs](#), or [saveOutput](#) were specified, this property may contain a reference to a new [bibus » output](#) object, depending on the [outputEncapsulation](#) run option value specified. This object exists in the session temporary folder and is only accessible as long as the passport used during the request is valid.

The following table describes the objects returned, depending on the value of `outputEncapsulation`.

<i>Table 148. Returned objects based on outputEncapsulation value</i>	
Value	Response description
<code>none</code>	<p>The report is returned in the response.</p> <p>For <code>PDF</code>, <code>CSV</code>, and <code>XML</code>, no output object is created.</p> <p>For <code>HTML</code>, <code>XHTML</code>, or <code>HTMLFragment</code>, an output object is created if the report contains charts. The output object does not contain any actual data but is used as a container for the graphic objects that contain the charts.</p> <p>For <code>XLS</code>, an output object is created that contains no data. It is used as a container for page objects which contain the individual sheets in HTML format.</p>
<code>HTML</code>	<p>An <code>HTML</code> wrapper is returned in the response.</p> <p>For <code>PDF</code>, <code>CSV</code>, and <code>XML</code>, an output object containing the report is created. Since <code>HTML</code> and <code>XLS</code> are not wrapped in an HTML document, the response is the same as for <code>none</code>.</p>
All other values	<p>An <code>bibus » output</code> object is returned, regardless of the specified format.</p> <p>The <code>bibus » output</code> object contains the report and any child <code>bibus » graphic</code> or <code>bibus » page</code> objects that are part of the report.</p>

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » output` or `bibus » reportVersion`
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path

outputPages

Returns the output report pages.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArray`
- is read-only

asynchDetailReportStatus

Returns status information specific to `reportService` or `batchReportService`.

This class

- inherits properties from the `bibus » asynchDetail` class

Properties

This class has the following properties.

status

Returns the status for the request sent to the `reportService` or `batchReportService` for this asynchronous conversation.

This property

- is of type [bibus » asynchDetailReportStatusEnum](#)
is encoded as type `tns:asynchDetailReportStatusEnum`
- is read-only

asynchDetailReportValidation

Defines the class that contains report validation results.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

defects

Contains the set of defects detected during validation of a report or report specification. The defects can be query defects or layout defects. If a query defect is returned, the layout is not validated.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

maximumSeverity

Contains the severity of the most serious defect detected.

This property

- is of type [bibus » validateSeverityEnum](#)
is encoded as type `tns:validateSeverityEnum`

queryInfo

Contains information, constructed from the report specification, that is particular to the data source.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

asynchDetailROLAPDataSourceState

Returns state information on specific ROLAP data sources on a specific [queryService](#).

This class

- inherits properties from the [bibus » asynchDetail](#) class

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This class was added.

Properties

This class has the following properties.

dataSourceName

Returns the name of the ROLAP data source.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

state

Returns the state of the ROLAP data source.

This property

- is of type `bibus » rolapDataSourceStateEnum`
 - is encoded as type `xs:string`
- is read-only

asynchDetailROLAPMessages

Returns message information for dynamic cubes.

This class

- inherits properties from the `bibus » asynchDetail` class

What's new

New in Version 10.2.2 – Dynamic cube messages

This class was added.

Properties

This class has the following properties.

rolapMessages

Contains an array of dynamic cube messages.

This property

- is an array of type `bibus » rolapCubeMessages`
 - is encoded as type `tns:rolapCubeMessagesArray`

asynchDetailROLAPMetrics

Returns message information for dynamic cubes.

This class

- inherits properties from the `bibus » asynchDetail` class

What's new

New in Version 10.2.2 – Dynamic cube metrics

This class was added.

Properties

This class has the following properties.

rolapMetrics

Contains an array of dynamic cube metrics.

This property

- is an array of type [bibus » rolapCubeMetrics](#)
is encoded as type `tns:rolapCubeMetricsArray`

asynchDetailSelectionContext

Defines the class that contains selection context data.

A selection context specifies and describes the values that are to be used for filtering the drill-through target.

This class

- inherits properties from the [bibus » asynchDetail](#) class

What's new

New in Version 8.4 – “Drill-Through Improvements” on page 1889

This class was added.

Properties

This class has the following properties.

context

Contains the selection context data.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to MIME. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchDetailUnstructuredData

Reserved.

This class

- inherits properties from the [bibus » asynchDetail](#) class

Properties

This class has the following properties.

data

Reserved.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchOption

Defines the abstract base class for all asynchronous option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » asynchOptionBoolean](#)
- [bibus » asynchOptionEncoding](#)
- [bibus » asynchOptionInt](#)
- [bibus » asynchOptionSearchPathSingleObject](#)
- [bibus » asynchOptionSearchPathSingleObjectArray](#)
- [bibus » asynchOptionStringArray](#)

Properties

This class has the following properties.

name

Identifies the asynchronous option.

This property

- is of type [bibus » asynchOptionEnum](#)
is encoded as type `tns:asynchOptionEnum`

asynchOptionBoolean

Defines boolean values for the asynchronous options.

This class

- inherits properties from the [bibus » asynchOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the asynchronous option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

asynchOptionEncoding

Defines [bibus » encodingEnum](#) values for the asynchronous options.

This class

- inherits properties from the [bibus » asynchOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the asynchronous option.

This property

- is of type [bibus » encodingEnum](#)
is encoded as type `tns:encodingEnum`

asynchOptionInt

Defines integer values for the asynchronous options.

This class

- inherits properties from the [bibus » asynchOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the asynchronous option.

This property

- is of type `int`
is encoded as type `xs:int`

asynchOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the asynchronous options.

This class

- inherits properties from the [bibus » asynchOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the asynchronous option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

asynchOptionSearchPathSingleObjectArray

Defines [bibus](#) » [searchPathSingleObject](#) values for the asynchronous options.

This class

- inherits properties from the [bibus](#) » [asynchOption](#) class

What's new

New in Version 8.3 – “Task Retry” on page 1909

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the asynchronous option.

This property

- is an array of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

asynchOptionStringArray

Defines string values for the asynchronous options.

This class

- inherits properties from the [bibus](#) » [asynchOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the asynchronous option.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

asynchReply

Defines the properties of all asynchronous responses returned by a service.

References

Used by the following method return values:

- [asynch](#) » [run\(objectPath, parameterValues, options\)](#) » [result](#)
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) » [result](#)
- [asynch](#) » [wait\(conversation, parameterValues, options\)](#) » [result](#)
- [delivery](#) » [addNotification\(objectPath\)](#) » [result](#)
- [delivery](#) » [clearNotifications\(objectPath\)](#) » [result](#)
- [delivery](#) » [deleteAllNotifications\(\)](#) » [result](#)

- [delivery](#) » [deleteNotification\(objectPath\)](#) » [result](#)
- [delivery](#) » [queryNotification\(objectPath\)](#) » [result](#)
- [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#) » [result](#)
- [drillThrough](#) » [findDrillThroughPaths\(objectPath, parameterValues, options\)](#) » [result](#)
- [drillThrough](#) » [queryDrillPath\(objectPath, parameterValues, options\)](#) » [result](#)
- [indexTerm](#) » [addTermAssociation\(term, parameterValues, options\)](#) » [result](#)
- [indexTerm](#) » [deleteTermAssociation\(term, parameterValues, options\)](#) » [result](#)
- [indexUpdate](#) » [add\(objectPath, parameterValues, options\)](#) » [result](#)
- [indexUpdate](#) » [delete\(objectPath, parameterValues, options\)](#) » [result](#)
- [indexUpdate](#) » [get\(objectPath, parameterValues, options\)](#) » [result](#)
- [monitor](#) » [background\(conversation\)](#) » [result](#)
- [paging](#) » [currentPage\(conversation, parameterValues, options\)](#) » [result](#)
- [paging](#) » [firstPage\(conversation, parameterValues, options\)](#) » [result](#)
- [paging](#) » [lastPage\(conversation, parameterValues, options\)](#) » [result](#)
- [paging](#) » [nextPage\(conversation, parameterValues, options\)](#) » [result](#)
- [paging](#) » [previousPage\(conversation, parameterValues, options\)](#) » [result](#)
- [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) » [result](#)
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) » [result](#)
- [parameter](#) » [getParameters\(objectPath, parameterValues, options\)](#) » [result](#)
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#) » [result](#)
- [promptPaging](#) » [back\(conversation, parameterValues, options\)](#) » [result](#)
- [promptPaging](#) » [forward\(conversation, parameterValues, options\)](#) » [result](#)
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [deliver\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [drill\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [getContext\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [getObjectContext\(objectPath, parameterValues, options\)](#) » [result](#)
- [report](#) » [getOutput\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [lineage\(conversation, parameterValues, options\)](#) » [result](#)
- [report](#) » [query\(objectPath, parameterValues, options\)](#) » [result](#)
- [report](#) » [render\(conversation, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [clearCubeWorkloadLog\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [getCubeState\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [refreshCubeDataCache\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [refreshCubeMemberCache\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [refreshCubeSecurity\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [restartCubes\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [startCubes\(cubeNames, parameterValues, options\)](#) » [result](#)
- [rolapCubeAdministration](#) » [stopCubes\(cubeNames, parameterValues, options\)](#) » [result](#)
- [validate](#) » [validate\(objectPath, parameterValues, options\)](#) » [result](#)
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) » [result](#)

Properties

This class has the following properties.

details

Contains the detailed reply information from the service.

This property

- is an array of type [bibus](#) » [asynchDetail](#)
is encoded as type `tns:asynchDetailArray`
- is read-only

primaryRequest

Returns the request that initiated the asynchronous conversation. The request may be updated by the server to facilitate processing on a different server.

Normally this property is only returned when the `status` property is `complete` to improve performance and reduce network load. The option `alwaysIncludePrimaryRequest` can be used to force this value to be returned.

This property

- is of type [bibus](#) » [asynchRequest](#)
is encoded as type `tns:asynchRequest`
- is read-only

secondaryRequests

Contains the secondary requests that are available for processing as part of the asynchronous conversation.

This property

- is an array of type [bibus](#) » [asynchSecondaryRequest](#)
is encoded as type `tns:asynchSecondaryRequestArray`
- is read-only

status

Specifies the status of the last request sent to the server for the asynchronous conversation.

This property

- is of type [bibus](#) » [asynchReplyStatusEnum](#)
is encoded as type `tns:asynchReplyStatusEnum`
- is read-only

asynchRequest

Defines the properties of asynchronous requests processed by a service.

References

Used by the following properties:

- [bibus](#) » [asynchReply](#) » [primaryRequest](#)

Used by the following method parameters:

- [asynch](#) » [cancel\(conversation\)](#) » [conversation](#)
- [asynch](#) » [release\(conversation\)](#) » [conversation](#)

- [asynch](#) » [wait\(conversation, parameterValues, options\)](#) » [conversation](#)
- [monitor](#) » [background\(conversation\)](#) » [conversation](#)
- [paging](#) » [currentPage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [paging](#) » [firstPage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [paging](#) » [lastPage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [paging](#) » [nextPage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [paging](#) » [previousPage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [promptPaging](#) » [back\(conversation, parameterValues, options\)](#) » [conversation](#)
- [promptPaging](#) » [forward\(conversation, parameterValues, options\)](#) » [conversation](#)
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [deliver\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [drill\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [getContext\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [getOutput\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [lineage\(conversation, parameterValues, options\)](#) » [conversation](#)
- [report](#) » [render\(conversation, parameterValues, options\)](#) » [conversation](#)

Properties

This class has the following properties.

name

Specifies the name of the request.

This property

- is of type `string`
- is encoded as type `xs:string`
- is read-only

objectPath

Specifies the search path of the object in the content store used in the primary request that initiated the conversation.

This property

- is of type `bibus` » [searchPathSingleObject](#)
- is encoded as type `tns:searchPathSingleObject`
- is read-only

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type `bibus` » [option](#)
- is encoded as type `tns:optionArray`
- is read-only

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`
- is read-only

specification

Specifies the inline specification. This specification was used in the primary request that initiated the conversation.

This property

- is an array of type [bibus » asynchSpecification](#)
is encoded as type `tns:asynchSpecificationArray`
- is read-only
- must have at most 1 item

stateData

Specifies the internal run-time state information that the server requires with every secondary request.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`
- is read-only

asynchSecondaryRequest

Defines properties used to build a secondary request.

References

Used by the following properties:

- [bibus » asynchReply » secondaryRequests](#)

Properties

This class has the following properties.

name

Specifies the name of the secondary request.

This property

- is of type `string`
is encoded as type `xs:string`
- is read-only

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`
- is read-only

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`
- is read-only

asynchSpecification

Defines the properties of a service request.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » dataIntegrationServiceSpecification](#)
- [bibus » dataMovementServiceSpecification](#)
- [bibus » deliveryServiceSpecification](#)
- [bibus » dimensionManagementServiceSpecification](#)
- [bibus » eventManagementServiceSpecification](#)
- [bibus » indexSearchServiceSpecification](#)
- [bibus » indexUpdateServiceSpecification](#)
- [bibus » metadataServiceSpecification](#)
- [bibus » migrationServiceSpecification](#)
- [bibus » monitorServiceSpecification](#)
- [bibus » planningAdministrationConsoleServiceSpecification](#)
- [bibus » planningRuntimeServiceSpecification](#)
- [bibus » planningTaskServiceSpecification](#)
- [bibus » powerPlayServiceSpecification](#)
- [bibus » relationalMetadataServiceSpecification](#)
- [bibus » reportDataServiceSpecification](#)
- [bibus » reportServiceDrillThroughSpecification](#)
- [bibus » reportServiceMetadataSpecification](#)
- [bibus » reportServiceSpecification](#)

References

Used by the following properties:

- [bibus](#) » [asynchDetailAsynchSpecification](#) » [specification](#)
- [bibus](#) » [asynchRequest](#) » [specification](#)

Used by the following method parameters:

- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) » [specification](#)
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) » [specification](#)
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#) » [specification](#)
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) » [specification](#)

Properties

This class has the following properties.

value

Defines the request specification for the service.

This property

- is of type [bibus](#) » [specification](#)

is encoded as type `tns:specification`

auditLevelEnumProp

Defines the simple property class for the [bibus](#) » [auditLevelEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [adaptiveAnalyticsService](#) » [aasAuditLevel](#)
- [bibus](#) » [configuration](#) » [aasAuditLevel](#)
- [bibus](#) » [configurationFolder](#) » [aasAuditLevel](#)
- [bibus](#) » [dispatcher](#) » [aasAuditLevel](#)
- [bibus](#) » [annotationService](#) » [ansAuditLevel](#)
- [bibus](#) » [configuration](#) » [ansAuditLevel](#)
- [bibus](#) » [configurationFolder](#) » [ansAuditLevel](#)
- [bibus](#) » [dispatcher](#) » [ansAuditLevel](#)
- [bibus](#) » [agentService](#) » [asAuditLevel](#)
- [bibus](#) » [configuration](#) » [asAuditLevel](#)
- [bibus](#) » [configurationFolder](#) » [asAuditLevel](#)
- [bibus](#) » [dispatcher](#) » [asAuditLevel](#)
- [bibus](#) » [batchReportService](#) » [brsAuditLevel](#)
- [bibus](#) » [configuration](#) » [brsAuditLevel](#)
- [bibus](#) » [configurationFolder](#) » [brsAuditLevel](#)
- [bibus](#) » [dispatcher](#) » [brsAuditLevel](#)

- [bibus » configuration » cmcsAuditLevel](#)
- [bibus » configurationFolder » cmcsAuditLevel](#)
- [bibus » contentManagerCacheService » cmcsAuditLevel](#)
- [bibus » dispatcher » cmcsAuditLevel](#)
- [bibus » configuration » cmsAuditLevel](#)
- [bibus » configurationFolder » cmsAuditLevel](#)
- [bibus » contentManagerService » cmsAuditLevel](#)
- [bibus » dispatcher » cmsAuditLevel](#)
- [bibus » configuration » dasAuditLevel](#)
- [bibus » configurationFolder » dasAuditLevel](#)
- [bibus » dataAdvisorService » dasAuditLevel](#)
- [bibus » dispatcher » dasAuditLevel](#)
- [bibus » configuration » dimsAuditLevel](#)
- [bibus » configurationFolder » dimsAuditLevel](#)
- [bibus » dimensionManagementService » dimsAuditLevel](#)
- [bibus » dispatcher » dimsAuditLevel](#)
- [bibus » configuration » disAuditLevel](#)
- [bibus » configurationFolder » disAuditLevel](#)
- [bibus » dataIntegrationService » disAuditLevel](#)
- [bibus » dispatcher » disAuditLevel](#)
- [bibus » configuration » dispatcherAuditLevel](#)
- [bibus » configurationFolder » dispatcherAuditLevel](#)
- [bibus » dispatcher » dispatcherAuditLevel](#)
- [bibus » configuration » dmsAuditLevel](#)
- [bibus » configurationFolder » dmsAuditLevel](#)
- [bibus » dataMovementService » dmsAuditLevel](#)
- [bibus » dispatcher » dmsAuditLevel](#)
- [bibus » configuration » dsAuditLevel](#)
- [bibus » configurationFolder » dsAuditLevel](#)
- [bibus » deliveryService » dsAuditLevel](#)
- [bibus » dispatcher » dsAuditLevel](#)
- [bibus » configuration » emsAuditLevel](#)
- [bibus » configurationFolder » emsAuditLevel](#)
- [bibus » dispatcher » emsAuditLevel](#)
- [bibus » eventManagementService » emsAuditLevel](#)
- [bibus » configuration » evsAuditLevel](#)
- [bibus » configurationFolder » evsAuditLevel](#)
- [bibus » dispatcher » evsAuditLevel](#)
- [bibus » EVService » evsAuditLevel](#)
- [bibus » configuration » gsAuditLevel](#)
- [bibus » configurationFolder » gsAuditLevel](#)
- [bibus » dispatcher » gsAuditLevel](#)
- [bibus » graphicsService » gsAuditLevel](#)

- [bibus » configuration » htsAuditLevel](#)
- [bibus » configurationFolder » htsAuditLevel](#)
- [bibus » dispatcher » htsAuditLevel](#)
- [bibus » humanTaskService » htsAuditLevel](#)
- [bibus » configuration » idsAuditLevel](#)
- [bibus » configurationFolder » idsAuditLevel](#)
- [bibus » dispatcher » idsAuditLevel](#)
- [bibus » indexDataService » idsAuditLevel](#)
- [bibus » configuration » idVizAuditLevel](#)
- [bibus » configurationFolder » idVizAuditLevel](#)
- [bibus » dispatcher » idVizAuditLevel](#)
- [bibus » idVizService » idVizAuditLevel](#)
- [bibus » configuration » issAuditLevel](#)
- [bibus » configurationFolder » issAuditLevel](#)
- [bibus » dispatcher » issAuditLevel](#)
- [bibus » indexSearchService » issAuditLevel](#)
- [bibus » configuration » iusAuditLevel](#)
- [bibus » configurationFolder » iusAuditLevel](#)
- [bibus » dispatcher » iusAuditLevel](#)
- [bibus » indexUpdateService » iusAuditLevel](#)
- [bibus » configuration » jsAuditLevel](#)
- [bibus » configurationFolder » jsAuditLevel](#)
- [bibus » dispatcher » jsAuditLevel](#)
- [bibus » jobService » jsAuditLevel](#)
- [bibus » configuration » mbsAuditLevel](#)
- [bibus » configurationFolder » mbsAuditLevel](#)
- [bibus » dispatcher » mbsAuditLevel](#)
- [bibus » mobileService » mbsAuditLevel](#)
- [bibus » configuration » mdsAuditLevel](#)
- [bibus » configurationFolder » mdsAuditLevel](#)
- [bibus » dispatcher » mdsAuditLevel](#)
- [bibus » metadataService » mdsAuditLevel](#)
- [bibus » configuration » misAuditLevel](#)
- [bibus » configurationFolder » misAuditLevel](#)
- [bibus » dispatcher » misAuditLevel](#)
- [bibus » migrationService » misAuditLevel](#)
- [bibus » configuration » mmsAuditLevel](#)
- [bibus » configurationFolder » mmsAuditLevel](#)
- [bibus » dispatcher » mmsAuditLevel](#)
- [bibus » metricsManagerService » mmsAuditLevel](#)
- [bibus » configuration » msAuditLevel](#)
- [bibus » configurationFolder » msAuditLevel](#)
- [bibus » dispatcher » msAuditLevel](#)

- [bibus » monitorService » msAuditLevel](#)
- [bibus » configuration » pacsAuditLevel](#)
- [bibus » configurationFolder » pacsAuditLevel](#)
- [bibus » dispatcher » pacsAuditLevel](#)
- [bibus » planningAdministrationConsoleService » pacsAuditLevel](#)
- [bibus » configuration » pdsAuditLevel](#)
- [bibus » configurationFolder » pdsAuditLevel](#)
- [bibus » dispatcher » pdsAuditLevel](#)
- [bibus » planningDataService » pdsAuditLevel](#)
- [bibus » configuration » ppsAuditLevel](#)
- [bibus » configurationFolder » ppsAuditLevel](#)
- [bibus » dispatcher » ppsAuditLevel](#)
- [bibus » powerPlayService » ppsAuditLevel](#)
- [bibus » configuration » prsAuditLevel](#)
- [bibus » configurationFolder » prsAuditLevel](#)
- [bibus » dispatcher » prsAuditLevel](#)
- [bibus » planningRuntimeService » prsAuditLevel](#)
- [bibus » configuration » psAuditLevel](#)
- [bibus » configurationFolder » psAuditLevel](#)
- [bibus » dispatcher » psAuditLevel](#)
- [bibus » presentationService » psAuditLevel](#)
- [bibus » configuration » ptsAuditLevel](#)
- [bibus » configurationFolder » ptsAuditLevel](#)
- [bibus » dispatcher » ptsAuditLevel](#)
- [bibus » planningTaskService » ptsAuditLevel](#)
- [bibus » configuration » qsAuditLevel](#)
- [bibus » configurationFolder » qsAuditLevel](#)
- [bibus » dispatcher » qsAuditLevel](#)
- [bibus » queryService » qsAuditLevel](#)
- [bibus » configuration » rdsAuditLevel](#)
- [bibus » configurationFolder » rdsAuditLevel](#)
- [bibus » dispatcher » rdsAuditLevel](#)
- [bibus » reportDataService » rdsAuditLevel](#)
- [bibus » configuration » reposAuditLevel](#)
- [bibus » configurationFolder » reposAuditLevel](#)
- [bibus » dispatcher » reposAuditLevel](#)
- [bibus » repositoryService » reposAuditLevel](#)
- [bibus » configuration » rmdsAuditLevel](#)
- [bibus » configurationFolder » rmdsAuditLevel](#)
- [bibus » dispatcher » rmdsAuditLevel](#)
- [bibus » relationalMetadataService » rmdsAuditLevel](#)
- [bibus » configuration » rsAuditLevel](#)
- [bibus » configurationFolder » rsAuditLevel](#)

- [bibus » dispatcher » rsAuditLevel](#)
- [bibus » reportService » rsAuditLevel](#)
- [bibus » configuration » saCAMAuditLevel](#)
- [bibus » configurationFolder » saCAMAuditLevel](#)
- [bibus » dispatcher » saCAMAuditLevel](#)
- [bibus » saCAMService » saCAMAuditLevel](#)
- [bibus » configuration » ssAuditLevel](#)
- [bibus » configurationFolder » ssAuditLevel](#)
- [bibus » dispatcher » ssAuditLevel](#)
- [bibus » systemService » ssAuditLevel](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnum`

authoredAgentDefinition

Defines an authored agent as a series of tasks and defines parameters for those tasks.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » baseAgentDefinition](#) class

Derived Classes

- [bibus » agentDefinition](#)
- [bibus » reportDataServiceAgentDefinition](#)

Container Information

Contains instances of the following classes

- [bibus » agentTaskDefinition](#)
- [bibus » baseRSSTask](#)
- [bibus » humanTask](#)
- [bibus » memo](#)
- [bibus » storedProcedureTask](#)
- [bibus » webServiceTask](#)

What's new

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This class was added.

Properties

This class has the following properties.

eventKey

Specifies the query items in the contained report that uniquely identify rows in the resulting data set. These items are used to determine the set of events to be processed by the agent.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`
- must have at least 1 item

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property was added. It replaces the [bibus » agentDefinition » eventKey](#) property.

items

Contains the set of task objects of the agent. The individual task objects cannot be managed independently and are not accessible via the portal.

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property was added. It replaces the [bibus » agentDefinition » items](#) property.

New in Version 10.1.0 – “Human Task” on page 1862

This property was extended to allow instances of the [bibus » humanTask](#) class to be contained by instances of this property.

sequencing

Specifies whether the agent steps can run at the same time or one after the other.

This property

- is of type [bibus » sequencingEnum](#)
 - is encoded as type `tns:nmtokenProp`

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property was added. It replaces the [bibus » agentDefinition » sequencing](#) property.

tasks

Specifies the tasks contained by the agent definition.

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This property was added. It replaces the [bibus » agentDefinition » tasks](#) property.

authoredPowerPlay8Report

Defines the abstract class for authored PowerPlay reports.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » basePowerPlay8Report](#) class

Derived Classes

- [bibus » powerPlay8Report](#)

References

Used by the following properties:

- [bibus](#) » [powerPlay8ReportView](#) » [base](#)

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [model](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [package](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

powerPlay8Configuration

Specifies the PowerPlay configuration data for this object.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

specification

Specifies the specification for the task.

This property

- is of type [bibus » powerPlayServiceSpecification](#)
is encoded as type `tns:powerPlayServiceSpecificationProp`

authoredReport

Defines the abstract class for authored reports.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » baseReport](#) class

Derived Classes

- [bibus » analysis](#)
- [bibus » interactiveReport](#)
- [bibus » query](#)
- [bibus » report](#)
- [bibus » reportTemplate](#)

References

Used by the following properties:

- [bibus » reportView » base](#)
- [bibus » asyncDetailReportObject » report](#)

Used by the following method parameters:

- [report » add\(parentPath, object, options\) » object](#)
- [report » update\(object, options\) » object](#)

Used by the following method return values:

- [report » add\(parentPath, object, options\) » result](#)
- [report » update\(object, options\) » result](#)

Properties

This class has the following properties.

canBurst

Specifies whether the report output can be burst.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

defaultTriggerDescription

Specifies the description of the external occurrence named by the `triggerName` property that best matches the description expressed in the user's preferred language. This is the content language specified by the `bibus » account » contentLocale` property. The user selects this language in the user preferences.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 32767 characters

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » model`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » package`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

paths

Contains the navigation paths. A navigation path is used to drill through from a source to a target.

For example, a navigation path can specify how to get from one report to another to allow a user to access related, or more detailed, information.

This property

- is an array of type `bibus » navigationPath`

is encoded as type `tns:navigationPathArrayProp`

queryMode

Reserved.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

New in Version 10.1.0 – “Query Modes” on page 1865

This property was added.

runWithOwnerCapabilities

Specifies whether the security context of the owner is used to perform specific capability checks when running a task.

For example, a report may contain user-defined SQL or HTML items. Running a report containing these items requires the `canUseUserDefinedSQL` or `canUseHTML` capabilities, that are normally assigned to report authors. When you set this property to `true`, other employees in the company may be able to run this report even if they do not have the required capabilities.

When this property is `true`, IBM Cognos uses the report owner's security context to perform certain user capability checks. This differs from `bibus » baseReport » runAsOwner` property in that it includes only the owner's capabilities and excludes other aspects of the owner's security context, such as access to data sources.

In the preceding example, the report owner sets this property to `true` to allow other employees in the company to run the report. This specifies that IBM Cognos uses the report owner's security context to determine whether user defined SQL can be used. Consumers of the report might not have the capability to include user defined SQL in their own reports, but when this property is `true`, the report owner's security context allows the use of user defined SQL.

A user must still have sufficient permissions for the target object in order to run it. For more information, see `asynch » run(objectPath, parameterValues, options)` method.

Other activities may also use this property to provide access to user capabilities. For example, `parameter » collectParameterValues(objectPath, parameterValues, options)` method collects all the parameters that must be specified before a report can run.

For more information about which owner capabilities are used when this property is `true`, see the **User Capability Details** table in the `bibus » userCapabilityEnum` enumeration set description.

If a request is sent to set this property to `true`, the `contentManagerService` service requires that the security context making the request (that is, accounts, groups and active roles) has either `read` or `execute` permissions on the credential of the account identified by the `owner` property of the object. This check is performed in addition to the normal permission checks for updating a property. If the security context does not have the appropriate permissions, the request fails.

If consumers of a report require access to privileged data to run the report, in addition to capabilities of the owner, use the `bibus » baseReport » runAsOwner` property instead.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

New in Version 10.1.0 – “Run with Owner Capabilities” on page 1871

This property was added.

New in Version 10.1.1 – “Run with Owner Capabilities” on page 1846

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

specification

Specifies the information required to execute the report. This property is the definition for the report.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`

triggerDescription

Specifies the description of the external occurrence named by the [triggerName](#) property.

This property

- is of type `string`
 - is encoded as type `tns:multilingualStringProp`
- can contain at least 32767 characters
- is multilingual

triggerName

Specifies the name of the external occurrence that initiates execution of the report.

When set, the value of this property is also stored in the [triggerName](#) property of the [bibus » schedule](#) object contained within the object (if it exists), as well as any [bibus » schedule](#) objects contained within any [bibus » reportView](#) objects associated with the [bibus » authoredReport](#) object (see the [bibus » reportView » base](#) property).

Note that if this property has not been set, the [bibus » schedule » triggerName](#) property may be set independently.

If you require multiple external occurrences to trigger the execution of this report, you must manage this in the application you use to call the [event » trigger\(triggerName\)](#) method.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- is searchable

base64BinaryMIMEProp

Defines the simple MIME property class for the [base64Binary](#).

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » cacheOutput » data](#)
- [bibus » dashboard » data](#)
- [bibus » documentContent » data](#)
- [bibus » graphic » data](#)
- [bibus » historyDetailDataMovementService » data](#)
- [bibus » output » data](#)
- [bibus » page » data](#)

- [bibus » visualization » data](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

The value may be placed in a MIME attachment.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

baseAgentDefinition

Defines the set of common properties shared by all agent classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » uiClass](#) class

Derived Classes

- [bibus » agentDefinitionView](#)
- [bibus » authoredAgentDefinition](#)

References

Used by the following properties:

- [bibus » agentState » definition](#)
- [bibus » asynchDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » shortcut » target](#)
- [bibus » agentTaskDefinition » taskObject](#)

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)

Properties

This class has the following properties.

allowNotification

Specifies whether the object allows alerts.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `true`

New in Version 8.3 – “Default Value Change” on page 1919

The default value for this property was changed to `true`.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » account`, `bibus » contact`, `bibus » distributionList`, `bibus » group`, or `bibus » role`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

defaultPortalAction

Specifies the default action for an object in a portal.

When users click the hyperlinked name in the portal, the default action specified, such as view, run, or open for editing, is performed.

This property

- is of type `bibus » baseAgentDefinitionActionEnum`
 - is encoded as type `tns:baseAgentDefinitionActionEnumProp`
- has a default value of `viewEventList`

New in Version 8.3 – “Agent Default Portal Action” on page 1901

This property was added.

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the `retentions` property to manage contained instances of `bibus » history`

mostRecentEventList

Refers to the most recently created event list for this agent.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [output](#)
is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 8.3 – “Agent Default Portal Action” on page 1901

This property was added.

notificationList

Refers to the set of accounts that receive alerts when the object runs.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [account](#)
is encoded as type `tns:baseClassArrayProp`
- cannot be processed directly using the IBM Cognos Software Development Kit
- refers to other objects in the content store using an id-based search path

New in Version 8.3 – “Notification List” on page 1920

This property cannot be accessed using the BI Bus API.

SDK clients that require access to this property must use the `deliveryService` methods `delivery` » `addNotification(objectPath)`, `delivery` » `clearNotifications(objectPath)`, `delivery` » `deleteAllNotifications()`, `delivery` » `deleteNotification(objectPath)` and `delivery` » `queryNotification(objectPath)`.

notificationListIsEmpty

Specifies whether the [notificationList](#) property is empty.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- is read-only

New in Version 8.3 – “Report Email Alerts” on page 1900

This property was added.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

ownerPassport

When [bibus » baseAgentDefinition » runAsOwner](#) property is true, this property specifies a CAM cookie created using the credentials associated with the object's owner. The passport and session created during the log on process are used to perform the requested task.

Once the task has completed, a logoff request is issued to invalidate the passport and terminate the session.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- cannot be processed directly using the IBM Cognos Software Development Kit

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic now accurately describes the contents of this property.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type `bibus » retentionRule`
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

runAsOwner

Specifies whether the security context of the owner is used to perform specific capability checks and to access data when running a task.

For example, the Human Resources department publishes a report that contains information about salary ranges across the company. The nature of the report requires the use of user defined SQL (controlled by `canUseUserDefinedSQL`). Since this report does not disclose individual salary information, it is not deemed to be confidential. However, the data required to run the report is only available to certain members of the Human Resources department. One of these employees is the owner of the report.

When this property is `true`, IBM Cognos uses the report owner's security context to access the data necessary to produce the report, as well as to perform certain user capability checks.

In the preceding example, the report owner sets this property to `true` to allow other employees in the company to run the report. Setting this property to `true` specifies that IBM Cognos uses the report owner's security context to access the data that is normally unavailable to the majority of employees. In addition, the capability check to determine whether user defined SQL can be used is performed using the report owner's security context. Consumers of the report may not be permitted access to the data or the ability to include user defined SQL in their own reports, but when this property is `true`,

the report owner's security context is used to access the data and to allow the use of user defined SQL.

A user must still have sufficient permissions for the target object in order to run it. For more information, see [asynch » run\(objectPath, parameterValues, options\)](#) method.

Other activities may also use this property to provide access to privileged data or user capabilities. For example, [parameter » collectParameterValues\(objectPath, parameterValues, options\)](#) method collects all the parameters that must be specified before a report can run.

For more information about which owner capabilities are used when this property is `true`, see the **User Capability Details** table in the [bibus » userCapabilityEnum](#) enumeration set description.

If a request is sent to set this property to `true`, the [contentManagerService](#) service requires that the security context making the request (that is, accounts, groups and active roles) has either `read` or `execute` permissions on the credential of the account identified by the [owner](#) of the object. This check is performed in addition to the normal permission checks for updating a property. If the security context does not have the appropriate permissions, the request fails.

If consumers of a report only require one of the owner's capabilities and do not require access to privileged data to run the report, use the [runWithOwnerCapabilities](#) property instead.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

baseAgentDefinitionActionEnumProp

Defines the simple property class for the [bibus » baseAgentDefinitionActionEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » baseAgentDefinition » defaultPortalAction](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » baseAgentDefinitionActionEnum](#)
 - is encoded as type `tns:baseAgentDefinitionActionEnum`

baseClass

Instances of this class are objects that can be manipulated with the set of methods that manage the content store.

Every `bibus » baseClass` class property has a `name`. A name can be specified in multiple locales. The name in the user's locale can be manipulated using `defaultName`.

The property `searchPath` identifies objects. This property is usually constructed from the name and class of the object, and the names and classes of all containing objects, although this is not always the case. This property is used to determine the set of objects that are returned in response to calling query methods, including the `content » query(searchPath, properties, sortBy, options)` method. This property is also used to identify the set of objects that are affected when other methods that manage the content store are called. The `storeId` defines a `searchPath` form that can be embedded in a URL.

The following properties provide basic information about the object that is used by Content Manager when managing the content store, or by IBM Cognos Connection when constructing portal pages:

- `creationTime`
- `modificationTime`
- `objectClass`
- `usage`
- `version`

Information about other objects that contain the object, as well as whether the object is a container for other `baseClass` objects, is provided in these properties:

- `ancestors`
- `hasChildren`
- `parent`

Certain properties are used to control how the object can be manipulated by users of the content store, as well as by other components:

- `disabled`
- `owner`
- `permissions`
- `policies`

The property `position` records the position of the object in the response returned by query methods like the `content » query(searchPath, properties, sortBy, options)` method.

Many of the objects in the BI Bus API are instances of a class that is derived from `baseClass`, either directly or indirectly. For example, instances of the `bibus » graphic` class are used to store images used by report outputs in the content store. The `bibus » report` class is an indirect derivation of `baseClass`. It is derived from the `bibus » authoredReport` class, which is derived from the `bibus » baseReport` class, which is derived from the `bibus » uiClass` class, which, finally, is derived from `baseClass`.

You cannot create an instance of `baseClass` because it is an abstract class. You can only create instances of derived classes.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- `bibus » agentOutputHotList`
- `bibus » agentState`
- `bibus » agentTaskState`

- [bibus](#) » [baseHistoryDetail](#)
- [bibus](#) » [cacheOutput](#)
- [bibus](#) » [documentContent](#)
- [bibus](#) » [graphic](#)
- [bibus](#) » [installedComponent](#)
- [bibus](#) » [modelView](#)
- [bibus](#) » [nil](#)
- [bibus](#) » [output](#)
- [bibus](#) » [page](#)
- [bibus](#) » [pageletInstance](#)
- [bibus](#) » [portletInstance](#)
- [bibus](#) » [runTimeState](#)
- [bibus](#) » [session](#)
- [bibus](#) » [SQL](#)
- [bibus](#) » [systemMetricThresholds](#)
- [bibus](#) » [transientStateFolder](#)
- [bibus](#) » [uiClass](#)

References

Used by the following properties:

- [bibus](#) » [agentDefinitionView](#) » [base](#)
- [bibus](#) » [dataMovementTaskAlias](#) » [base](#)
- [bibus](#) » [personalization](#) » [base](#)
- [bibus](#) » [portalSkin](#) » [base](#)
- [bibus](#) » [powerPlay8ReportView](#) » [base](#)
- [bibus](#) » [reportView](#) » [base](#)
- [bibus](#) » [asynchDetailEventRecord](#) » [cancelledBy](#)
- [bibus](#) » [baseRSSTask](#) » [channel](#)
- [bibus](#) » [baseDataIntegrationTask](#) » [connections](#)
- [bibus](#) » [dataSourceSignon](#) » [consumers](#)
- [bibus](#) » [adminFolder](#) » [contact](#)
- [bibus](#) » [aliasRoot](#) » [contact](#)
- [bibus](#) » [annotation](#) » [contact](#)
- [bibus](#) » [annotationFolder](#) » [contact](#)
- [bibus](#) » [archiveLocation](#) » [contact](#)
- [bibus](#) » [baseAgentDefinition](#) » [contact](#)
- [bibus](#) » [baseDataIntegrationTask](#) » [contact](#)
- [bibus](#) » [baseDataMovementTask](#) » [contact](#)
- [bibus](#) » [basePowerPlay8Report](#) » [contact](#)
- [bibus](#) » [basePowerPlayClass](#) » [contact](#)
- [bibus](#) » [baseReport](#) » [contact](#)
- [bibus](#) » [baseROLAPDataSource](#) » [contact](#)
- [bibus](#) » [contentTask](#) » [contact](#)

- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » schedule » credential](#)
- [bibus » dataSourceConnection » credentialNamespaces](#)
- [bibus » baseROLAPDataSource » dataSourceAccessAccount](#)
- [bibus » packageConfiguration » defaultAnalysis](#)
- [bibus » agentState » definition](#)

- [bibus » agentTaskState » definition](#)
- [bibus » deploymentDetail » deployedObject](#)
- [bibus » group » distributionMembers](#)
- [bibus » role » distributionMembers](#)
- [bibus » baseReport » executionPageDefinition](#)
- [bibus » asynchDetailEventRecord » history](#)
- [bibus » eventRecord » history](#)
- [bibus » asynchDetailEventRecord » historyDetails](#)
- [bibus » session » identity](#)
- [bibus » configuration » jmxProxyHostDispatchers](#)
- [bibus » shortcutRSSTask » link](#)
- [bibus » historyDetailRelatedReports » linkPaths](#)
- [bibus » distributionList » members](#)
- [bibus » group » members](#)
- [bibus » role » members](#)
- [bibus » authoredPowerPlay8Report » metadataModel](#)
- [bibus » authoredReport » metadataModel](#)
- [bibus » baseDataIntegrationTask » metadataModel](#)
- [bibus » reportCache » metadataModel](#)
- [bibus » reportVersion » metadataModel](#)
- [bibus » storedProcedureTask » metadataModel](#)
- [bibus » authoredPowerPlay8Report » metadataModelPackage](#)
- [bibus » authoredReport » metadataModelPackage](#)
- [bibus » baseDataIntegrationTask » metadataModelPackage](#)
- [bibus » reportCache » metadataModelPackage](#)
- [bibus » reportVersion » metadataModelPackage](#)
- [bibus » storedProcedureTask » metadataModelPackage](#)
- [bibus » historyDetailMigrationService » migratedObject](#)
- [bibus » baseAgentDefinition » mostRecentEventList](#)
- [bibus » baseAgentDefinition » notificationList](#)
- [bibus » baseReport » notificationList](#)
- [bibus » deploymentReference » objects](#)
- [bibus » historyDetailAgentService » output](#)
- [bibus » historyDetailReportService » output](#)
- [bibus » asynchDetailReportOutput » outputObjects](#)
- [bibus » annotation » outputs](#)
- [bibus » baseClass » owner](#)
- [bibus » account » page](#)
- [bibus » pageletInstance » pagelet](#)
- [bibus » baseClass » parent](#)
- [bibus » schedule » periodicalProducer](#)
- [bibus » account » portalPages](#)
- [bibus » portletInstance » portlet](#)

- [bibus](#) » [periodical](#) » [producer](#)
- [bibus](#) » [queryReply](#) » [queryResult](#)
- [bibus](#) » [documentContent](#) » [recipients](#)
- [bibus](#) » [output](#) » [recipients](#)
- [bibus](#) » [agentTaskState](#) » [related](#)
- [bibus](#) » [historyDetailRelatedHistory](#) » [related](#)
- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [group](#) » [securityMembers](#)
- [bibus](#) » [role](#) » [securityMembers](#)
- [bibus](#) » [policy](#) » [securityObject](#)
- [bibus](#) » [userCapabilityPolicy](#) » [securityObject](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)
- [bibus](#) » [asynchDetailEventRecord](#) » [suspendedBy](#)
- [bibus](#) » [drillPath](#) » [target](#)
- [bibus](#) » [navigationPath](#) » [target](#)
- [bibus](#) » [shortcut](#) » [target](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)
- [bibus](#) » [asynchDetailEventRecord](#) » [user](#)
- [bibus](#) » [history](#) » [user](#)
- [bibus](#) » [baseClassArrayProp](#) » [value](#)
- [bibus](#) » [baseClassProp](#) » [value](#)

Used by the following method parameters:

- [content](#) » [add\(parentPath, objects, options\)](#) » [objects](#)
- [content](#) » [addAnnotations\(containerPath, objects, options\)](#) » [objects](#)
- [content](#) » [copy\(objects, targetPath, options\)](#) » [objects](#)
- [content](#) » [copyRename\(objects, targetPath, newNames, options\)](#) » [objects](#)
- [content](#) » [delete\(objects, options\)](#) » [objects](#)
- [content](#) » [move\(objects, targetPath, options\)](#) » [objects](#)
- [content](#) » [moveRename\(objects, targetPath, newNames, options\)](#) » [objects](#)
- [content](#) » [update\(objects, options\)](#) » [objects](#)

Used by the following method return values:

- [content](#) » [add\(parentPath, objects, options\)](#) » [result](#)
- [content](#) » [addAnnotations\(containerPath, objects, options\)](#) » [result](#)
- [content](#) » [copy\(objects, targetPath, options\)](#) » [result](#)
- [content](#) » [copyRename\(objects, targetPath, newNames, options\)](#) » [result](#)
- [content](#) » [move\(objects, targetPath, options\)](#) » [result](#)
- [content](#) » [moveRename\(objects, targetPath, newNames, options\)](#) » [result](#)
- [content](#) » [query\(searchPath, properties, sortBy, options\)](#) » [result](#)
- [content](#) » [update\(objects, options\)](#) » [result](#)

Related information:

- [content](#) method set

Properties

This class has the following properties.

ancestors

Contains information about the ancestors of the object. The parent of the object is the last element, the grandparent is the second-to-last element, and so on.

This property

- is an array of type [bibus](#) » [ancestorInfo](#)
is encoded as type `tns:ancestorInfoArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

creationTime

Specifies the time that the object was created, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

defaultName

Specifies the name of the object that best matches the name expressed in the user's preferred language. This is the content language specified by the [bibus](#) » [account](#) » [contentLocale](#) property. The user selects this language in the user preferences.

Any Unicode character is allowed in an object name, with the exception of tabs, carriage returns, and line feeds. Content Manager trims leading and trailing spaces, and reduces two or more consecutive internal spaces to one. Names consisting only of spaces are reduced to empty strings. We recommend that you do not use only spaces for object names.

This property

- is of type `token`
is encoded as type `tns:tokenProp`
- can contain at least 255 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

disabled

Specifies whether a user must have `write` or `setPolicy` permission to be granted `traverse`, `read`, or `execute` permission on the object.

Set to `true` to deny access to users when a maintenance operation is in progress.

For example, when a user tries to execute the object, the policy of the object is checked to ensure that the user has `execute` permission.

If the user does not have `execute` permission, the user is not allowed to execute the object.

If the user has `execute` permission and the `disabled` property is set to `true`, Content Manager checks to ensure that the user has either `write` or `setPolicy` access to the object. If the user does not have either of these permissions, the user is not allowed to execute the object.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- can be acquired from a containing object
- can be retrieved without requiring `read` permission on the object; only `traverse` permission on the full path to the containing object is required

hasChildren

Specifies whether the object contains other objects. Set to `true` if the object contains other objects.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only
- can be retrieved without requiring `read` permission on the object; only `traverse` permission on the full path to the containing object is required

modificationTime

Specifies the time that the object was last modified, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable
- can be retrieved without requiring `read` permission on the object; only `traverse` permission on the full path to the containing object is required

name

Specifies the name of the object.

Comparison of object names is case-insensitive. However, when Content Manager returns an object name, it has the same case as when the object was created.

Any Unicode character is allowed in an object name, with the exception of tabs, carriage returns, and line feeds. Content Manager trims leading and trailing spaces, and reduces two or more consecutive internal spaces to one. Names consisting only of spaces are reduced to empty strings. We recommend that you do not use only spaces for object names.

The name of an object must be unique among its siblings across all locales. Any given object may have the same name in different locales.

When you create an object, you do not have to specify a name. If you do not specify a name, Content Manager generates a name using a string that represents the current date and time, in a format that is a subset of the ISO 8601 format. Fractional seconds can be as precise as milliseconds. For example, a date and time of `2005-05-31T14:39:25.035Z` represents the thirty-first day of May in the year 2005. The time, measured in Coordinated Universal Time (UTC) as indicated by the `Z`, is 14 hours, 39 minutes, 25 seconds, and 35 milliseconds.

This property

- is of type `token`

is encoded as type `tns:multilingualTokenProp`

- can contain at least 255 characters
- is multilingual
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

objectClass

Specifies the class of the object.

This property

- is of type `bibus » classEnum`
 - is encoded as type `tns:classEnumProp`
- is read-only
- must contain no more than 64 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

owner

Refers to the owner of the object.

If a user is the owner of an object, the user has `read`, `write`, `traverse`, `setPolicy`, and `execute` permissions on the object. This ensures that users can always access and modify the objects that they own. By default, the owner of the object is the user who creates it, as identified by the primary account that the passport used in the request references.

A user with `setPolicy` permission on an object can set the owner property of that object to their account. The passport of the request that is used to change the owner property must reference the account.

System administrators have `setPolicy` permission on every object in the content store implicitly and can change the owner property of any object. A system administrator can also set the owner property to refer to any account known to the system.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » account`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 10.1.0 – “[New Limitations for Setting Object Ownership](#)” on page 1877

This property has been updated to limit the ability to set the value to an account in the request passport, unless the request is made by a system administrator.

parent

Refers to the container of this object. Note that the root object does not have a parent.

When you create an object, you must specify its parent.

This property

- is an array of type `bibus » baseClass`
is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

permissions

Contains the current user's access permissions for the object, such as `read`, `write`, `setPolicy`, or `execute`.

This property

- is an array of type `NMTOKEN`
is encoded as type `tns:nmtokenArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

policies

Contains the security policy for the object.

This property

- is an array of type `bibus » policy`
is encoded as type `tns:policyArrayProp`
- can be acquired from a containing object

position

Specifies the location of the object in the result set returned by the `content » query(searchPath, properties, sortBy, options)` method. The result set is defined by the values assigned to the method options.

The display order begins with 0 and increases.

This property

- is of type `nonNegativeInteger`
is encoded as type `tns:nonNegativeIntegerProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

searchPath

Specifies the path used to retrieve the object.

You may not be able to read or understand this path.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

storeID

Identifies the object in the content store.

This property

- is of type [bibus » guid](#)
 - is encoded as type `tns:guidProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

tenantID

Identifies the tenant space which this object belongs to in a multi-tenant content store.

The default value is an empty string, indicating that the object is either public in a multi-tenant content store or that the content store is not configured to support multiple tenants.

Content Manager trims leading and trailing spaces, and reduces two or more consecutive internal spaces to one. A `tenantID` consisting of spaces only is reduced to an empty string. For correct results, do not use spaces only for tenantIDs and do not use internal spaces to differentiate tenantIDs.

This property:

- Is set implicitly at object creation time based on the tenant ID in the current user session (for non-System Administrator users).
- Can be set explicitly by System Administrators at object creation time.
- Can be retrieved without requiring read permission on the object. Only traverse permission on the full path of the containing object is required.
- Is searchable.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

New in Version 10.2.0 – “Support for multi-tenancy” on page 1838

This property was added.

usage

Specifies how a portal can represent the object.

This property

- is of type [bibus » usageEnum](#)
 - is encoded as type `tns:nmtokenProp`
- is read-only
- can contain at most 255 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 10.2.2 – “Documentation Updates” on page 1827

This property size was amended to at most 255 characters.

version

Specifies the version of the object used for optimistic concurrency control. This prevents a user from overwriting the changes made by another user when both users are making changes to the same object.

This property is not related to object retention.

To use optimistic concurrency control, you must retrieve the version property for the object in the query request, and then include the retrieved value of the version property in the update request.

For more information about optimistic concurrency control, see [“Concurrency Control” on page 55](#).

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- is read-only
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

baseClassArrayProp

Defines the array property class for the [bibus » baseClass](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » agentDefinitionView » base](#)
- [bibus » dataMovementTaskAlias » base](#)
- [bibus » personalization » base](#)
- [bibus » portalSkin » base](#)
- [bibus » powerPlay8ReportView » base](#)
- [bibus » reportView » base](#)
- [bibus » baseRSSTask » channel](#)
- [bibus » baseDataIntegrationTask » connections](#)
- [bibus » dataSourceSignon » consumers](#)
- [bibus » adminFolder » contact](#)
- [bibus » aliasRoot » contact](#)
- [bibus » annotation » contact](#)
- [bibus » annotationFolder » contact](#)
- [bibus » archiveLocation » contact](#)
- [bibus » baseAgentDefinition » contact](#)
- [bibus » baseDataIntegrationTask » contact](#)
- [bibus » baseDataMovementTask » contact](#)
- [bibus » basePowerPlay8Report » contact](#)
- [bibus » basePowerPlayClass » contact](#)
- [bibus » baseReport » contact](#)

- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » schedule » credential](#)
- [bibus » dataSourceConnection » credentialNamespaces](#)
- [bibus » baseROLAPDataSource » dataSourceAccessAccount](#)

- [bibus » packageConfiguration » defaultAnalysis](#)
- [bibus » agentState » definition](#)
- [bibus » agentTaskState » definition](#)
- [bibus » deploymentDetail » deployedObject](#)
- [bibus » group » distributionMembers](#)
- [bibus » role » distributionMembers](#)
- [bibus » baseReport » executionPageDefinition](#)
- [bibus » session » identity](#)
- [bibus » configuration » jmxProxyHostDispatchers](#)
- [bibus » shortcutRSSTask » link](#)
- [bibus » historyDetailRelatedReports » linkPaths](#)
- [bibus » distributionList » members](#)
- [bibus » group » members](#)
- [bibus » role » members](#)
- [bibus » authoredPowerPlay8Report » metadataModel](#)
- [bibus » authoredReport » metadataModel](#)
- [bibus » baseDataIntegrationTask » metadataModel](#)
- [bibus » reportCache » metadataModel](#)
- [bibus » reportVersion » metadataModel](#)
- [bibus » storedProcedureTask » metadataModel](#)
- [bibus » authoredPowerPlay8Report » metadataModelPackage](#)
- [bibus » authoredReport » metadataModelPackage](#)
- [bibus » baseDataIntegrationTask » metadataModelPackage](#)
- [bibus » reportCache » metadataModelPackage](#)
- [bibus » reportVersion » metadataModelPackage](#)
- [bibus » storedProcedureTask » metadataModelPackage](#)
- [bibus » historyDetailMigrationService » migratedObject](#)
- [bibus » baseAgentDefinition » mostRecentEventList](#)
- [bibus » baseAgentDefinition » notificationList](#)
- [bibus » baseReport » notificationList](#)
- [bibus » historyDetailAgentService » output](#)
- [bibus » historyDetailReportService » output](#)
- [bibus » annotation » outputs](#)
- [bibus » baseClass » owner](#)
- [bibus » account » page](#)
- [bibus » pageletInstance » pagelet](#)
- [bibus » baseClass » parent](#)
- [bibus » schedule » periodicalProducer](#)
- [bibus » account » portalPages](#)
- [bibus » portletInstance » portlet](#)
- [bibus » periodical » producer](#)
- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)

- [bibus » agentTaskState » related](#)
- [bibus » historyDetailRelatedHistory » related](#)
- [bibus » group » securityMembers](#)
- [bibus » role » securityMembers](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » drillPath » target](#)
- [bibus » shortcut » target](#)
- [bibus » agentTaskDefinition » taskObject](#)
- [bibus » history » user](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » baseClass](#)
is encoded as type `tns:baseClassArray`

baseClassProp

Defines the simple property class for the [bibus » baseClass](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » baseClass](#)
is encoded as type `tns:baseClass`

baseDataIntegrationTask

Defines the set of common properties shared by all data integration service tasks.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » uiClass](#) class

Derived Classes

- [bibus » metricsDataSourceETLTask](#)

- [bibus](#) » [metricsExportTask](#)
- [bibus](#) » [metricsFileImportTask](#)
- [bibus](#) » [metricsMaintenanceTask](#)

References

Used by the following properties:

- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [history](#)
- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [dashboard](#)
- [bibus](#) » [folder](#)
- [bibus](#) » [package](#)

Properties

This class has the following properties.

connections

Refers to the set of data source connections and signons used to run this object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [dataSourceConnection](#) or [bibus](#) » [dataSourceSignon](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [model](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [package](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

ownerPassport

When [bibus » baseDataIntegrationTask » runAsOwner](#) property is true, this property specifies a CAM cookie created using the credentials associated with the object's owner. The passport and session created during the log on process are used to perform the requested task.

Once the task has completed, a logoff request is issued to invalidate the passport and terminate the session.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- cannot be processed directly using the IBM Cognos Software Development Kit

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic now accurately describes the contents of this property.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type `bibus » retentionRule`
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

runAsOwner

Specifies whether the security context of the owner is used to perform specific capability checks and to access data when running a task.

For example, the Human Resources department publishes a report that contains information about salary ranges across the company. The nature of the report requires the use of user defined SQL (controlled by `canUseUserDefinedSQL`). Since this report does not disclose individual salary information, it is not deemed to be confidential. However, the data required to run the report is only available to certain members of the Human Resources department. One of these employees is the owner of the report.

When this property is `true`, IBM Cognos uses the report owner's security context to access the data necessary to produce the report, as well as to perform certain user capability checks.

In the preceding example, the report owner sets this property to `true` to allow other employees in the company to run the report. Setting this property to `true` specifies that IBM Cognos uses the report owner's security context to access the data that is normally unavailable to the majority of employees. In addition, the capability check to determine whether user defined SQL can be used is performed using the report owner's security context. Consumers of the report may not be permitted access to the data or the ability to include user defined SQL in their own reports, but when this property is `true`,

the report owner's security context is used to access the data and to allow the use of user defined SQL.

A user must still have sufficient permissions for the target object in order to run it. For more information, see [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

Other activities may also use this property to provide access to privileged data or user capabilities. For example, [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method collects all the parameters that must be specified before a report can run.

For more information about which owner capabilities are used when this property is `true`, see the **User Capability Details** table in the [bibus](#) » [userCapabilityEnum](#) enumeration set description.

If a request is sent to set this property to `true`, the [contentManagerService](#) service requires that the security context making the request (that is, accounts, groups and active roles) has either `read` or `execute` permissions on the credential of the account identified by the [owner](#) of the object. This check is performed in addition to the normal permission checks for updating a property. If the security context does not have the appropriate permissions, the request fails.

If consumers of a report only require one of the owner's capabilities and do not require access to privileged data to run the report, use the [runWithOwnerCapabilities](#) property instead.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

specification

Specifies the specification for the task.

This property

- is of type [bibus](#) » [dataIntegrationServiceSpecification](#)
 - is encoded as type `tns:dataIntegrationServiceSpecificationProp`

baseDataMovementTask

Defines the set of common properties shared by all data movement task classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus](#) » [uiClass](#) class

Derived Classes

- [bibus](#) » [dataMovementTask](#)
- [bibus](#) » [dataMovementTaskAlias](#)

References

Used by the following properties:

- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)

- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [dashboard](#)
- [bibus](#) » [folder](#)
- [bibus](#) » [package](#)

What's new

New in Version 10.1.0 – [“Variable Support for Data Movement Tasks” on page 1854](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus](#) » [option](#)
 - is encoded as type `tns:optionArrayProp`

ownerPassport

When [bibus](#) » [baseDataMovementTask](#) » [runAsOwner](#) property is true, this property specifies a CAM cookie created using the credentials associated with the object's owner. The passport and session created during the log on process are used to perform the requested task.

Once the task has completed, a logoff request is issued to invalidate the passport and terminate the session.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- cannot be processed directly using the IBM Cognos Software Development Kit

New in Version 10.1.0 – “[Documentation Updates](#)” on page 1878

This topic now accurately describes the contents of this property.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type `bibus » parameterValue`
 - is encoded as type `tns:parameterValueArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “[Software Editions](#)” on page 1893

This property was added.

runAsOwner

Specifies whether the security context of the owner is used to perform specific capability checks and to access data when running a task.

For example, the Human Resources department publishes a report that contains information about salary ranges across the company. The nature of the report requires the use of user defined SQL (controlled by `canUseUserDefinedSQL`). Since this report does not disclose individual salary information, it is not deemed to be confidential. However, the data required to run the report is only available to certain members of the Human Resources department. One of these employees is the owner of the report.

When this property is `true`, IBM Cognos uses the report owner's security context to access the data necessary to produce the report, as well as to perform certain user capability checks.

In the previous example, the report owner sets this property to `true` to allow other employees in the company to run the report. Setting this property to `true` specifies that IBM Cognos uses the report owner's security context to access the data that is normally unavailable to the majority of employees. In addition, the capability check to determine whether user defined SQL can be used is performed using the report owner's security context. Consumers of the report may not be permitted access to the data or the ability to include user defined SQL in their own reports, but when this property is `true`, the report owner's security context is used to access the data and to allow the use of user defined SQL.

A user must still have sufficient permissions for the target object in order to run it. For more information, see [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

Other activities may also use this property to provide access to privileged data or user capabilities. For example, [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method collects all the parameters that must be specified before a report can run.

For more information about which owner capabilities are used when this property is `true`, see the **User Capability Details** table in the [bibus](#) » [userCapabilityEnum](#) enumeration set description.

If a request is sent to set this property to `true`, the [contentManagerService](#) service requires that the security context making the request (that is, accounts, groups and active roles) has either `read` or `execute` permissions on the credential of the account identified by the `owner` of the object. This check is performed in addition to the normal permission checks for updating a property. If the security context does not have the appropriate permissions, the request fails.

If consumers of a report only require one of the owner's capabilities and do not require access to privileged data to run the report, use the [runWithOwnerCapabilities](#) property instead.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

baseHistoryDetail

Provides details about the running of a task.

This abstract class defines a set of properties common to all history detail objects.

This class

- is an abstract type used to define properties that are inherited by other types

- inherits properties from the [bibus » baseClass](#) class

Derived Classes

- [bibus » deploymentDetail](#)
- [bibus » historyDetail](#)
- [bibus » historyDetailAgentService](#)
- [bibus » historyDetailDataMovementService](#)
- [bibus » historyDetailDeploymentSummary](#)
- [bibus » historyDetailMigrationService](#)
- [bibus » historyDetailRelatedHistory](#)
- [bibus » historyDetailRelatedReports](#)
- [bibus » historyDetailReportService](#)
- [bibus » historyDetailRequestArguments](#)

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » historyDetails](#)

Container Information

Contained by instances of the following classes

- [bibus » history](#)

Properties

This class has the following properties.

detail

Specifies the message for this history detail.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

detailTime

Specifies the time that the activity occurred, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`

severity

Specifies the severity of the message contained in the [detail](#) property.

This property

- is of type [bibus » severityEnum](#)
 - is encoded as type `tns:severityEnumProp`
- is searchable

baseParameter

Provides the abstract base class for classes that define parameter metadata.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » parameter](#)
- [bibus » parameterDataSource](#)

References

Used by the following properties:

- [bibus » asynchDetailParameters » parameters](#)
- [bibus » asynchDetailPromptPage » parameters](#)

Properties

This class has the following properties.

capabilities

Contains the parameter's capabilities.

This property

- is an array of type [bibus » parameterCapabilityEnum](#)
is encoded as type `tns:parameterCapabilityEnumArray`

caption

Specifies the title of the parameter. This is the title that a report user sees.

This property

- is of type `string`
is encoded as type `tns:multilingualStringArray`
- is multilingual

defaultValue

Contains the default value for the parameter.

This property

- is an array of type [bibus » parmValueItem](#)
is encoded as type `tns:parmValueItemArray`
- must have at most 1 item

name

Identifies the parameter.

Because the prefix `credential:` is reserved for parameters that represent data sources, do not use a colon (:) in a parameter name.

This property

- is of type `token`
is encoded as type `xs:string`

promptType

Specifies the type of the prompt associated with the parameter.

Note: This property always contains NULL.

This property

- is of type [bibus » promptTypeEnum](#)
is encoded as type `tns:promptTypeEnum`

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic now indicates that this property always has the value NULL.

type

Specifies the data type of the parameter.

This property

- is of type [bibus » parameterDataTypeEnum](#)
is encoded as type `tns:parameterDataTypeEnum`

values

Contains the set of possible values for the parameter.

This property

- is an array of type [bibus » parmValueItem](#)
is encoded as type `tns:parmValueItemArray`

baseParameterAssignment

Defines the base class used to specify the assignment of data to a parameter.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » parameterAssignmentDataItem](#)

References

Used by the following properties:

- [bibus » agentTaskDefinition » parameterAssignments](#)
- [bibus » drillPath » parameterAssignments](#)
- [bibus » baseParameterAssignmentArrayProp » value](#)
- [bibus » baseParameterAssignmentProp » value](#)

Properties

This class has the following properties.

parameterName

Specifies the name of the parameter.

This property

- is of type `string`

is encoded as type `xs:string`

baseParameterAssignmentArrayProp

Defines the array property class for the [bibus » baseParameterAssignment](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » agentTaskDefinition » parameterAssignments](#)
- [bibus » drillPath » parameterAssignments](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » baseParameterAssignment](#)
is encoded as type `tns:baseParameterAssignmentArray`

baseParameterAssignmentProp

Defines the simple property class for the [bibus » baseParameterAssignment](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » baseParameterAssignment](#)
is encoded as type `tns:baseParameterAssignment`

basePowerPlay8Report

Defines the set of common properties shared by all PowerPlay report classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » uiClass](#) class

Derived Classes

- [bibus » authoredPowerPlay8Report](#)

- [bibus](#) » [powerPlay8ReportView](#)

References

Used by the following properties:

- [bibus](#) » [shortcutRSSTask](#) » [link](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)
- [bibus](#) » [shortcut](#) » [target](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [reportVersion](#)
- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [dashboard](#)
- [bibus](#) » [folder](#)
- [bibus](#) » [package](#)

What's new

New in Version 8.4 — “[PowerPlay 8 Integration](#)” on page 1886

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

defaultOutputFormat

Specifies an output format if the object contains output that the current user has permission to read.

If the object contains output that the current user has permission to read, this property contains the output format used by the `defaultOutput(searchPath)` function specified in the search path to select objects.

For more information about the `defaultOutput(searchPath)` function, see [“defaultOutput\(searchPath\)” on page 1571](#).

This property

- is an array of type [bibus » powerPlay8OutputFormatEnum](#)
is encoded as type `tns:powerPlay8OutputFormatEnumArrayProp`
- is read-only
- must have at most 1 item

defaultPortalAction

Specifies the default action for an object in a portal.

When users click the hyperlinked name in the portal, the default action specified, such as view, run, or open for editing, is performed.

This property

- is of type [bibus » basePowerPlay8ReportActionEnum](#)
is encoded as type `tns:basePowerPlay8ReportActionEnumProp`
- has a default value of `run`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Table 151. Rules for new basePowerPlay8Report object

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » reportVersion class		1	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

versions

Manages the report versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » reportVersion](#)

basePowerPlay8ReportActionEnumProp

Defines the simple property class for the [bibus » basePowerPlay8ReportActionEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » basePowerPlay8Report » defaultPortalAction](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » basePowerPlay8ReportActionEnum](#)
is encoded as type `tns:basePowerPlay8ReportActionEnum`

basePowerPlayClass

Defines the set of common properties shared by PowerPlay classes that can be queried from Content Manager.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » uiClass](#) class

Derived Classes

- [bibus » powerPlayCube](#)
- [bibus » powerPlayReport](#)

References

Used by the following properties:

- [bibus » shortcutRSSTask » link](#)
- [bibus » shortcut » target](#)

Container Information

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

cubeContentName

Specifies the content name of the associated cube on the PPES server.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is searchable

dataBlocks

Specifies the data blocks for the object.

This property

- is an array of type [bibus » powerPlayDataBlock](#)
 - is encoded as type `tns:powerPlayDataBlockArrayProp`

gateway

Specifies the PPES gateway used to run the object.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is searchable

options

Specifies the PowerPlay options for the object.

This property

- is an array of type [bibus » powerPlayOption](#)
 - is encoded as type `tns:powerPlayOptionArrayProp`

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus](#) » [baseClass](#) » [ancestors](#) property instead.

baseProp

Defines the abstract base class for all property types.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus](#) » [addressSMTPArrayProp](#)
- [bibus](#) » [addressSMTPProp](#)
- [bibus](#) » [agentOutputEnumProp](#)
- [bibus](#) » [ancestorInfoArrayProp](#)
- [bibus](#) » [ancestorInfoProp](#)
- [bibus](#) » [anyTypeMIMEProp](#)
- [bibus](#) » [anyTypeProp](#)
- [bibus](#) » [anyURIArrayProp](#)
- [bibus](#) » [anyURIProp](#)
- [bibus](#) » [auditLevelEnumProp](#)
- [bibus](#) » [base64BinaryMIMEProp](#)
- [bibus](#) » [baseAgentDefinitionActionEnumProp](#)
- [bibus](#) » [baseClassArrayProp](#)
- [bibus](#) » [baseClassProp](#)
- [bibus](#) » [baseParameterAssignmentArrayProp](#)
- [bibus](#) » [baseParameterAssignmentProp](#)
- [bibus](#) » [basePowerPlay8ReportActionEnumProp](#)
- [bibus](#) » [baseReportActionEnumProp](#)
- [bibus](#) » [baseROLAPCubeConfigurationArrayProp](#)
- [bibus](#) » [baseROLAPCubeConfigurationProp](#)
- [bibus](#) » [booleanProp](#)
- [bibus](#) » [classEnumArrayProp](#)
- [bibus](#) » [classEnumProp](#)
- [bibus](#) » [currencyArrayProp](#)
- [bibus](#) » [currencyProp](#)
- [bibus](#) » [dataIntegrationServiceSpecificationArrayProp](#)
- [bibus](#) » [dataIntegrationServiceSpecificationProp](#)

- [bibus » dataMovementServiceSpecificationProp](#)
- [bibus » dataSourceCommandBlockProp](#)
- [bibus » dateTimeProp](#)
- [bibus » decimalProp](#)
- [bibus » deploymentOptionArrayProp](#)
- [bibus » deploymentOptionProp](#)
- [bibus » deploymentReferenceArrayProp](#)
- [bibus » deploymentReferenceProp](#)
- [bibus » deploymentStatusEnumProp](#)
- [bibus » durationProp](#)
- [bibus » eventTypeEnumArrayProp](#)
- [bibus » eventTypeEnumProp](#)
- [bibus » faultDetailArrayProp](#)
- [bibus » faultDetailProp](#)
- [bibus » floatProp](#)
- [bibus » gatewayMappingArrayProp](#)
- [bibus » gatewayMappingProp](#)
- [bibus » governorArrayProp](#)
- [bibus » governorProp](#)
- [bibus » guidArrayProp](#)
- [bibus » guidProp](#)
- [bibus » indexUpdateServiceSpecificationProp](#)
- [bibus » installedComponentEnumProp](#)
- [bibus » intProp](#)
- [bibus » languageProp](#)
- [bibus » loadBalancingModeEnumProp](#)
- [bibus » localeArrayProp](#)
- [bibus » localeMapEntryArrayProp](#)
- [bibus » localeMapEntryProp](#)
- [bibus » localeProp](#)
- [bibus » metadataModelItemNameArrayProp](#)
- [bibus » metadataModelItemNameProp](#)
- [bibus » migrationServiceSpecificationProp](#)
- [bibus » multilingualStringProp](#)
- [bibus » multilingualTokenProp](#)
- [bibus » navigationPathArrayProp](#)
- [bibus » navigationPathProp](#)
- [bibus » ncnameProp](#)
- [bibus » nmtokenArrayProp](#)
- [bibus » nmtokenProp](#)
- [bibus » nonNegativeIntegerProp](#)
- [bibus » optionArrayProp](#)
- [bibus » optionProp](#)

- [bibus » packageActionEnumProp](#)
- [bibus » parameterValueArrayProp](#)
- [bibus » parameterValueProp](#)
- [bibus » pdfCharacterEncodingEnumProp](#)
- [bibus » pdfCompressionTypeEnumProp](#)
- [bibus » pdfFontEmbeddingEnumProp](#)
- [bibus » planningAdministrationConsoleServiceSpecificationProp](#)
- [bibus » planningStateEnumProp](#)
- [bibus » planningTaskServiceSpecificationProp](#)
- [bibus » policyArrayProp](#)
- [bibus » policyProp](#)
- [bibus » positiveIntegerProp](#)
- [bibus » powerPlay8OutputFormatEnumArrayProp](#)
- [bibus » powerPlay8OutputFormatEnumProp](#)
- [bibus » powerPlayDataBlockArrayProp](#)
- [bibus » powerPlayDataBlockProp](#)
- [bibus » powerPlayOptionArrayProp](#)
- [bibus » powerPlayOptionProp](#)
- [bibus » powerPlayReportActionEnumProp](#)
- [bibus » powerPlayServiceSpecificationProp](#)
- [bibus » reportDataServiceSpecificationProp](#)
- [bibus » reportServiceDrillThroughSpecificationProp](#)
- [bibus » repositoryRuleArrayProp](#)
- [bibus » repositoryRuleProp](#)
- [bibus » retentionRuleArrayProp](#)
- [bibus » retentionRuleProp](#)
- [bibus » routingTableEntryArrayProp](#)
- [bibus » routingTableEntryProp](#)
- [bibus » runConditionEnumProp](#)
- [bibus » runningStateEnumProp](#)
- [bibus » runOptionArrayProp](#)
- [bibus » runOptionProp](#)
- [bibus » searchPathSingleObjectArrayProp](#)
- [bibus » searchPathSingleObjectProp](#)
- [bibus » severityEnumProp](#)
- [bibus » softwareEditionArrayProp](#)
- [bibus » softwareEditionProp](#)
- [bibus » stringArrayProp](#)
- [bibus » stringMapEntryArrayProp](#)
- [bibus » stringMapEntryProp](#)
- [bibus » stringProp](#)
- [bibus » timeProp](#)
- [bibus » tokenArrayProp](#)

- [bibus](#) » [tokenProp](#)
- [bibus](#) » [uiComponentEnumArrayProp](#)
- [bibus](#) » [uiComponentEnumProp](#)
- [bibus](#) » [uriValueArrayProp](#)
- [bibus](#) » [uriValueProp](#)
- [bibus](#) » [userCapabilityEnumArrayProp](#)
- [bibus](#) » [userCapabilityEnumProp](#)
- [bibus](#) » [userCapabilityPolicyArrayProp](#)
- [bibus](#) » [userCapabilityPolicyProp](#)
- [bibus](#) » [xmlEncodedXMLArrayProp](#)
- [bibus](#) » [xmlEncodedXMLMIMEProp](#)
- [bibus](#) » [xmlEncodedXMLProp](#)

Properties

This class has the following properties.

schemaInfo

Specifies the schema information for this property value.

This property

- is of type [bibus](#) » [schemaInfo](#)
is encoded as type `tns:schemaInfo`

baseReport

Defines the set of common properties shared by all report classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus](#) » [uiClass](#) class

Derived Classes

- [bibus](#) » [authoredReport](#)
- [bibus](#) » [reportView](#)

References

Used by the following properties:

- [bibus](#) » [shortcutRSSTask](#) » [link](#)
- [bibus](#) » [schedule](#) » [periodicalProducer](#)
- [bibus](#) » [periodical](#) » [producer](#)
- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)
- [bibus](#) » [navigationPath](#) » [target](#)
- [bibus](#) » [shortcut](#) » [target](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » reportCache](#)
- [bibus » reportVersion](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)

Properties

This class has the following properties.

allowNotification

Specifies whether the object allows alerts.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `true`

New in Version 8.3 — “Report Email Alerts” on page 1900

This property was added.

allowSubscription

Specifies whether the report allows subscription.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

cache

Manages the report cache.

This property

- must have at most 1 item
- uses the [retentions](#) property to manage contained instances of [bibus » reportCache](#)

New in Version 8.4 — “On Demand Refresh of Prompt Cache” on page 1884

This property now supports retention rules on [bibus » reportCache](#) objects.

connections – obsolete

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

defaultOutputFormat

Specifies an output format if the object contains output that the current user has permission to read.

If the object contains output that the current user has permission to read, this property contains the output format used by the `defaultOutput(searchPath)` function specified in the search path to select objects.

For more information about the `defaultOutput(searchPath)` function, see [“defaultOutput\(searchPath\)” on page 1571](#).

This property

- is an array of type [bibus » outputFormatEnum](#)
is encoded as type `tns:nmtokenArrayProp`
- is read-only
- must have at most 1 item

defaultPortalAction

Specifies the default action for an object in a portal.

When users click the hyperlinked name in the portal, the default action specified, such as view, run, or open for editing, is performed.

This property

- is of type [bibus » baseReportActionEnum](#)
is encoded as type `tns:baseReportActionEnumProp`
- has a default value of `viewOutput`

executionFormat – deprecated

Specifies the default output format for the object.

This property

- is of type [bibus » outputFormatEnum](#)
is encoded as type `tns:nmtokenProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters

executionLocale – deprecated

Specifies the default locale used for the output. The code appears as a hyphenated pair, such as `en-us` or `en-gb`, in accordance with Internet Engineering Task Force (IETF) open standard RFC3066.

This property

- is of type `language`
 - is encoded as type `tns:languageProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters

executionPageDefinition

Refers to the page settings for this object, such as the height and width of the page.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » pageDefinition`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

executionPageOrientation – deprecated

Specifies the default page orientation for the output.

This property

- is of type `bibus » pageOrientationEnum`
 - is encoded as type `tns:nmtokenProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters

executionPrompt – deprecated

Specifies the prompt option for the object. Allows a value to be stored for the administrator-defined `bibus » runOptionEnum » prompt` value.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `true`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the `retentions` property to manage contained instances of `bibus » history`

notificationList

Refers to the set of accounts that receive alerts when the object runs.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » account`
 - is encoded as type `tns:baseClassArrayProp`
- cannot be processed directly using the IBM Cognos Software Development Kit
- refers to other objects in the content store using an id-based search path

New in Version 8.3 – “Report Email Alerts” on page 1900

This property was added.

notificationListIsEmpty

Specifies whether the [notificationList](#) property is empty.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only

New in Version 8.3 – “Report Email Alerts” on page 1900

This property was added.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

ownerPassport

When [bibus » baseReport » runAsOwner](#) property is true, this property specifies a CAM cookie created using the credentials associated with the object's owner. The passport and session created during the log on process are used to perform the requested task.

Once the task has completed, a logoff request is issued to invalidate the passport and terminate the session.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- cannot be processed directly using the IBM Cognos Software Development Kit

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic now accurately describes the contents of this property.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property
bibus » reportVersion class		1	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters

- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

runAsOwner

Specifies whether the security context of the owner is used to perform specific capability checks and to access data when running a task.

For example, the Human Resources department publishes a report that contains information about salary ranges across the company. The nature of the report requires the use of user defined SQL (controlled by `canUseUserDefinedSQL`). Since this report does not disclose individual salary information, it is not deemed to be confidential. However, the data required to run the report is only available to certain members of the Human Resources department. One of these employees is the owner of the report.

When this property is `true`, IBM Cognos uses the report owner's security context to access the data necessary to produce the report, as well as to perform certain user capability checks.

In the preceding example, the report owner sets this property to `true` to allow other employees in the company to run the report. Setting this property to `true` specifies that IBM Cognos uses the report owner's security context to access the data that is normally unavailable to the majority of employees. In addition, the capability check to determine whether user defined SQL can be used is performed using the report owner's security context. Consumers of the report may not be permitted access to the data or the ability to include user defined SQL in their own reports, but when this property is `true`, the report owner's security context is used to access the data and to allow the use of user defined SQL.

A user must still have sufficient permissions for the target object in order to run it. For more information, see [asynch » run\(objectPath, parameterValues, options\)](#) method.

Other activities may also use this property to provide access to privileged data or user capabilities. For example, `parameter » collectParameterValues(objectPath, parameterValues, options)` method collects all the parameters that must be specified before a report can run.

For more information about which owner capabilities are used when this property is `true`, see the **User Capability Details** table in the [bibus » userCapabilityEnum](#) enumeration set description.

If a request is sent to set this property to `true`, the `contentManagerService` service requires that the security context making the request (that is, accounts, groups and active roles) has either `read` or `execute` permissions on the credential of the account identified by the `owner` of the object. This check is performed in addition to the normal permission checks for updating a property. If the security context does not have the appropriate permissions, the request fails.

If consumers of a report only require one of the owner's capabilities and do not require access to privileged data to run the report, use the [runWithOwnerCapabilities](#) property instead.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

versions

Manages the report versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [reportVersion](#)

baseReportActionEnumProp

Defines the simple property class for the [bibus](#) » [baseReportActionEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [drillPath](#) » [action](#)
- [bibus](#) » [baseReport](#) » [defaultPortalAction](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [baseReportActionEnum](#)
 - is encoded as type `tns:baseReportActionEnum`

baseROLAPCubeConfiguration

Defines the abstract class for the set of common ROLAP configuration data.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus](#) » [rolapCubeConfiguration](#)
- [bibus](#) » [rolapVirtualCubeConfiguration](#)

References

Used by the following properties:

- [bibus » configuration » qsROLAPCubeConfigurations](#)
- [bibus » configurationFolder » qsROLAPCubeConfigurations](#)
- [bibus » dispatcher » qsROLAPCubeConfigurations](#)
- [bibus » queryService » qsROLAPCubeConfigurations](#)
- [bibus » baseROLAPCubeConfigurationArrayProp » value](#)
- [bibus » baseROLAPCubeConfigurationProp » value](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This class was added.

Properties

This class has the following properties.

dataCacheSizeLimit

Specifies the maximum size, in MB, of the ROLAP cube data cache.

This property

- is of type `int`
 - is encoded as type `xs:int`
- has a default value of 1024
- must contain a value greater than or equal to 0

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

dataSourceName

Specifies the name of the ROLAP cube data source to be configured.

This property

- is of type `string`
 - is encoded as type `xs:string`

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

disabled

Specifies whether the ROLAP cube is disabled.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

disableResultSetCache

Specifies whether the result set cache is disabled.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This property was added.

multiDimensionalQuerySizeLimit

Specifies the memory size, in MB, allotted to each locally executed multi-dimensional query. A value less than 0 means that the actual value should be obtained from [bibus » configuration » qsMultiDimensionalQuerySizeLimit](#), [bibus » configurationFolder » qsMultiDimensionalQuerySizeLimit](#), [bibus » dispatcher » qsMultiDimensionalQuerySizeLimit](#), and [bibus » queryService » qsMultiDimensionalQuerySizeLimit](#).

This property

- is of type `int`
 - is encoded as type `xs:int`
- has a default value of `-1`

New in Version 10.2.0 – “[IBM Cognos Dynamic Cubes Aggregate Advisor configuration](#)” on page 1836

This property was added.

resultSetCacheSizeLimit

Specifies the maximum size, in MB, of the result set cache for the ROLAP cube.

This property

- is of type `int`
 - is encoded as type `xs:int`
- has a default value of `1024`
- must contain a value greater than or equal to `100`

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This property was added.

startupTriggerName

Specifies the trigger event that is initiated once the ROLAP cube has started.

This property

- is of type `string`
 - is encoded as type `xs:string`

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This property was added.

baseROLAPCubeConfigurationArrayProp

Defines the array property class for the [bibus » baseROLAPCubeConfiguration](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » configuration » qsROLAPCubeConfigurations](#)
- [bibus » configurationFolder » qsROLAPCubeConfigurations](#)
- [bibus » dispatcher » qsROLAPCubeConfigurations](#)
- [bibus » queryService » qsROLAPCubeConfigurations](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » baseROLAPCubeConfiguration](#)
is encoded as type `tns:baseROLAPCubeConfigurationArray`

baseROLAPCubeConfigurationProp

Defines the simple property class for the [bibus » baseROLAPCubeConfiguration](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » baseROLAPCubeConfiguration](#)
is encoded as type `tns:baseROLAPCubeConfiguration`

baseROLAPDataSource

Defines the abstract class for the set of common ROLAP data source properties.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » uiClass](#) class

Derived Classes

- [bibus » rolapDataSource](#)
- [bibus » rolapVirtualDataSource](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [model](#)

Contained by instances of the following classes

- [bibus](#) » [namespace](#)

What's new

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This class was added.

Properties

This class has the following properties.

capabilities

Contains the set of capabilities that can be used with the ROLAP data source.

This property

- is an array of type [bibus](#) » [dataSourceCapabilityEnum](#)
is encoded as type `tns:anyURIArrayProp`
- is searchable

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

dataSourceAccessAccount

Refers to the account used to access data in the relational data source underlying the ROLAP data source.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [account](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

- must have at most 1 item

model

Specifies the metadata that defines the ROLAP data source.

This property

- must have at most 1 item

baseRoutingRule

Defines the base class for all dispatcher routing rules.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus](#) » [routingRuleObject](#)

References

Used by the following properties:

- [bibus](#) » [routingTableEntry](#) » [rule](#)

baseRSSTask

Defines the abstract base class that describes tasks that update an RSS channel.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus](#) » [uiClass](#) class

Derived Classes

- [bibus](#) » [shortcutAgentRSSTask](#)
- [bibus](#) » [shortcutRSSTask](#)
- [bibus](#) » [urlRSSTask](#)

References

Used by the following properties:

- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [history](#)

Contained by instances of the following classes

- [bibus](#) » [authoredAgentDefinition](#)

Properties

This class has the following properties.

channel

Refers to the object that represents the RSS channel. This object will contain the RSS item created by this task.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » folder](#) or [bibus » package](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type `bibus » retentionRule`
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

title

Specifies a title for the RSS item created by this task.

This property

- is of type `string`
is encoded as type `tns:multilingualStringProp`
- is multilingual

batchReportService

Defines run-time configuration parameters for the `batchReportService`.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated `bibus » dispatcher` object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

brsAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type int
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsAuditLevel

Specifies the auditing level for the batch report service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

brsAuditNativeQuery

Specifies whether to log native query information for the batch report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

brsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 – “Chart Hotspot Limit” on page 1909

This property was added.

brsDataSourceChange

Specifies the change time of the data source for the batch report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

brsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

brsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the batch report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

brsMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

brsNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

brsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the batch report service.

This property

- is of type `bibus » pdfCharacterEncodingEnum`
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of `auto`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPDFCompressionLevel

Specifies the compression level for PDF documents created by the batch report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

`bibus » batchReportService » brsPDFCompressionType` property

brsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the batch report service.

This property

- is of type [bibus » pdfCompressionTypeEnum](#)
 - is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » batchReportService » brsPDFCompressionLevel](#) property

brsPDFEmbedFonts

Specifies whether the batch report service should embed fonts in generated PDF documents.

This property

- is of type [bibus » pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

brsPeakAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 — “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsPeakMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

brsPeakNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

biBusHeader

Defines the class for the SOAP 1.1 header entry used by IBM Cognos Analytics.

This header entry must be included in all BI Bus API Simple Object Access Protocol (SOAP) messages.

This class defines part of the web service protocol for IBM Cognos.

Properties

This class has the following properties.

CAF

Specifies the CAF settings to be passed on requests.

This property

- is of type [bibus](#) » [CAF](#)
is encoded as type `tns:CAF`

CAM

Specifies the structured header information required to implement security when handling BI Bus API requests, including settings and actions such as `logon`, `logoff`, and `namespace`.

This property

- is of type [bibus](#) » [CAM](#)
is encoded as type `tns:CAM`

CAMProtect

Specifies security controls that are related to a particular service.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [CAMProtect](#)
is encoded as type `tns:CAMProtect`

dispatcherTransportVars

Specifies information that governs the way that dispatchers process requests.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is an array of type [bibus](#) » [dispatcherTransportVar](#)
is encoded as type `tns:dispatcherTransportVarArray`

extension

Specifies a location for future header properties.

This property

- is of type [bibus](#) » [biBusHeaderExtension1](#)
is encoded as type `tns:biBusHeaderExtension1`

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

hdrSession

Specifies the environment of the Web server gateway that received the request and the browser cookies for the request and the response.

This property

- is of type [bibus](#) » [hdrSession](#)
is encoded as type `tns:hdrSession`

providers

Specifies the patch level of an API or behavior.

This property

- is an array of type [bibus](#) » [provider](#)
is encoded as type `tns:providerArray`

routing

Specifies the structured header information required to implement request routing.

This property

- is of type [bibus](#) » [routingInfo](#)
is encoded as type `tns:routingInfo`

tracking

Specifies the information about a request or response for the log message facility.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [tracking](#)
is encoded as type `tns:tracking`

userCapabilityCache

Specifies the global user capability cache.

This property

- is of type [bibus](#) » [userCapabilityCache](#)
is encoded as type `tns:userCapabilityCache`

New in Version 8.4 — “Object Capabilities” on page 1885

This property was added.

userPreferenceVars

Specifies the current set of user preferences.

This property

- is an array of type [bibus](#) » [userPreferenceVar](#)

is encoded as type `tns:userPreferenceVarArray`

biBusHeaderExtension1

Defines the class for [bibus](#) » [biBusHeader](#) extensions.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [extension](#)

What's new

New in Version 8.4 — “Software Editions” on page 1893

This class was added.

booleanProp

Defines the simple property class for the `boolean`.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [namespace](#) » [active](#)
- [bibus](#) » [schedule](#) » [active](#)
- [bibus](#) » [agentOutputHotList](#) » [allowAnnotations](#)
- [bibus](#) » [reportVersion](#) » [allowAnnotations](#)
- [bibus](#) » [baseAgentDefinition](#) » [allowNotification](#)
- [bibus](#) » [baseReport](#) » [allowNotification](#)
- [bibus](#) » [baseReport](#) » [allowSubscription](#)
- [bibus](#) » [batchReportService](#) » [brsAuditNativeQuery](#)
- [bibus](#) » [configuration](#) » [brsAuditNativeQuery](#)
- [bibus](#) » [configurationFolder](#) » [brsAuditNativeQuery](#)
- [bibus](#) » [dispatcher](#) » [brsAuditNativeQuery](#)
- [bibus](#) » [humanTask](#) » [bulkEvents](#)
- [bibus](#) » [webServiceTask](#) » [bulkEvents](#)
- [bibus](#) » [authoredReport](#) » [canBurst](#)
- [bibus](#) » [reportVersion](#) » [canBurst](#)
- [bibus](#) » [portlet](#) » [canCustomize](#)
- [bibus](#) » [configuration](#) » [cookieCAMPassportHttpOnly](#)
- [bibus](#) » [baseClass](#) » [disabled](#)
- [bibus](#) » [baseClass](#) » [hasChildren](#)
- [bibus](#) » [deploymentDetail](#) » [hasMessage](#)
- [bibus](#) » [historyDetailMigrationService](#) » [hasMessage](#)
- [bibus](#) » [powerPlayReport](#) » [hasPrompts](#)
- [bibus](#) » [uiClass](#) » [hidden](#)

- [bibus » baseAgentDefinition » notificationListIsEmpty](#)
- [bibus » baseReport » notificationListIsEmpty](#)
- [bibus » configuration » pdsShowCellAnnotations](#)
- [bibus » configurationFolder » pdsShowCellAnnotations](#)
- [bibus » dispatcher » pdsShowCellAnnotations](#)
- [bibus » planningDataService » pdsShowCellAnnotations](#)
- [bibus » portalSkin » published](#)
- [bibus » configuration » qsDiagnosticsEnabled](#)
- [bibus » configurationFolder » qsDiagnosticsEnabled](#)
- [bibus » dispatcher » qsDiagnosticsEnabled](#)
- [bibus » queryService » qsDiagnosticsEnabled](#)
- [bibus » configuration » qsDisableQueryPlanCache](#)
- [bibus » configurationFolder » qsDisableQueryPlanCache](#)
- [bibus » dispatcher » qsDisableQueryPlanCache](#)
- [bibus » queryService » qsDisableQueryPlanCache](#)
- [bibus » configuration » qsDisableVerboseGCLogging](#)
- [bibus » configurationFolder » qsDisableVerboseGCLogging](#)
- [bibus » dispatcher » qsDisableVerboseGCLogging](#)
- [bibus » queryService » qsDisableVerboseGCLogging](#)
- [bibus » configuration » qsDumpModelToFile](#)
- [bibus » configurationFolder » qsDumpModelToFile](#)
- [bibus » dispatcher » qsDumpModelToFile](#)
- [bibus » queryService » qsDumpModelToFile](#)
- [bibus » configuration » qsGenerateCommentsInNativeSQL](#)
- [bibus » configurationFolder » qsGenerateCommentsInNativeSQL](#)
- [bibus » dispatcher » qsGenerateCommentsInNativeSQL](#)
- [bibus » queryService » qsGenerateCommentsInNativeSQL](#)
- [bibus » configuration » qsManualCubeStart](#)
- [bibus » configurationFolder » qsManualCubeStart](#)
- [bibus » dispatcher » qsManualCubeStart](#)
- [bibus » queryService » qsManualCubeStart](#)
- [bibus » configuration » qsMetricsEnabled](#)
- [bibus » configurationFolder » qsMetricsEnabled](#)
- [bibus » dispatcher » qsMetricsEnabled](#)
- [bibus » queryService » qsMetricsEnabled](#)
- [bibus » configuration » qsQueryExecutionTrace](#)
- [bibus » configurationFolder » qsQueryExecutionTrace](#)
- [bibus » dispatcher » qsQueryExecutionTrace](#)
- [bibus » queryService » qsQueryExecutionTrace](#)
- [bibus » configuration » qsQueryPlanningTrace](#)
- [bibus » configurationFolder » qsQueryPlanningTrace](#)
- [bibus » dispatcher » qsQueryPlanningTrace](#)
- [bibus » queryService » qsQueryPlanningTrace](#)

- [bibus » configuration » rsAuditNativeQuery](#)
- [bibus » configurationFolder » rsAuditNativeQuery](#)
- [bibus » dispatcher » rsAuditNativeQuery](#)
- [bibus » reportService » rsAuditNativeQuery](#)
- [bibus » baseAgentDefinition » runAsOwner](#)
- [bibus » baseDataIntegrationTask » runAsOwner](#)
- [bibus » baseDataMovementTask » runAsOwner](#)
- [bibus » baseReport » runAsOwner](#)
- [bibus » authoredReport » runWithOwnerCapabilities](#)
- [bibus » dataMovementTask » runWithOwnerCapabilities](#)
- [bibus » uiClass » shown](#)
- [bibus » contentManagerService » startAsActive](#)
- [bibus » account » useAccessibilityFeatures](#)
- [bibus » uiClass » viewed](#)
- [bibus » schedule » weeklyFriday](#)
- [bibus » schedule » weeklyMonday](#)
- [bibus » schedule » weeklySaturday](#)
- [bibus » schedule » weeklySunday](#)
- [bibus » schedule » weeklyThursday](#)
- [bibus » schedule » weeklyTuesday](#)
- [bibus » schedule » weeklyWednesday](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `boolean`
- is encoded as type `xs:boolean`

boundRangeParmValueItem

Defines a value item for the bound range parameter.

This class

- inherits properties from the [bibus » parmValueItem](#) class

Properties

This class has the following properties.

end

Specifies the maximum value of the bound range.

This property

- is of type [bibus » simpleParmValueItem](#)
- is encoded as type `tns:simpleParmValueItem`

start

Specifies the minimum value of the bound range.

This property

- is of type [bibus » simpleParmValueItem](#)
is encoded as type `tns:simpleParmValueItem`

cacheOutput

Provides report cache output.

In order to use cache data to render prompt data when a report is run, the language and region code specified for the [locale](#) need to match. If a match occurs, the cached data is used to render the prompt controls. If a match does not occur, the cached data is not used and a request is made to retrieve the prompt data.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contains instances of the following classes

- [bibus » graphic](#)
- [bibus » page](#)

Contained by instances of the following classes

- [bibus » reportCache](#)

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

images

Reserved.

locale

Specifies the locale for this object. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

When used by [bibus](#) » [contact](#), this property determines the language and data format of the returned content.

Use the appropriate language so that users understand object names and search paths. Use the appropriate region so that date, time, and currency values are presented in the proper localized format.

This property

- is of type `language`
 - is encoded as type `tns:languageProp`
- must contain no more than 64 characters
- is searchable

pages

Reserved.

CAF

Use this class to specify security properties for the IBM Cognos Application Firewall.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [CAF](#)

Properties

This class has the following properties.

contextID

Specifies the value for the CAF Context ID. It is used by XML-based applications (other than the SDK) to pass the CAF contextID on all requests.

This property

- is of type `string`
is encoded as type `xs:string`

secureState

Specifies the value for CAF secure state. It is used to avoid redundant validations.

This property

- is of type `string`
is encoded as type `xs:string`

CAM

Defines the class for the structured header information required to implement security when handling BI Bus API requests.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [CAM](#)

Properties

This class has the following properties.

action

Specifies the action to be taken by the security provider before the request specified by the Simple Object Access Protocol (SOAP) body is sent to the message recipient. The action can be any one of the following.

logon

This action specifies that the security provider either authenticates the user by using the credentials provided with the request or by using the anonymous facilities provided by the security provider.

If anonymous access is enabled on the server, the logon action logs the user on as an anonymous user.

If anonymous access is disabled on the server, a user is either logged on as they are on their local computer, or by NT Challenge/Response, if the server is running Internet Information Services (IIS) and the security provider is either NTLM or Microsoft Active Directory Server.

If authentication is successful, a passport is stored in the [CAMPassport](#) property.

logonAs

This action specifies that the security provider authenticates the credentials provided with the request. The user is authenticated as a named user, regardless of the setting for anonymous logon.

The logonAs action exists so that you can specify a named user, regardless of whether anonymous logon is enabled. You can also use logonAs without the namespace prompt by binding each Web server to a particular namespace, using a different virtual directory for each CGI gateway. The system automatically inserts the appropriate namespace element and value into the CAM element

of every message passed through the gateway, regardless of the authentication method normally used.

You can use the `logonAs` action multiple times to be authenticated using different namespaces at the same time. One `logoff` action ends the session for every authenticated namespace.

If authentication is successful, a passport is stored in the `CAMPassport` property.

logoff

This action specifies that the passport stored in the `bibus » CAM » CAMPassport` property should be expired and removed from the request.

The result of this logout request is an empty CAM element. Also, the `CAM_passport` cookie for Web portals in the `bibus » setCookieVar` class is cleared.

One `logoff` action ends the session for every authenticated namespace.

selectRole

This action specifies that the security provider should authenticate the user's credentials that are provided with the request and assign the specified roles to the session. If authentication is successful, a passport is stored in the `CAMPassport` property.

generateTC

This action specifies that the security provider should generate a trusted credential for the user identified by the passport stored in the `CAMPassport` property.

Trusted credentials are a form of user credentials used to schedule tasks as a named or anonymous user.

Trusted credentials define information that can be used to re-establish a session with a security infrastructure without the need for user intervention. The set of information used may vary by security infrastructure. However, IBM Cognos Analytics encodes and protects the trusted credentials through encryption.

If generation of the trusted credential is successful, the path to the trusted credential is stored in the `CAMCredentialPath` property. To authenticate using the generated trusted credentials, the action must be set to either `logon` or `logonAs`.

All requests for security information require that the user be authenticated by a security system. For example, requests for users, groups, or user attributes are passed through the BI Bus API to IBM Cognos Analytics. IBM Cognos Analytics then calls the API of the security system for the information. When a scheduled action is run, the trusted credentials of the user are supplied to IBM Cognos Analytics during the authentication request and result in a `CAMPassport` property being generated. The trusted credentials are modified by IBM Cognos Analytics and result in a Base64-encoded version that contains the encrypted user name and password.

When a user is successfully authenticated using the trusted credentials, a passport ID is generated and stored in the `CAMPassport` property.

If anonymous logon is enabled and the user is logged on anonymously, any requests for trusted credentials result in the generation of trusted credentials representing the anonymous user. For trusted credentials for a named user, the user must be logged on as a named user.

This property

- is of type `string`
is encoded as type `xs:string`

authenticityToken

Specifies a pseudo-random value calculated in the browser by applying the SHA-1 hashing algorithm to the value of the `bibus » CAM » userSessionID` property obtained from the browser cookie.

The value is used by server-side components to thwart cross-site request forgery attacks.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

New in Version 8.3 – “Client Token” on page 1918

This property was added.

CAMCredentialPath

Specifies the search path of a [bibus](#) » [credential](#) object in the content store.

This property

- is of type `string`
is encoded as type `xs:string`

CAMPassport

Specifies the passport for the request. The passport contains information about the authentication of a user. IBM Cognos Analytics creates a passport each time a user logs on. The passport is retained until the session ends. The session can end in one of the following ways:

- When the user logs off
- After a specified period of inactivity
- After a preconfigured time span, regardless of the level of activity

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [CAMPassport](#)
is encoded as type `tns:CAMPassport`

exception

Specifies that a security-related error, or SOAP fault, has occurred. This property is only used in cases in which the body of the response lacks further details about the error.

This property

- is of type [bibus](#) » [CAMException](#)
is encoded as type `tns:CAMException`

roles

Specifies the names of roles to be selected during authentication. This property is used in conjunction with the [bibus](#) » [CAM](#) » [action](#) property with values `logon`, `logonAs`, or `selectRole`.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

securityBlob

Specifies security information, in the form of a binary large object.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

settings

This property

- is of type `bibus » CAMSettings`
is encoded as type `tns:CAMSettings`

New in Version 10.1.0 – “CAM Passport Changes” on page 1864

This property was added.

userSessionID

Specifies a pseudo-random value calculated when the user logs on to IBM Cognos Analytics.

The value of this property is maintained in a browser cookie and is used by browser software to compute the `bibus » CAM » authenticityToken` property value.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

New in Version 8.3 – “Client Token” on page 1918

This property was added.

CAMException

Specifies a class that indicates a security-related error has occurred.

This property is only used in cases in which the body of the response lacks further details about the error.

A security-related error may arise for a number of reasons, such as if incorrect or insufficient authentication information is provided. The error is recoverable if the correct information can be supplied by the user. IBM Cognos Analytics generates a fault with a `CAMAuthUserRecoverable` or `CAMAuthSystemRecoverable` error code for recoverable faults, or a `CAMAuthUnrecoverable` error code for unrecoverable faults. For a `CAMAuthUserRecoverable` fault, the user may be prompted to provide the correct information if using a Web portal.

For example, if a CAM passport times out because it was not referenced for a predefined period of time, IBM Cognos Analytics generates a fault. The `errorCodeString` property element contains information about the exact cause of the failure.

Another example is when the trusted credentials are invalid or they fail the decryption process. An unrecoverable authentication fault is returned, and an error page appears.

References

Used by the following properties:

- `bibus » CAM » exception`

Properties

This class has the following properties.

errorCode

Specifies the error code. This error code must appear on all messages with a severity of `fatal`, `error`, or `warn`.

This property

- is of type `string`
is encoded as type `xs:string`

errorCodeString

Specifies the type of error. The value of this property is readable text.

This property

- is of type `string`
is encoded as type `xs:string`

messages

Contains the messages associated with the error.

This property

- is an array of type `bibus » message`
is encoded as type `tns:messageArray`

promptInfo

Specifies the template used to build a page that prompts the user.

This property

- is of type `bibus » promptInfo`
is encoded as type `tns:promptInfo`

severity

Specifies the minimum severity level of messages displayed to the user.

This property

- is of type `bibus » severityEnum`
is encoded as type `tns:severityEnum`
- has a default value of `error`

CAMPassport

Defines the passport that contains information about the authenticated user.

IBM Cognos Analytics creates a passport each time a user logs on. The passport is retained until the session ends. The session can end in one of the following ways:

- When the user logs off
- After a specified period of inactivity
- After a preconfigured time span, regardless of the level of activity

References

Used by the following properties:

- [bibus](#) » [CAM](#) » [CAMPassport](#)

Properties

This class has the following properties.

authOrigin

Reserved.

This property

- is of type `string`

is encoded as type `xs:string`

New in Version 10.2.0 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1838](#)

This property was added.

New in Version 10.2.1 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1832](#)

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

canCallLogon

This property

- is of type `boolean`

is encoded as type `xs:boolean`

New in Version 10.1.0 – [“CAM Passport Changes” on page 1864](#)

This property was added.

generation

This property

- is of type `int`

is encoded as type `xs:int`

New in Version 10.1.0 – [“CAM Passport Changes” on page 1864](#)

This property was added.

id

Identifies the passport. This identifier is returned after the user's credentials are checked.

When IBM Cognos Analytics receives a request that does not specify a value for this property, it tries to authenticate the user.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`

is encoded as type `xs:string`

isAnonymous

This property

- is of type `boolean`
is encoded as type `xs:boolean`

New in Version 10.1.0 – “CAM Passport Changes” on page 1864

This property was added.

logEnabled

This property

- is of type `boolean`
is encoded as type `xs:boolean`

New in Version 10.1.0 – “CAM Passport Changes” on page 1864

This property was added.

shareable

Reserved.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

CAMProtect

Defines the class that contains information about IBM Cognos Access Manager security controls related to a particular service.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [CAMProtect](#)

Properties

This class has the following properties.

serviceId

Identifies the service protected by IBM Cognos Access Manager.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

trustedValue

Specifies the value for the trusted credential.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

CAMSettings

Defines properties that reflect global IBM Cognos Access Manager configuration information.

References

Used by the following properties:

- [bibus](#) » [CAM](#) » [settings](#)

What's new

New in Version 10.1.0 – “CAM Passport Changes” on page 1864

This class was added.

Properties

This class has the following properties.

anonymousAllowed

Specifies whether the session can authenticate using the Anonymous user.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

namespacesConfigured

This property

- is of type `boolean`
is encoded as type `xs:boolean`

capability

Contains objects that control access to product features.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [capability](#)
- [bibus](#) » [productCapability](#)
- [bibus](#) » [securedFunction](#)

Contained by instances of the following classes

- [bibus](#) » [capability](#)
- [bibus](#) » [root](#)

Properties

This class has the following properties.

items

Contains the user-accessible product features for this instance.

catalog

Contains all [bibus](#) » [catalogFolder](#) class objects in the content store.

This object is created when the content store is initialized, and cannot be deleted by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [catalogFolder](#)

Contained by instances of the following classes

- [bibus](#) » [root](#)

What's new

New in Version 10.2.1 – “Visualization support” on page 1831

This class was added.

Properties

This class has the following properties.

items

Contains [bibus](#) » [catalogFolder](#) class objects.

catalogFolder

Contains objects that store visualization data and Reporting user interface profiles..

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [catalogFolder](#)
- [bibus](#) » [visualization](#)

- [bibus » userInterfaceProfile](#)

Contained by instances of the following classes

- [bibus » catalog](#)
- [bibus » catalogFolder](#)

What's new

New in Version 10.2.2 – “Reporting user interface profiles” on page 1825

This class was updated to include Reporting user interface profiles.

New in Version 10.2.1 – “Visualization support” on page 1831

This class was added.

Properties

This class has the following properties.

items

Contains [bibus » userInterfaceProfile](#) and [bibus » visualization](#). objects.

classEnumArrayProp

Defines the array property class for the [bibus » classEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » classEnum](#)
is encoded as type `tns:classEnumArray`

classEnumProp

Defines the simple property class for the [bibus » classEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » deploymentDetail » deployedObjectClass](#)
- [bibus » baseClass » objectClass](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » classEnum](#)
is encoded as type `tns:classEnum`

configuration

Defines the root object in the configuration hierarchy.

This class defines application-wide configuration data for IBM Cognos components, such as the routing table used by the dispatchers to route a request to the appropriate server for execution.

The single instance of this class in the content store also contains configuration objects such as [bibus » pageDefinition](#) objects, and the user profile [bibus » account](#) object, which initializes user preferences when a user first logs on to IBM Cognos.

This class also defines run-time configuration parameters for the [dispatcher](#), as well as for other services that are managed by the dispatcher.

Values for many of the properties of this class can be acquired by child objects. Property acquisition can be used to simplify the configuration of IBM Cognos installations. Using [bibus » configurationFolder](#) objects can simplify service administration while providing maximum flexibility.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » account](#)
- [bibus » aliasRoot](#)
- [bibus » archiveLocation](#)
- [bibus » configurationFolder](#)
- [bibus » dispatcher](#)
- [bibus » pageDefinition](#)
- [bibus » systemMetricThresholds](#)
- [bibus » uiProfileFolder](#)

Contained by instances of the following classes

- [bibus » root](#)

What's new

New in Version 10.2.0 – [“Support for interactive discovery and visualization” on page 1839](#)

This class was added.

Properties

This class has the following properties.

aasAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasAuditLevel

Specifies the auditing level for the adaptive analytics service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the adaptive analytics service, low affinity requests are used by client applications of IBM® Cognos® Analytic Applications to retrieve metadata and query information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

activeJMXProxyURI

Specifies the URI of the active Java Management Extensions (JMX) proxy.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is read-only

New in Version 10.1.0 – “JMX Proxy Server Scalability” on page 1871

This property was added.

New in Version 10.1.1 – “JMX Proxy Server Scalability” on page 1846

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

Related information:

IBM Cognos IBM Cognos Analytics Administration and Security Guide

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

aliasLocations – obsolete

Contains the alias locations.

New in Version 8.4 – “Package Data Sources” on page 1895

This property was added.

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Administration” on page 1836

This property is deprecated.

ansAnnotationLifetime

Specifies the lifetime of the annotation in XML Schema 1.0 type `xs:duration` form.

The lifetime is the length of time in days after the entry associated with the annotation is deleted. For example, if the lifetime for an annotation is set to 60 days, the annotation is deleted 60 days after the associated report is deleted.

For an annotation object associated with a reportVersion object, the annotations are deleted when the report is deleted.

The default is 180 days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of `P180D`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

ansAuditLevel

Specifies the auditing level for the annotation service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

archiveLocations

Contains the archive locations.

New in Version 8.3 – “Saving Report Output to File System” on page 1901

This property was added.

asAuditLevel

Specifies the auditing level for the agent service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

asConnections

Specifies the maximum number of connections that a process of the agent service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

asMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the agent service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

asPeakConnections

Specifies the number of connections that a agent service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

bpmRestURI

Specifies the REST URI of the IBM Business Process Manager server.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`

New in Version 10.2.0 – “IBM Business Process Server integration” on page 1841

This property was added.

brsAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsAuditLevel

Specifies the auditing level for the batch report service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

brsAuditNativeQuery

Specifies whether to log native query information for the batch report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

brsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 — [“Chart Hotspot Limit” on page 1909](#)

This property was added.

brsDataSourceChange

Specifies the change time of the data source for the batch report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

brsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

brsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the batch report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

brsMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

brsNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

brsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the batch report service.

This property

- is of type `bibus » pdfCharacterEncodingEnum`
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of `auto`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPDFCompressionLevel

Specifies the compression level for PDF documents created by the batch report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

`bibus » configuration » brsPDFCompressionType` property

brsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the batch report service.

This property

- is of type `bibus » pdfCompressionTypeEnum`
 - is encoded as type `tns:pdfCompressionTypeEnumProp`

- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus](#) » [configuration](#) » [brsPDFCompressionLevel](#) property

brsPDFEmbedFonts

Specifies whether the batch report service should embed fonts in generated PDF documents.

This property

- is of type [bibus](#) » [pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

brsPeakAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 — “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsPeakMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

brsPeakNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

capacity

Specifies the processing capacity of a dispatcher.

Use with the `ibus » dispatcher` class to indicate the processing capacity of a dispatcher relative to other dispatchers.

For example, if the first dispatcher is twice as fast as the second, set the capacity of the first to 2.0 and the capacity of the second to 1.0. Incoming requests will be directed to these dispatchers in the same relative proportion (2-to-1). That is, the first dispatcher will receive two-thirds of the requests.

Use with the `ibus » configuration` class and the `ibus » configurationFolder` class to indicate the processing capacity of all dispatchers in the configuration folder.

When you add a dispatcher to the configuration folder, it automatically inherits the configuration settings of the folder. However, if you change the default values of a dispatcher or service, the dispatcher's properties will take precedence over those of the `ibus » configurationFolder` class.

This property

- is of type `float`
is encoded as type `tns:floatProp`

- has a default value of 1.0
- must contain a value greater than 0.0
- can be acquired from a containing object

cmcsAuditLevel

Specifies the auditing level for the content manager cache service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmcsHeapLimit

Specifies the upper limit of the cache size for the Content Manager cache service. This value is represented as a percentage of the JVM heap size.

The default is 10%.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 10
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmsAuditLevel

Specifies the auditing level for the content manager service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

cmsConnections

Specifies the maximum number of connections that a process of the content manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

cmsPeakConnections

Specifies the number of connections that a content manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

collaborationDiscoveryURI

Specifies the discovery URI of the IBM Connections server to use as the collaboration provider.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`

New in Version 10.1.0 – “Collaboration Tool Integration” on page 1867

This property was added.

Related information:

IBM Cognos Administration and Security Guide

configuration – obsolete

cookieCAMPassportHttpOnly

Specifies whether the `HttpOnly` attribute of the `CAM_Passport` session cookie in the response header should be set.

Setting the `HttpOnly` attribute prevents a web browser from being able to read or manipulate the CAM Passport cookie that identifies a user's session to the server. With this attribute set, the browser only uses the cookie to send HTTP requests back to the server, but does not allow scripts to access the contents. Malicious scripts are often inserted into the browser using a Cross Site Scripting (XSS) attack on the same server or web application that the user has authenticated to. If the `HttpOnly` attribute is set, the browser does not allow access to the cookie, preventing such as script from stealing a user's identity.

Administrators who want to enable this feature should ensure that users have a browser that supports the `HttpOnly` attribute.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`

New in Version 10.1.1 – “HttpOnly Attribute Support for the CAM_Passport Cookie” on page 1847

This property was added.

dasAuditLevel

Specifies the auditing level for the data advisor service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Updated Support for IBM Cognos Express” on page 1871

This property was added.

dimsAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsAuditLevel

Specifies the auditing level for the dimensionManagement service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus](#) » [dimensionManagementServiceSpecification](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsPeakAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus](#) » [dimensionManagementServiceSpecification](#) class.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsQueueLimit

Specifies the number of seconds that a request for the dimensionManagement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

disAuditLevel

Specifies the auditing level for the data integration service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

disConnections

Specifies the maximum number of connections that a process of the data integration service can use concurrently to execute requests.

For more information, see “Tune Server Performance” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

dispatcherAuditLevel

Specifies the auditing level for the dispatcher(s) .

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

dispatchers

Contains the dispatchers.

disPeakConnections

Specifies the number of connections that a data integration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

dmsAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsAuditLevel

Specifies the auditing level for the data movement service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

dmsConnections – deprecated

Specifies the maximum number of connections that a process of the data movement service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- is deprecated and will be removed in a future version of the product
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Data Movement Service Configuration Parameters” on page 1917](#)

This property is deprecated. Use the following properties instead:

- [bibus](#) » [configuration](#) » [dmsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsPeakNonAffineConnections](#) property

dmsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 – [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsPeakAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsPeakMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsPeakNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Data Movement Service Configuration Parameters](#)” on page 1917

This property was added.

dmsQueueLimit

Specifies the number of seconds that a request for the data movement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.3 — “[Data Movement Service Configuration Parameters](#)” on page 1917

This property was added.

dsAuditLevel

Specifies the auditing level for the delivery service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

dsCompressAttachmentLimit

Specifies the maximum size, in MB, of an uncompressed email attachment. The delivery service will compress an attachment that is larger than the maximum size before sending it.

Use a value of 0 to disable email attachment compression.

Use a nil value to compress all email attachments.

Setting a non-nil, non-zero value improves performance when sending email messages with large attachments, such as report outputs.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — “Email Delivery Enhancements” on page 1916

This property was added.

dsConnections

Specifies the maximum number of connections that a process of the delivery service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

dsMaximumEMailSize

Specifies the maximum amount of data in MB that can be placed in an email by the delivery service.

Base this value on your mail server configuration. Please contact your mail server administrator to determine an appropriate value. If email sizes exceed that which your mail server is configured to support, consider a different transfer method.

If the uncompressed size of the email exceeds the specified configuration parameter value, the `deliveryService` service removes the largest attachments from the message until the total size of the message is lower than the specified configuration parameter value. If an attachment is removed, an error message is created and added as a plain text attachment to the email indicating that the message size has exceeded the configured limit. If the uncompressed size of the email body (for example, an HTML report) exceeds the configured limit, the error message replaces the content of the body.

Use a value of 0 to specify that the total size of an email is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0

- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

dsPeakConnections

Specifies the number of connections that a delivery service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

editions

Contains information about the software editions available in this installation.

This property

- is an array of type `bibus » softwareEdition`
 - has items that must be of class `bibus » softwareEdition`
 - is encoded as type `tns:softwareEditionArrayProp`

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

emsAuditLevel

Specifies the auditing level for the event management service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

evsAuditLevel

Specifies the auditing level for the EV service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was added.

folders

Contains the configuration folders.

glossaryURI

Specifies the URI of the glossary provider agent.

This property

- is of type anyURI
- is encoded as type `tns:anyURIProp`

New in Version 8.4 – “Support for IBM® WebSphere® Business Glossary” on page 1897

This property was added.

gsAffineConnections

Specifies the number of connections that a graphics service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a graphics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the graphics service.

This property

- is of type `int`
- is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

gsAuditLevel

Specifies the auditing level for the graphics service.

This property

- is of type `bibus » auditLevelEnum`
- is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

gsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

gsMaximumProcesses

Specifies the maximum number of graphics service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

gsNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the graphics service, low affinity requests process requests to generate graphics (charts) for report run requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsPeakAffineConnections

Specifies the number of connections that a graphics service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a graphics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

gsPeakMaximumProcesses

Specifies the maximum number of graphics service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

gsPeakNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsQueueLimit

Specifies the number of seconds that a request for the graphics service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240

- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

htsAuditLevel

Specifies the auditing level for the human task service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – [“Human Task Service” on page 1854](#)

This property was added.

htsCompletedTaskLifetime

Specifies the lifetime of completed human tasks in XML Schema 1.0 type `xs:duration` form. If the lifetime is set to P90D, the human task will be deleted after 90 days if all linked reports or dashboards are deleted.

The default lifetime is ninety days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of P90D
- can be acquired from a containing object

New in Version 10.1.0 – [“Human Task Service” on page 1854](#)

This property was added.

idsAuditLevel

Specifies the auditing level for the index data service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

idsConnections

Specifies the maximum number of connections that a process of the index data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

idsPeakConnections

Specifies the number of connections that a index data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

idVizAuditLevel

Specifies the auditing level for the idViz service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – [“Support for interactive discovery and visualization” on page 1839](#)

This property was added.

issAuditLevel

Specifies the auditing level for the index search service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

issConnections

Specifies the maximum number of connections that a process of the index search service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

issPeakConnections

Specifies the number of connections that a index search service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

iusAuditLevel

Specifies the auditing level for the index update service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

iusConnections

Specifies the maximum number of connections that a process of the index update service can use concurrently to execute requests.

For more information, see “[Tune Server Performance](#)” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

iusPeakConnections

Specifies the number of connections that a index update service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

jmxProxyHostDispatchers

Specifies the dispatchers that can host the Java Management Extensions (JMX) proxy used by IBM Cognos Analytics, in priority order.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [dispatcher](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

New in Version 10.1.0 – [“JMX Proxy Server Scalability” on page 1871](#)

This property was added.

New in Version 10.1.1 – [“JMX Proxy Server Scalability” on page 1846](#)

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

jsAuditLevel

Specifies the auditing level for the job service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

jsConnections

Specifies the maximum number of connections that a process of the job service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

jsmNonPeakDemandBeginHour – obsolete

jsmNonPeakDemandMaximumJobs – obsolete

jsmPeakDemandBeginHour – obsolete

jsmPeakDemandMaximumJobs – obsolete

jsPeakConnections

Specifies the number of connections that a job service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

lastConfigurationModificationTime

Specifies the last time any configuration data was modified, in Coordinated Universal Time (UTC). Also specifies the last time that objects of the `bibus » capability` class and their descendants were modified. Changes made to instances of the `bibus » runTimeState` class that are directly or indirectly contained in an instance of the `bibus » configuration` class do not affect this property.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable

loadBalancingMode

Specifies the load balancing mode for a dispatcher or group of dispatchers.

Use this property to indicate which load balancing algorithm should be used for

- a specific `bibus » dispatcher`
- a group of dispatchers contained within a `bibus » configurationFolder`
- all the dispatchers contained within a `bibus » configuration` object

This property

- is of type `bibus » loadBalancingModeEnum`
 - is encoded as type `tns:loadBalancingModeEnumProp`
- has a default value of `weightedRoundRobin`
- can be acquired from a containing object

IsAuditAdminLevel – obsolete

IsAuditLevel – obsolete

IsAuditNativeQuery – obsolete

IsAuditOtherLevel – obsolete

IsAuditUsageLevel – obsolete

mbsAuditLevel

Specifies the auditing level for the mobile service.

This property

- is of type `bibus » auditLevelEnum`

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

mbsConnections

Specifies the maximum number of connections that a process of the mobile service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

mbsPeakConnections

Specifies the number of connections that a mobile service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

mdsAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The [metadataService](#) does not use this property as this service only processes synchronous requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsAuditLevel

Specifies the auditing level for the metadata service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

mdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no

benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests process lineage requests that process metadata in published Framework Manager models as well as in the query section of a report specification.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsPeakAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsPeakMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsPeakNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsQueueLimit

Specifies the number of seconds that a request for the metadata service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

metadataInformationURI

Specifies the URI of the metadata information provider agent as a relative or an absolute URI.

A URI starting with / is interpreted as a relative URI with the gateway URI being the base URI. Use an absolute URI for an external metadata information provider agent.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- has a default value of `/lineageUIService`

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

misAuditLevel

Specifies the auditing level for the migration service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`

- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

misConnections

Specifies the maximum number of connections that a process of the migration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

misPeakConnections

Specifies the number of connections that a migration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

mmsAuditLevel

Specifies the auditing level for the metrics manager service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mmsConnections

Specifies the maximum number of connections that a process of the metrics manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

mmsPeakConnections

Specifies the number of connections that a metrics manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

mobileConfiguration

Reserved.

This property

- is an array of type `bibus » stringMapEntry`
 - is encoded as type `tns:stringMapEntryArrayProp`

New in Version 10.2.1 – “Administrative changes for Cognos Analytics Mobile Reports” on page 1830

This property was added.

msAuditLevel

Specifies the auditing level for the monitor service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

nonPeakDemandBeginHour

Specifies the hour of the day at which the non-peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 18
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 23

- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

overrideOptions

Constrains the set of possible option values that can be used when running a request. To constrain the option values that the server may use, you must add the option to this property and specify the acceptable values.

When an option has values specified in this property, a run request can only use option values that are specified in this property for the option. If a run request uses an option that is specified in this property, and an option value specified in the run request is not included in the option values specified in this property, the system returns a fault.

If the system default value is not included in the set of values for an option in this property, the default value is no longer valid. You must then set a new default value in the [bibus » configuration » serviceDefaultOptions](#) property that is one of the option values specified in this property. For example, the default value for the [bibus » systemOptionEnum » accessibilityFeatures](#) value is `false`. If you set the [bibus » systemOptionEnum » accessibilityFeatures](#) value in this property to `true`, you should add the value `true` to the [bibus » systemOptionEnum » accessibilityFeatures](#) value in the [bibus » configuration » serviceDefaultOptions](#) property

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

New in Version 10.1.0 – “Accessibility” on page 1859

This property was added.

pacsauditLevel

Specifies the auditing level for the planning administration console service.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

pacsaConnections

Specifies the maximum number of connections that a process of the planning administration console service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

pacPeakConnections

Specifies the number of connections that a planning administration console service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pageDefinitions

Contains page definitions.

pdsAuditLevel

Specifies the auditing level for the planning data service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

pdsConnections

Specifies the maximum number of connections that a process of the planning data service can use concurrently to execute requests.

For more information, see “Tune Server Performance” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsEListAccessCacheLimit

Specifies the number of seconds that an e-list access rights cache entry can remain in the cache before it must be recalculated.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 3600
- must contain a value greater than or equal to 0

- can be acquired from a containing object

pdsMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsPeakConnections

Specifies the number of connections that a planning data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsPeakMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsShowCellAnnotations

Specifies whether authoring studios should display cell annotations.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

peakDemandBeginHour

Specifies the hour of the day at which the peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 7
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 23
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

periodicalDocumentVersionRetentionAge

Specifies the default maximum age of `bibus » documentVersion` objects to be retained in a `bibus » periodical bibus » document`. This value is used to construct a retention rule for new documents in periodicals.

Default: 1 day

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of P1D

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

periodicalDocumentVersionRetentionCount

Specifies the default maximum number of `bibus » documentVersion` objects to be retained in a `bibus » periodical bibus » document`. This value is used to construct a retention rule for new documents in periodicals.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 0

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

ppsAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to retrieve saved output, and also when a request must go to a particular instance of `powerPlayService` service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsAuditLevel

Specifies the auditing level for the PowerPlay service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the PowerPlay service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of `0` to specify that the size of an attachment is unlimited.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

ppsNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute a report, and also when a request can be sent to any instance of `powerPlayService` service.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

ppsPeakAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

ppsPeakNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

ppsQueueLimit

Specifies the number of seconds that a request for the PowerPlay service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

prsAuditLevel

Specifies the auditing level for the planning runtime service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

prsConnections

Specifies the maximum number of connections that a process of the planning runtime service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

prsPeakConnections

Specifies the number of connections that a planning runtime service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

psAuditLevel

Specifies the auditing level for the presentation service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

ptsAuditLevel

Specifies the auditing level for the planning task service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

ptsConnections

Specifies the maximum number of connections that a process of the planning task service can use concurrently to execute requests.

For more information, see “[Tune Server Performance](#)” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

ptsPeakConnections

Specifies the number of connections that a planning task service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

qsAdditionalJVMArguments

Specifies additional arguments that control the JVM. The arguments may vary depending on the JVM.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsAuditLevel

Specifies the auditing level for the query service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsDiagnosticsEnabled – deprecated

Specifies whether diagnostics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property is deprecated.

qsDisableQueryPlanCache – deprecated

Specifies whether query plans are cached for possible re-use. A query plan represents a set of transformations applied to query data objects to obtain desired query results.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

New in Version 10.1.1 – “Deprecation of qsDisableQueryPlanCache” on page 1849

This property is deprecated.

qsDisableVerboseGCLogging

Specifies whether garbage collection information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property was added.

qsDumpModelToFile

Specifies whether the model is dumped to a file when a query is run.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsGCPolicy

Specifies the garbage collection policy used to manage JVM heap storage.

Default: Generational

This property

- is of type `bibus » gcPolicyEnum`

is encoded as type `tns:anyURIProp`

- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsGenerateCommentsInNativeSQL

Specifies whether comments in native SQL are recorded in a log file.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsIdleConnectionTimeout

Specifies the timeout period, in seconds, for an idle database connection.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of `300`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsInitialJVMHeapSize

Specifies the initial size, in MB, of the JVM heap.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of `1024`
- must contain a value greater than `0`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsInitialJVMMemorySize

Specifies the initial nursery size, in MB, of the JVM. A value of zero indicates that the initial nursery size is system managed.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of `0`
- must contain a value greater than or equal to `0`

- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsJVMHeapSizeLimit

Specifies the maximum size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsJVMNurserySizeLimit

Specifies the maximum size, in MB, of the JVM nursery. A value of zero indicates that the nursery size limit is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsManualCubeStart

Specifies whether ROLAP cubes are started manually.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsMetricsEnabled – deprecated

Specifies whether metrics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`

- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsMultiDimensionalQuerySizeLimit

Specifies memory size, in MB, allotted to each locally executed multi-dimensional query. A value of 0 means the memory size is limited to available system memory.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836](#)

This property was added.

qsQueryExecutionTrace

Specifies whether information tracing the execution of queries is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsQueryPlanningTrace

Specifies whether information tracing the development of query plans is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsResultSetCacheQueryTimeThreshold

Specifies the minimum time, in milliseconds, that must elapse during the construction of a result set before it is considered for caching.

Important: The unit of measurement for this configuration parameter is *milliseconds* (1/1000th of a second) whereas other configuration parameters that specify a duration use *seconds*.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 1000
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsROLAPCubeAdministrationCommandTimeout

Specifies the time, in seconds, that the `queryService` will wait for internal resources to become available while executing a ROLAP cube administration command. If the specified timeout period has elapsed without resources becoming available, the `queryService` will time out.

Use a value of 0 when you do not want the `queryService` to time out.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 120
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsROLAPCubeConfigurations

Defines configuration data for ROLAP cubes.

This property

- is an array of type `bibus » baseROLAPCubeConfiguration`
is encoded as type `tns:baseROLAPCubeConfigurationArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsROLAPMemberCacheAliasRoot

Reserved.

This property

- is of type `token`
is encoded as type `tns:tokenProp`
- can be acquired from a containing object

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Administration” on page 1836

This property was added.

qsVerboseGCLogLimit

Specifies the maximum number of JVM garbage collection cycles that are logged when `bibus » configuration » qsDisableVerboseGCLogging` is enabled.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of `1000`
- must contain a value greater than `0`
- can be acquired from a containing object

New in Version 10.2.0 – “[New queryService configuration options](#)” on page 1837

This property was added.

rdsAuditLevel

Specifies the auditing level for the report data service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

rdsGatewayMappings

Contains the mappings between the internal and external PowerPlay gateways.

This property

- is an array of type `bibus » gatewayMapping`
is encoded as type `tns:gatewayMappingArrayProp`
- can be acquired from a containing object

rdsMaximumDataSize

Specifies the maximum amount of data that can be read from a content provider in MB.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of `10`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

reposAuditLevel

Specifies the auditing level for the repository service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – “[Repository service](#)” on page 1840

This property was added.

reposCacheObjTTL

Reserved.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- has a default value of 1200
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposNumObjDisk

Reserved.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- has a default value of 1000
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposNumObjMem

Reserved.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- has a default value of 100
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

rmdsAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsAuditLevel

Specifies the auditing level for the relational metadata service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

rmdsConnections

Specifies the maximum number of connections that a process of the relational metadata service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Relational metadata service” on page 1835

This property was added.

rmdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

rmdsNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no

benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Writer comment

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsPeakAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsPeakConnections

Specifies the number of connections that a relational metadata service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – [“Relational metadata service” on page 1835](#)

This property was added.

rmdsPeakNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

routingTable

Contains the routing table entries used by the dispatchers to route requests to the appropriate server group.

This property

- is an array of type `bibus » routingTableEntry`
 - is encoded as type `tns:routingTableEntryArrayProp`

rsAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsAuditLevel

Specifies the auditing level for the report service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

rsAuditNativeQuery

Specifies whether to log native query information for the report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

rsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 — [“Chart Hotspot Limit” on page 1909](#)

This property was added.

rsDataSourceChange

Specifies the change time of the data source for the report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

rsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

rsMaximumEmailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

rsMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

rsNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the report service.

This property

- is of type `bibus » pdfCharacterEncodingEnum`
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of `auto`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

rsPDFCompressionLevel

Specifies the compression level for PDF documents created by the report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

`bibus » configuration » rsPDFCompressionType` property

rsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the report service.

This property

- is of type `bibus » pdfCompressionTypeEnum`
 - is encoded as type `tns:pdfCompressionTypeEnumProp`

- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus](#) » [configuration](#) » [rsPDFCompressionLevel](#) property

rsPDFEmbedFonts

Specifies whether the report service should embed fonts in generated PDF documents.

This property

- is of type [bibus](#) » [pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

rsPeakAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 — “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsPeakMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

rsPeakNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsQueueLimit

Specifies the number of seconds that a request for the report service can be queued before it exceeds the timeout period.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

saCAMAuditLevel

Specifies the auditing level for the saCAM service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1838

This property was added.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

serverGroup

Specifies the server group associated with this object.

Use this property to identify groups of dispatchers in different application server clusters. Dispatchers that share the same value for this property act as a single cluster.

By default, all dispatchers registered in the same content store act as a single, load-balancing cluster. If your installation uses application server clusters, use this property to identify members of different clusters. For example, if you need some of your dispatchers to operate in `weightedRoundRobin` mode while others operate in `clusterCompatible` mode, use the `serverGroup` property to distinguish these groupings. When you group your dispatchers using this property, requests are only forwarded to a dispatcher that is part of the same server group.

You must use this setting in conjunction with the values you have set in the `bibus » loadBalancingModeEnum` enumeration set to properly contain your dispatchers in groups that use the same load-balancing algorithm. Failure to meet this requirement will result in unpredictable forwarding of requests by dispatchers.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- can be acquired from a containing object

serviceDefaultOptions

Specifies default option values for some IBM Cognos Analytics services.

Use this property to change the default option values for services. Options specified in this property have a lower priority than the `bibus » account » options` property, but a higher priority than the built-in default option values for the service. For more information, see [“How IBM Cognos Determines Search Order When Building a Request” on page 68](#) and [“Constraining Option Values and Setting Service Default Values” on page 69](#).

This property

- is an array of type `bibus » option`
 - is encoded as type `tns:optionArrayProp`

New in Version 10.1.0 – “Accessibility” on page 1859

This property was added.

ssAuditLevel

Specifies the auditing level for the system service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`

- has a default value of [minimal](#)
- can be acquired from a containing object

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

temporaryObjectLifetime

Specifies the (approximate) length of time that temporary objects are stored in the file system. This property is ignored when the storage location for temporary objects is defined as the content store.

The default value is four hours (PT4H).

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of PT4H

New in Version 10.1.0 – “Storing Temporary Output Objects Outside the Content Store” on page 1857

This property was added.

temporaryObjectLocation

Specifies the location for storing temporary objects created by IBM Cognos services.

This property

- is of type [bibus » temporaryObjectLocationEnum](#)
 - is encoded as type `tns:anyURIProp`
- has a default value of [serverFileSystem](#)

New in Version 10.1.0 – “Storing Temporary Output Objects Outside the Content Store” on page 1857

This property was added.

unixRepositoryURI

Defines the location where report output is stored on a UNIX[®] operating system when an external repository is used instead of the content store.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is read-only

New in Version 10.2.1 – “External object store for report archiving” on page 1829

This property was added.

userProfile

Contains the default account as populated during the Content Manager initialization process.

The default account specifies initial settings, such as user preferences, for a new user.

This property

- must have at most 1 item

windowsRepositoryURI

Defines the location where report output is stored on a Microsoft® Windows operating system when an external repository is used instead of the content store.

This property

- is of type anyURI
- is encoded as type `tns:anyURIProp`

New in Version 10.2.1 – “External object store for report archiving” on page 1829

This property was added.

configurationData

Defines the configuration data returned by the `system » getConfiguration(properties)` method.

References

Used by the following method return values:

- `system » getConfiguration(properties) » result`

Properties

This class has the following properties.

bpmRestURI

Specifies the URI of the BPM server.

This property

- is an array of type anyURI
- is encoded as type `tns:anyURIArray`
- is read-only
 - must have at most 1 item

New in Version 10.2.0 – “IBM Business Process Server integration” on page 1841

This property was added.

contentLocaleMap

Contains the mappings that determine which locale will be used for the content of reports, prompts, data, and metadata. The input to these mappings is the user preference `contentLocale`. The mappings can be changed during product configuration.

This property

- is an array of type `bibus » localeMapEntry`
- is encoded as type `tns:localeMapEntryArray`
- is read-only

defaultFont

Specifies the default font.

To display output correctly, fonts must be available where a report or chart is rendered. In the case of charts and PDF reports, the fonts must be installed on the IBM Cognos Analytics server. For example,

if a Reporting user selects the Arial font for a report, Arial must be installed on the IBM Cognos Analytics server to properly render charts and PDF files. If a requested font is not available, IBM Cognos Analytics renders PDF files and charts using the default font.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArray`
- is read-only
- must have at most 1 item

Related information:

IBM Cognos Analytics *Administration and Security Guide*

glossaryURI

Specifies the URI of the glossary provider agent.

This property

- is an array of type `anyURI`
 - is encoded as type `tns:anyURIArray`
- is read-only
- must have at most 1 item

New in Version 8.4 — “[Support for IBM® WebSphere® Business Glossary](#)” on page 1897

This property was added.

metadataInformationURI

Specifies the URI of the metadata information provider agent as a relative or an absolute URI.

A URI starting with / is interpreted as a relative URI with the gateway URI being the base URI. Use an absolute URI for an external metadata information provider agent.

This property

- is an array of type `anyURI`
 - is encoded as type `tns:anyURIArray`
- is read-only
- must have at most 1 item

New in Version 8.4 — “[Lineage Metadata](#)” on page 1891

This property was added.

productLocaleMap

Contains the mappings that determine which locale will be used for the IBM Cognos Analytics user interface. The input to these mappings is the user preference `productLocale`. The mappings can be changed during product configuration.

This property

- is an array of type `bibus » localeMapEntry`
 - is encoded as type `tns:localeMapEntryArray`
- is read-only

serverLocale

Contains the IBM Cognos Analytics server locale, which determines the language used for the event log.

This property

- is an array of type [bibus](#) » [locale](#)
is encoded as type `tns:localeArray`
- is read-only
- must have at most 1 item

serverTimeZone

Contains the standard ICU time zone ID used by the IBM Cognos Analytics servers.

This property

- is an array of type [bibus](#) » [timeZone](#)
is encoded as type `tns:timeZoneArray`
- is read-only
- must have at most 1 item

supportedContentLocales

Contains the supported content locales.

This property

- is an array of type [bibus](#) » [locale](#)
is encoded as type `tns:localeArray`
- is read-only

supportedCurrencies

Contains the supported currencies.

This property

- is an array of type [bibus](#) » [currency](#)
is encoded as type `tns:currencyArray`
- is read-only

supportedFonts

Contains the list of supported fonts.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`
- is read-only

supportedProductLocales

Contains the supported product locales.

This property

- is an array of type [bibus](#) » [locale](#)
is encoded as type `tns:localeArray`
- is read-only

timeZones

Contains the set of time zones available for use by the IBM Cognos Analytics servers.

This property

- is an array of type [bibus](#) » [timeZone](#)

is encoded as type `tns:timeZoneArray`

- is read-only

configurationFolder

Defines a configuration folder.

Use this class to organize `bibus » dispatcher` objects into hierarchies to simplify the configuration of IBM Cognos Analytics. This class defines run-time configuration parameters for the `dispatcher`, as well as other services that are managed by the dispatcher.

Values for many of the properties of this class can be acquired from its parent object. In addition, values for these properties can be acquired by its child objects. You can take advantage of property acquisition to simplify the configuration of IBM Cognos Analytics installations.

For example, you can share a common configuration for five dispatchers by placing the `bibus » dispatcher` objects for those dispatchers into the same `configurationFolder`, so that the property values from the `configurationFolder` object are acquired by the five dispatchers contained within it.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contains instances of the following classes

- `bibus » configurationFolder`
- `bibus » dispatcher`

Contained by instances of the following classes

- `bibus » configuration`
- `bibus » configurationFolder`

What's new

New in Version 10.2.0 – [“Support for interactive discovery and visualization” on page 1839](#)

This class was added.

Properties

This class has the following properties.

aasAffineConnections

Specifies the number of connections that an adaptive analytics service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by an adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasAuditLevel

Specifies the auditing level for the adaptive analytics service.

This property

- is of type `bibus » auditLevelEnum`

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the adaptive analytics service, low affinity requests are used by client applications of IBM® Cognos® Analytic Applications to retrieve metadata and query information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

ansAnnotationLifetime

Specifies the lifetime of the annotation in XML Schema 1.0 type `xs:duration` form.

The lifetime is the length of time in days after the entry associated with the annotation is deleted. For example, if the lifetime for an annotation is set to 60 days, the annotation is deleted 60 days after the associated report is deleted.

For an annotation object associated with a reportVersion object, the annotations are deleted when the report is deleted.

The default is 180 days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of P180D

- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

ansAuditLevel

Specifies the auditing level for the annotation service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

asAuditLevel

Specifies the auditing level for the agent service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

asConnections

Specifies the maximum number of connections that a process of the agent service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

asMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the agent service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

asPeakConnections

Specifies the number of connections that a agent service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

brsAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsAuditLevel

Specifies the auditing level for the batch report service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

brsAuditNativeQuery

Specifies whether to log native query information for the batch report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

brsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 — “Chart Hotspot Limit” on page 1909

This property was added.

brsDataSourceChange

Specifies the change time of the data source for the batch report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

brsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

brsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the batch report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

brsMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

brsNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

brsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the batch report service.

This property

- is of type `bibus » pdfCharacterEncodingEnum`
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of `auto`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPDFCompressionLevel

Specifies the compression level for PDF documents created by the batch report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

`bibus » configurationFolder » brsPDFCompressionType` property

brsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the batch report service.

This property

- is of type `bibus » pdfCompressionTypeEnum`
 - is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of `classic`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus](#) » [configurationFolder](#) » [brsPDFCompressionLevel](#) property

brsPDFEmbedFonts

Specifies whether the batch report service should embed fonts in generated PDF documents.

This property

- is of type [bibus](#) » [pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPeakAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsPeakMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

brsPeakNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

capacity

Specifies the processing capacity of a dispatcher.

Use with the `bibus » dispatcher` class to indicate the processing capacity of a dispatcher relative to other dispatchers.

For example, if the first dispatcher is twice as fast as the second, set the capacity of the first to 2.0 and the capacity of the second to 1.0. Incoming requests will be directed to these dispatchers in the same relative proportion (2-to-1). That is, the first dispatcher will receive two-thirds of the requests.

Use with the `bibus » configuration` class and the `bibus » configurationFolder` class to indicate the processing capacity of all dispatchers in the configuration folder.

When you add a dispatcher to the configuration folder, it automatically inherits the configuration settings of the folder. However, if you change the default values of a dispatcher or service, the dispatcher's properties will take precedence over those of the `bibus » configurationFolder` class.

This property

- is of type `float`

is encoded as type `tns:floatProp`

- has a default value of 1.0

- must contain a value greater than 0.0
- can be acquired from a containing object

cmcsAuditLevel

Specifies the auditing level for the content manager cache service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmcsHeapLimit

Specifies the upper limit of the cache size for the Content Manager cache service. This value is represented as a percentage of the JVM heap size.

The default is 10%.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmsAuditLevel

Specifies the auditing level for the content manager service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

cmsConnections

Specifies the maximum number of connections that a process of the content manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

cmsPeakConnections

Specifies the number of connections that a content manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

configuration – obsolete

dasAuditLevel

Specifies the auditing level for the data advisor service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Updated Support for IBM Cognos Express” on page 1871

This property was added.

dimsAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsAuditLevel

Specifies the auditing level for the dimensionManagement service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus » dimensionManagementServiceSpecification](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus » dimensionManagementServiceSpecification](#) class.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “[Dimension Management Service](#)” on page 1896

This property was added.

dimsQueueLimit

Specifies the number of seconds that a request for the dimensionManagement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 – “[Dimension Management Service](#)” on page 1896

This property was added.

disAuditLevel

Specifies the auditing level for the data integration service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

disConnections

Specifies the maximum number of connections that a process of the data integration service can use concurrently to execute requests.

For more information, see “[Tune Server Performance](#)” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

dispatcherAuditLevel

Specifies the auditing level for the dispatcher(s) .

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

dispatchers

Contains the IBM Cognos Analytics dispatchers.

disPeakConnections

Specifies the number of connections that a data integration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

dmsAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsAuditLevel

Specifies the auditing level for the data movement service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

dmsConnections – deprecated

Specifies the maximum number of connections that a process of the data movement service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- is deprecated and will be removed in a future version of the product
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property is deprecated. Use the following properties instead:

- [bibus » configurationFolder » dmsAffineConnections](#) property
- [bibus » configurationFolder » dmsNonAffineConnections](#) property
- [bibus » configurationFolder » dmsPeakAffineConnections](#) property
- [bibus » configurationFolder » dmsPeakNonAffineConnections](#) property

dmsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsPeakAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsPeakMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsPeakNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a `bibus » dataMovementServiceSpecification` or a `bibus » dataMovementTask`.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsQueueLimit

Specifies the number of seconds that a request for the data movement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dsAuditLevel

Specifies the auditing level for the delivery service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

dsCompressAttachmentLimit

Specifies the maximum size, in MB, of an uncompressed email attachment. The delivery service will compress an attachment that is larger than the maximum size before sending it.

Use a value of 0 to disable email attachment compression.

Use a `nil` value to compress all email attachments.

Setting a non-`nil`, non-zero value improves performance when sending email messages with large attachments, such as report outputs.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — “Email Delivery Enhancements” on page 1916

This property was added.

dsConnections

Specifies the maximum number of connections that a process of the delivery service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

dsMaximumEMailSize

Specifies the maximum amount of data in MB that can be placed in an email by the delivery service.

Base this value on your mail server configuration. Please contact your mail server administrator to determine an appropriate value. If email sizes exceed that which your mail server is configured to support, consider a different transfer method.

If the uncompressed size of the email exceeds the specified configuration parameter value, the [deliveryService](#) service removes the largest attachments from the message until the total size of the message is lower than the specified configuration parameter value. If an attachment is removed, an error message is created and added as a plain text attachment to the email indicating that the

message size has exceeded the configured limit. If the uncompressed size of the email body (for example, an HTML report) exceeds the configured limit, the error message replaces the content of the body.

Use a value of 0 to specify that the total size of an email is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – [“New Email Configuration Parameters” on page 1868](#)

This property was added.

dsPeakConnections

Specifies the number of connections that a delivery service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

emsAuditLevel

Specifies the auditing level for the event management service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

evsAuditLevel

Specifies the auditing level for the EV service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

This property was added.

folders

Contains the IBM Cognos Analytics folder hierarchy.

gsAuditLevel

Specifies the auditing level for the graphics service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

gsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

gsNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the graphics service, low affinity requests process requests to generate graphics (charts) for report run requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsPeakNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsQueueLimit

Specifies the number of seconds that a request for the graphics service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

htsAuditLevel

Specifies the auditing level for the human task service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

htsCompletedTaskLifetime

Specifies the lifetime of completed human tasks in XML Schema 1.0 type `xs:duration` form. If the lifetime is set to P90D, the human task will be deleted after 90 days if all linked reports or dashboards are deleted.

The default lifetime is ninety days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of P90D
- can be acquired from a containing object

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

idsAuditLevel

Specifies the auditing level for the index data service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

idsConnections

Specifies the maximum number of connections that a process of the index data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

idsPeakConnections

Specifies the number of connections that a index data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

idVizAuditLevel

Specifies the auditing level for the idViz service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.2.0 – “[Support for interactive discovery and visualization](#)” on page 1839

This property was added.

issAuditLevel

Specifies the auditing level for the index search service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

issConnections

Specifies the maximum number of connections that a process of the index search service can use concurrently to execute requests.

For more information, see “[Tune Server Performance](#)” on page 89.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

issPeakConnections

Specifies the number of connections that a index search service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

iusAuditLevel

Specifies the auditing level for the index update service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

iusConnections

Specifies the maximum number of connections that a process of the index update service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

iusPeakConnections

Specifies the number of connections that a index update service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

jsAuditLevel

Specifies the auditing level for the job service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

jsConnections

Specifies the maximum number of connections that a process of the job service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

jsmNonPeakDemandBeginHour – obsolete

jsmNonPeakDemandMaximumJobs – obsolete

jsmPeakDemandBeginHour – obsolete

jsmPeakDemandMaximumJobs – obsolete

jsPeakConnections

Specifies the number of connections that a job service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

loadBalancingMode

Specifies the load balancing mode for a dispatcher or group of dispatchers.

Use this property to indicate which load balancing algorithm should be used for

- a specific [bibus » dispatcher](#)
- a group of dispatchers contained within a [bibus » configurationFolder](#)
- all the dispatchers contained within a [bibus » configuration](#) object

This property

- is of type [bibus » loadBalancingModeEnum](#)
 - is encoded as type `tns:loadBalancingModeEnumProp`
- has a default value of [weightedRoundRobin](#)
- can be acquired from a containing object

lsAuditAdminLevel – obsolete

lsAuditLevel – obsolete

IsAuditNativeQuery – obsolete

IsAuditOtherLevel – obsolete

IsAuditUsageLevel – obsolete

mbsAuditLevel

Specifies the auditing level for the mobile service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mbsConnections

Specifies the maximum number of connections that a process of the mobile service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

mbsPeakConnections

Specifies the number of connections that a mobile service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

mbsAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The `metadataService` does not use this property as this service only processes synchronous requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsAuditLevel

Specifies the auditing level for the metadata service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

mdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `2`
- must contain a value greater than or equal to `1`

- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests process lineage requests that process metadata in published Framework Manager models as well as in the query section of a report specification.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsPeakAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsPeakMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsPeakNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsQueueLimit

Specifies the number of seconds that a request for the metadata service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

misAuditLevel

Specifies the auditing level for the migration service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 — “Migration Service” on page 1883

This property was added.

misConnections

Specifies the maximum number of connections that a process of the migration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

misPeakConnections

Specifies the number of connections that a migration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

mmsAuditLevel

Specifies the auditing level for the metrics manager service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

mmsConnections

Specifies the maximum number of connections that a process of the metrics manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

mmsPeakConnections

Specifies the number of connections that a metrics manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

msAuditLevel

Specifies the auditing level for the monitor service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

nonPeakDemandBeginHour

Specifies the hour of the day at which the non-peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 18
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 23
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pacsAuditLevel

Specifies the auditing level for the planning administration console service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

pacsConnections

Specifies the maximum number of connections that a process of the planning administration console service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

pacsPeakConnections

Specifies the number of connections that a planning administration console service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

pdsAuditLevel

Specifies the auditing level for the planning data service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

pdsConnections

Specifies the maximum number of connections that a process of the planning data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsEListAccessCacheLimit

Specifies the number of seconds that an e-list access rights cache entry can remain in the cache before it must be recalculated.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 3600
- must contain a value greater than or equal to 0
- can be acquired from a containing object

pdsMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsPeakConnections

Specifies the number of connections that a planning data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsPeakMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsShowCellAnnotations

Specifies whether authoring studios should display cell annotations.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

peakDemandBeginHour

Specifies the hour of the day at which the peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 7
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 23
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

ppsAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to retrieve saved output, and also when a request must go to a particular instance of `powerPlayService` service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsAuditLevel

Specifies the auditing level for the PowerPlay service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the PowerPlay service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 — “New Email Configuration Parameters” on page 1868

This property was added.

ppsNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute a report, and also when a request can be sent to any instance of `powerPlayService` service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsPeakAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsPeakNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

ppsQueueLimit

Specifies the number of seconds that a request for the PowerPlay service can be queued before it exceeds the timeout period.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

prsAuditLevel

Specifies the auditing level for the planning runtime service.

This property

- is of type `bibus » auditLevelEnum`

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

prsConnections

Specifies the maximum number of connections that a process of the planning runtime service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

prsPeakConnections

Specifies the number of connections that a planning runtime service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

psAuditLevel

Specifies the auditing level for the presentation service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

ptsAuditLevel

Specifies the auditing level for the planning task service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

ptsConnections

Specifies the maximum number of connections that a process of the planning task service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

ptsPeakConnections

Specifies the number of connections that a planning task service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

qsAdditionalJVMArguments

Specifies additional arguments that control the JVM. The arguments may vary depending on the JVM.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsAuditLevel

Specifies the auditing level for the query service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsDiagnosticsEnabled – deprecated

Specifies whether diagnostics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsDisableQueryPlanCache – deprecated

Specifies whether query plans are cached for possible re-use. A query plan represents a set of transformations applied to query data objects to obtain desired query results.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product

- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.1.1 – [“Deprecation of `qsDisableQueryPlanCache`” on page 1849](#)

This property is deprecated.

`qsDisableVerboseGCLogging`

Specifies whether garbage collection information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – [“New `queryService` configuration options” on page 1837](#)

This property was added.

`qsDumpModelToFile`

Specifies whether the model is dumped to a file when a query is run.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

`qsGCPolicy`

Specifies the garbage collection policy used to manage JVM heap storage.

Default: [Generational](#)

This property

- is of type `bibus » gcPolicyEnum`
 - is encoded as type `tns:anyURIProp`
- can be acquired from a containing object

New in Version 10.2.0 – [“New `queryService` configuration options” on page 1837](#)

This property was added.

`qsGenerateCommentsInNativeSQL`

Specifies whether comments in native SQL are recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`

- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsIdleConnectionTimeout

Specifies the timeout period, in seconds, for an idle database connection.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 300
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsInitialJVMHeapSize

Specifies the initial size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsInitialJVMMemorySize

Specifies the initial nursery size, in MB, of the JVM. A value of zero indicates that the initial nursery size is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsJVMHeapSizeLimit

Specifies the maximum size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0

- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsJVMMemorySizeLimit

Specifies the maximum size, in MB, of the JVM nursery. A value of zero indicates that the nursery size limit is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- must contain a value greater than or equal to `0`
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsManualCubeStart

Specifies whether ROLAP cubes are started manually.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsMetricsEnabled – deprecated

Specifies whether metrics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsMultiDimensionalQuerySizeLimit

Specifies memory size, in MB, allotted to each locally executed multi-dimensional query. A value of 0 means the memory size is limited to available system memory.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836](#)

This property was added.

qsQueryExecutionTrace

Specifies whether information tracing the execution of queries is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsQueryPlanningTrace

Specifies whether information tracing the development of query plans is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsResultSetCacheQueryTimeThreshold

Specifies the minimum time, in milliseconds, that must elapse during the construction of a result set before it is considered for caching.

Important: The unit of measurement for this configuration parameter is *milliseconds* (1/1000th of a second) whereas other configuration parameters that specify a duration use *seconds*.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 1000
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPCubeAdministrationCommandTimeout

Specifies the time, in seconds, that the `queryService` will wait for internal resources to become available while executing a ROLAP cube administration command. If the specified timeout period has elapsed without resources becoming available, the `queryService` will time out.

Use a value of 0 when you do not want the `queryService` to time out.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 120
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPCubeConfigurations

Defines configuration data for ROLAP cubes.

This property

- is an array of type `bibus » baseROLAPCubeConfiguration`
 - is encoded as type `tns:baseROLAPCubeConfigurationArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPMemberCacheAliasRoot

Reserved.

This property

- is of type `token`
 - is encoded as type `tns:tokenProp`
- can be acquired from a containing object

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Administration” on page 1836](#)

This property was added.

qsVerboseGCLogLimit

Specifies the maximum number of JVM garbage collection cycles that are logged when `bibus » configurationFolder » qsDisableVerboseGCLogging` is enabled.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1000
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

rdsAuditLevel

Specifies the auditing level for the report data service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

rdsGatewayMappings

Contains the mappings between the internal and external PowerPlay gateways.

This property

- is an array of type [bibus » gatewayMapping](#)
 - is encoded as type `tns:gatewayMappingArrayProp`
- can be acquired from a containing object

rdsMaximumDataSize

Specifies the maximum amount of data that can be read from a content provider in MB.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- must contain a value greater than or equal to 1
- can be acquired from a containing object

reposAuditLevel

Specifies the auditing level for the repository service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposCacheObjTTL

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 1200
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposNumObjDisk

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of `1000`
- can be acquired from a containing object

New in Version 10.2.0 – “[Repository service](#)” on page 1840

This property was added.

reposNumObjMem

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of `100`
- can be acquired from a containing object

New in Version 10.2.0 – “[Repository service](#)” on page 1840

This property was added.

rmdsAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “[Relational Metadata Service](#)” on page 1858

This property was added.

rmdsAuditLevel

Specifies the auditing level for the relational metadata service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`

- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmadsConnections

Specifies the maximum number of connections that a process of the relational metadata service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – [“Relational metadata service” on page 1835](#)

This property was added.

rmadsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmadsNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Writer comment

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmadsPeakAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmadsPeakConnections

Specifies the number of connections that a relational metadata service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – [“Relational metadata service” on page 1835](#)

This property was added.

rmadsPeakNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

rsAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsAuditLevel

Specifies the auditing level for the report service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

rsAuditNativeQuery

Specifies whether to log native query information for the report service.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`

- can be acquired from a containing object

rsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a nil value to generate all hotspots in a chart.

Setting a non-nil value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 – “Chart Hotspot Limit” on page 1909

This property was added.

rsDataSourceChange

Specifies the change time of the data source for the report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

rsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

rsMaximumEmailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the report service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value

of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

rsMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

rsNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the report service.

This property

- is of type [bibus » pdfCharacterEncodingEnum](#)
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of [auto](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

rsPDFCompressionLevel

Specifies the compression level for PDF documents created by the report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » configurationFolder » rsPDFCompressionType](#) property

rsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the report service.

This property

- is of type [bibus » pdfCompressionTypeEnum](#)
 - is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » configurationFolder » rsPDFCompressionLevel](#) property

rsPDFEmbedFonts

Specifies whether the report service should embed fonts in generated PDF documents.

This property

- is of type [bibus » pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

rsPeakAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsPeakMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

rsPeakNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no

benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsQueueLimit

Specifies the number of seconds that a request for the report service can be queued before it exceeds the timeout period.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

saCAMAuditLevel

Specifies the auditing level for the saCAM service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1838

This property was added.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

serverGroup

Specifies the server group associated with this object.

Use this property to identify groups of dispatchers in different application server clusters. Dispatchers that share the same value for this property act as a single cluster.

By default, all dispatchers registered in the same content store act as a single, load-balancing cluster. If your installation uses application server clusters, use this property to identify members of different clusters. For example, if you need some of your dispatchers to operate in [weightedRoundRobin](#) mode while others operate in [clusterCompatible](#) mode, use the `serverGroup` property to distinguish these groupings. When you group your dispatchers using this property, requests are only forwarded to a dispatcher that is part of the same server group.

You must use this setting in conjunction with the values you have set in the `bibus » loadBalancingModeEnum` enumeration set to properly contain your dispatchers in groups that use the same load-balancing algorithm. Failure to meet this requirement will result in unpredictable forwarding of requests by dispatchers.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- can be acquired from a containing object

ssAuditLevel

Specifies the auditing level for the system service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

contact

Describes a contact.

A contact can be either a person or a distribution list in an email system that is external to IBM Cognos Analytics. A contact can receive reports created with IBM Cognos Analytics, but cannot log on to IBM Cognos Analytics as an authenticated user.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- [bibus » adminFolder » contact](#)
- [bibus » aliasRoot » contact](#)
- [bibus » annotation » contact](#)
- [bibus » annotationFolder » contact](#)
- [bibus » archiveLocation » contact](#)
- [bibus » baseAgentDefinition » contact](#)
- [bibus » baseDataIntegrationTask » contact](#)
- [bibus » baseDataMovementTask » contact](#)
- [bibus » basePowerPlay8Report » contact](#)

- [bibus » basePowerPlayClass » contact](#)
- [bibus » baseReport » contact](#)
- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » distributionList » members](#)

- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)

Container Information

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Properties

This class has the following properties.

businessPhone

Specifies the business telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `telephonenumber` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

email

Specifies the email address. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `mail` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 128 characters

faxPhone

Specifies the fax number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `facsimiletelephonenumber` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

format

Specifies the rendering format.

When used in the [bibus » account](#) class or the [bibus » contact](#) class, this property specifies the preferred output format of reports for the account or contact.

When used in the [bibus » documentContent](#) class or [bibus » output](#) class this property specifies the format of the data contained in the object

This property

- is of type `bibus » outputFormatEnum`
is encoded as type `tns:nmtokenProp`
- must contain no more than 10 characters
- is searchable

givenName

Specifies the given name. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `givenname` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

homePhone

Specifies the home telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `homephone` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- must contain no more than 20 characters

horizontalElementsRenderingLimit

Specifies the number of elements to be rendered horizontally on a canvas of a predefined size, such as an HTML page. Provides the default value for the runOption `horizontalElements`.

This property

- is of type `int`
is encoded as type `tns:intProp`

locale

Specifies the locale for this object. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

When used by `bibus » contact`, this property determines the language and data format of the returned content.

Use the appropriate language so that users understand object names and search paths. Use the appropriate region so that date, time, and currency values are presented in the proper localized format.

This property

- is of type `language`
is encoded as type `tns:languageProp`
- must contain no more than 64 characters
- is searchable

mobilePhone

Specifies the mobile telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `mobile` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

pagerPhone

Specifies the pager telephone number. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `pager` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters

postalAddress

Specifies the postal address. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `postaladdress` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 4095 characters

surname

Specifies the surname. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the `sn` value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 32 characters

timeZoneID

Specifies the time zone for the object. The specification for the ID string follows the International Components for Unicode (ICU) time zone format.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 20 characters
- is searchable

userName

Specifies the user name. If the authentication provider supports a corresponding property, this property reflects the value recorded in the account in the namespace. For example, this property specifies the uid value stored in an LDAP namespace. It may be possible to override this value.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 64 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

verticalElementsRenderingLimit

Specifies the number of elements to be rendered vertically on a canvas of a predefined size, such as an HTML page. Provides the default value for the runOption `verticalElements`.

This property

- is of type `int`
 - is encoded as type `tns:intProp`

content

Contains all packages and top-level folders.

This object is created when the content store is initialized. This object cannot be deleted by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [account](#) » [portalPages](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [folder](#)
- [bibus](#) » [launchable](#)
- [bibus](#) » [package](#)

Contained by instances of the following classes

- [bibus](#) » [root](#)

Properties

This class has the following properties.

effectiveUserCapabilities

Contains the current user's capabilities for the object. This value is determined by intersecting the set of user capabilities granted by [bibus » content » userCapabilityPolicies](#) property with the set of user capabilities granted to the user globally, as returned by [bibus » session » userCapabilities](#) property.

In order for a capability to be effective for the object, it must be granted to the user both globally and for the object.

This property

- is an array of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

New in Version 10.1.0 – “Object Capabilities Properties” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

items

Contains the folders and packages within the content object.

New in Version 10.1.0 – “Multi-Instance IBM Cognos Connection” on page 1863

This property was extended to allow instances of the [bibus » launchable](#) class to be contained by instances of this property.

powerPlay8Configuration

Specifies the PowerPlay configuration data for this object.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

repositoryRules

This property

- is an array of type [bibus » repositoryRule](#)
is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The `bibus » account`, `bibus » namespace`, and `bibus » namespaceFolder` classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

userCapabilities

Contains the current user's capabilities for the object, as defined by the `userCapabilityPolicies` property.

This property

- is an array of type `bibus » userCapabilityEnum`
 - is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

New in Version 10.1.0 – “Object Capabilities Properties” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

userCapabilityPolicies

Contains the user capability policy for the object.

This property

- is an array of type `bibus » userCapabilityPolicy`
 - is encoded as type `tns:userCapabilityPolicyArrayProp`
- can be acquired from a containing object

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

contentManagerCacheService

Defines run-time configuration parameters for the content manager cache service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Content Manager Cache Service” on page 1862](#)

This class was added.

Related information:

- [“Content Manager cache service” on page 7](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

cmcsAuditLevel

Specifies the auditing level for the content manager cache service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmcsHeapLimit

Specifies the upper limit of the cache size for the Content Manager cache service. This value is represented as a percentage of the JVM heap size.

The default is 10%.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

contentManagerQueryOption

Defines the abstract base class for all content manager query option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the `bibus » option` class

Derived Classes

- `bibus » contentManagerQueryOptionBoolean`
- `bibus » contentManagerQueryOptionInt`
- `bibus » contentManagerQueryOptionPropEnumArray`
- `bibus » contentManagerQueryOptionRefPropArray`
- `bibus » contentManagerQueryOptionString`

Properties

This class has the following properties.

name

Identifies the content manager query option.

This property

- is of type [bibus » contentManagerQueryOptionEnum](#)
is encoded as type `tns:contentManagerQueryOptionEnum`

contentManagerQueryOptionBoolean

Defines boolean values for the content manager query options.

This class

- inherits properties from the [bibus » contentManagerQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the content manager query option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

contentManagerQueryOptionInt

Defines int values for the content manager query options.

This class

- inherits properties from the [bibus » contentManagerQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the content manager query option.

This property

- is of type `int`
is encoded as type `xs:int`

contentManagerQueryOptionPropEnumArray

Defines [bibus » propEnum](#) values for the content manager query options.

This class

- inherits properties from the [bibus » contentManagerQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the content manager query option.

This property

- is an array of type [bibus](#) » [propEnum](#)
is encoded as type `tns:propEnumArray`

contentManagerQueryOptionRefPropArray

Defines [bibus](#) » [refProp](#) values for the content manager query options.

This class

- inherits properties from the [bibus](#) » [contentManagerQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the content manager query option.

This property

- is an array of type [bibus](#) » [refProp](#)
is encoded as type `tns:refPropArray`

contentManagerQueryOptionString

Defines string values for the content manager query options.

This class

- inherits properties from the [bibus](#) » [contentManagerQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the content manager query option.

This property

- is of type `string`
is encoded as type `xs:string`

contentManagerService

Defines run-time configuration parameters for the [contentManagerService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Related information:

- [Chapter 5, “Managing content,” on page 55](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

cmsAuditLevel

Specifies the auditing level for the content manager service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

cmsConnections

Specifies the maximum number of connections that a process of the content manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

cmsPeakConnections

Specifies the number of connections that a content manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

startAsActive

Specifies that the Content Manager service represented by this instance is preferred as the active Content Manager service when IBM Cognos Analytics starts.

On startup, Content Manager services negotiate to determine which service should become active and which services should run in standby mode. The maximum negotiation time is determined by the value of `CMSYNC_NEGOTIATIONTIME` set in the `bibus » contentManagerService » advancedSettings` property. Negotiations are necessary only when no Content Manager service is active. If a service starts when a Content Manager service is active, the starting service is automatically placed on standby.

The following conditions are used to determine which service becomes active:

- If the service has the value of `startAsActive` set to `true`, it is selected as the active Content Manager service and negotiations end.
- When the negotiation time expires, the service controlled by the `bibus » dispatcher` with the highest `capacity` value is selected as the active Content Manager service.

Any active Content Manager service will remain active until

- the active service fails
- the `content » activate\(searchPath\)` method is called to change the currently active service

If a Content Manager service fails, the services running in standby mode enter into negotiation and elect a new service, based on the conditions described. No negotiation occurs as a result of a call to the `content » activate(searchPath)` method.

No more than one instance of this class can have the value `true` assigned to this property.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

contentStoreUtilizationConfiguration

Defines the content store utilization properties.

References

Used by the following properties:

- [bibus » contentTaskOptionCSUtilizationConfiguration » value](#)

Properties

This class has the following properties.

filePerTenant

Specifies whether to include information in a single or multiple files.

When set to `false`, information for all tenants is stored in a single file. When set to `true`, information for each individual tenant is stored in a separate file.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

granularity

Summarizes information for each tenant.

The following values can be used:

classesPerTenant

Summarizes information for each tenant, indicating how many instances of each object class are associated with the tenant in the content store, and an approximation of how much space is used by those objects. If report output is stored in an external repository, the amount of space used in the external source is indicated as well.

The information is presented in comma-delimited format, as in the following example:

```
tenantID,objectClass,instance count,size (bytes),size in EOS (bytes)
,root,1,100,0
,folder,22,273,0
landlord,folder,3,3,0
tenant3,folder,1,1,0
```

perObject

For each object associated with this tenant, the class of object, the value of the [name](#), [storeID](#), and [parent](#), and the size of the object in the content store as well as in an external repository if one is used.

The information is presented in comma-delimited format, as in the following example:

```
tenantID,objectName,objectClass,size (bytes),size in EOS
(bytes),storeID,parentStoreID
,/,root,100,0,iF9B7EE5777F34079815320C7227D1527,iF9B7EE5777F34079815320C7227D15
27
,Administration,adminFolder,552,0,i65E28445E459442D852DF6DE391DAC58
,iF9B7EE5777F34079815320C7227D1527
,Public Folders,content,8089,0,iE1EF5B7BEE02488A8CAF565E87004547
,iF9B7EE5777F34079815320C7227D1527
,Capability,capability,97,0,i6FD1543C4C0F4562A1397E61FF3F630B
,iF9B7EE5777F34079815320C7227D1527
```

This property

- is of type `string`
 - is encoded as type `xs:string`
- has a default value of `classesPerTenant`

tenantFilter

A list of tenant IDs that you want to collect information for. Only objects belonging to the tenants specified are used.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArray`

contentTask

Defines a Content Manager task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#)
- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)

- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `contentTask` class.

<i>Table 154. Services and methods for the contentTask class.</i>			
Action	Mode	Service	Method
Run	All	contentManagerService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus](#) » [history](#)
- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [adminFolder](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

contentTaskOption

Defines the abstract base class for all content task option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » contentTaskOptionAuditLevelEnum](#)
- [bibus » contentTaskOptionBoolean](#)
- [bibus » contentTaskOptionClassEnumArray](#)
- [bibus » contentTaskOptionCSUtilizationConfiguration](#)
- [bibus » contentTaskOptionRetentionRuleArray](#)
- [bibus » contentTaskOptionSearchPathSingleObjectArray](#)

Properties

This class has the following properties.

name

Identifies the content task option.

This property

- is of type [bibus » contentTaskOptionEnum](#)
 - is encoded as type `tns:contentTaskOptionEnum`

contentTaskOptionAuditLevelEnum

Defines [bibus » auditLevelEnum](#) values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

What's new

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the content task option.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnum`

contentTaskOptionBoolean

Defines boolean values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the content task option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

contentTaskOptionClassEnumArray

Defines [bibus » classEnum](#) values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the content task option.

This property

- is an array of type [bibus » classEnum](#)
is encoded as type `tns:classEnumArray`

contentTaskOptionCSUtilizationConfiguration

Defines [bibus » contentStoreUtilizationConfiguration](#) values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the content task option.

This property

- is of type [bibus » contentStoreUtilizationConfiguration](#)
is encoded as type `tns:contentStoreUtilizationConfiguration`

contentTaskOptionRetentionRuleArray

Defines [bibus » retentionRule](#) values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

What's new

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the content task option.

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArray`

contentTaskOptionSearchPathSingleObjectArray

Defines [bibus » searchPathSingleObject](#) values for the content task options.

This class

- inherits properties from the [bibus » contentTaskOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the content task option.

This property

- is an array of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

contextOption

Defines the abstract base class for all context option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » contextOptionFormat](#)
- [bibus » contextOptionStringArray](#)
- [bibus » contextOptionType](#)

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This class was added.

Properties

This class has the following properties.

name

Identifies the context option.

This property

- is of type [bibus » contextOptionEnum](#)
is encoded as type `tns:contextOptionEnum`

contextOptionFormat

Defines [bibus » contextFormatEnum](#) values for the context options.

This class

- inherits properties from the [bibus » contextOption](#) class

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the context option.

This property

- is of type [bibus » contextFormatEnum](#)

is encoded as type `tns:contextFormatEnum`

contextOptionStringArray

Defines string values for the context options.

This class

- inherits properties from the [bibus » contextOption](#) class

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the context option.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

contextOptionType

Defines [bibus » contextTypeEnum](#) values for the context options.

This class

- inherits properties from the [bibus » contextOption](#) class

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the context option.

This property

- is of type [bibus » contextTypeEnum](#)
is encoded as type `tns:contextTypeEnum`

conversationContext

Defines the class for the state information of each asynchronous operation.

Information contained by instances of this class is used by the dispatcher to properly route requests in the same asynchronous operation to the appropriate Simple Object Access Protocol (SOAP) node.

References

Used by the following properties:

- [bibus](#) » [tracking](#) » [conversationContext](#)

Properties

This class has the following properties.

affinityStrength

Specifies the amount of time that can be saved by routing a high affinity request to the same process.

Specifies the amount of time, in milliseconds, that can be saved by routing a request back to the process that handled the original request in the sequence. Affinity describes how tightly the request is associated with a process.

The request for the first page of the report output can take much longer than requests for additional output pages. The process handles the subsequent requests more efficiently because it caches information from the first request. Subsequent requests have a high affinity to the process because of the efficiency gained by sending these requests to the same process. If a subsequent request is sent to a different process, that process must perform all the activities performed by the process that received the first request, which takes more time.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `int`
is encoded as type `xs:int`

id

Identifies the asynchronous operation.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

nodeID

Identifies the dispatcher that handled the initial request of the asynchronous operation.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

processID

Identifies the process that handled the initial request of the asynchronous operation.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `int`
is encoded as type `xs:int`

status

Specifies the status of the current asynchronous conversation. The valid values for this property are specified in the [bibus » `asynchStatusEnum`](#) enumeration set.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

cookieVar

Defines the class used to store information about a browser cookie for the request.

References

Used by the following properties:

- [bibus » `hdrSession` » `cookieVars`](#)

Properties

This class has the following properties.

name

Identifies the browser cookie.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the value assigned to the browser cookie.

This property

- is of type `string`
is encoded as type `xs:string`

copyAccountOption

Defines the abstract base class for all copy account option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » `option`](#) class

Derived Classes

- [bibus » `copyAccountOptionBoolean`](#)

Properties

This class has the following properties.

name

Identifies the copy account option.

This property

- is of type [bibus](#) » [copyAccountOptionEnum](#)
is encoded as type `tns:copyAccountOptionEnum`

copyAccountOptionBoolean

Defines boolean values for the copy account options.

This class

- inherits properties from the [bibus](#) » [copyAccountOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the copy account option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

copyOptions

Defines the options that can be specified for copy operations.

References

Used by the following method parameters:

- [content](#) » [copy\(objects, targetPath, options\)](#) » [options](#)
- [content](#) » [copyRename\(objects, targetPath, newNames, options\)](#) » [options](#)

Properties

This class has the following properties.

faultIfObjectReferenced

Specifies how references to objects being deleted or replaced are processed by Content Manager.

If this property is set to `false`, any reference to the objects being deleted are automatically deleted. If this property is set to `true`, Content Manager generates a fault if any object in the content store continues to reference any of the deleted objects.

During recursive delete, a fault is not generated if one deleted object refers to another object that is also deleted.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

- has a default value of `false`

keepTenancy

Controls the tenancy of the copied objects.

If this property is set to `false`, the target object `tenantID` is inherited from its parent object (virtual inheritance). If this property is set to `true`, the `tenantID` of the source object is used for the target object.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.2.2 – “Support for delegated tenant administration” on page 1826

This property was added.

recursive

Specifies whether contained objects are copied.

If set to `true`, all descendant objects for which the current security context has read access are also copied.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`

targetTenantID

This property

- is of type `string`
 - is encoded as type `xs:string`

updateAction

Specifies what happens when an object already exists at the new object's location in the content store. Either the properties of the object are updated, or the object and all its descendants are deleted and a new object is created that has the properties specified in the request.

This property

- is of type `bibus » updateActionEnum`
 - is encoded as type `tns:updateActionEnum`

credential

Provides the information to allow authentication by a security provider.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- `bibus » schedule » credential`

Container Information

Contained by instances of the following classes

- [bibus](#) » [account](#)

Properties

This class has the following properties.

credentials

Specifies credentials for IBM Cognos Analytics and other authentication providers.

Each `credential` element includes information required by an authentication provider or a data source. Typically, this information is in the form of a username and password, but additional information may be required.

For an authentication provider, IBM Cognos Analytics requires a `namespace` element to determine which provider to use. The `namespace` element contains the namespace ID as specified during configuration of the namespace. For example, IBM Cognos Analytics configured with Windows native security (NTLM) requires a `namespace` element, a `username` element, and a `password` element, as shown in the following plaintext example:

Authentication providers may require additional information. New elements can be added within the `credential` element as required for a particular provider.

For a data source, IBM Cognos Analytics does not require a `namespace` element. The data source connection string identifies the data source as the authentication provider.

The `SAPLogonTicket` element is used to sign on to a SAP data source. This element contains a string. For more information about the contents of the `SAPLogonTicket` element, see your SAP database administrator.

When sending multiple credentials, the entire set is nested within a `credentials` element, as follows:

When a `credential` is stored for use with a schedule, or as a trusted credential, the information is encrypted as shown:

The entire XML fragment is encrypted, encoded in base64 format, and stored in an `encryptedValue` element to secure the credential.

To store a credential, use the `generateTC` parameter of the [bibus](#) » [CAM](#) » [action](#) property. Grant users access to a trusted credential by:

- adding the user, group, or role to the [bibus](#) » [baseClass](#) » [policies](#) property for the credential
- granting execute permissions to the user, group, or role added

Authentication providers may require additional information. New elements can be added within the `credential` element as required for a particular provider.

For a data source, IBM Cognos Analytics does not require a `namespace` element. The data source connection string identifies the data source as the authentication provider.

The `SAPLogonTicket` element is used to sign on to a SAP data source. This element contains a string. For more information about the contents of the `SAPLogonTicket` element, see your SAP database administrator.

When sending multiple credentials, the entire set is nested within a `credentials` element, as shown.

```
<credential>
  <namespace>...</namespace>
  <username>...</username>
  <password>...</password>
```

```
</credential>
```

When a `credential` is stored for use with a schedule, or as a trusted credential, the information is encrypted as shown.

```
<credentials>
  <credential>...</credential>
  ...
  <credential>...</credential>
</credentials>
```

The entire XML fragment is encrypted, encoded in base64 format, and stored in an `encryptedValue` element to secure the credential.

To store a credential, use the `generateTC` parameter of the [bibus » CAM » action](#) property. Grant users access to a trusted credential by:

1. adding the user, group, or role to the [bibus » baseClass » policies](#) property for the credential
2. granting execute permissions to the user, group, or role added

```
<credential>
  <encryptedValue>
    <!-- base64 encoded data -->
    AbcdEf1jKdkhuW8...
  </encryptedValue>
</credential>
```

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- is write-only

Related information:

IBM Cognos IBM Cognos Analytics Administration and Security Guide

IBM Cognos IBM Cognos Analytics Installation and Configuration Guide

currency

Defines the currency code and provides a description of the currency.

References

Used by the following properties:

- [bibus » configurationData » supportedCurrencies](#)
- [bibus » currencyArrayProp » value](#)
- [bibus » currencyProp » value](#)

Properties

This class has the following properties.

currency

Specifies the currency code in accordance with the ISO 4217 standard.

This property

- is of type `token`
is encoded as type `xs:string`
- is read-only
- must contain exactly 3 characters
- must contain exactly 3 uppercase ASCII letters

description

Specifies a description of the currency in the user's preferred language, if a locale-specific description is available.

The system determines if there is a locale-specific description by searching the following, in sequence:

- the currency description in the language specified by the `bibus » account » contentLocale` property
- the currency description in the language specified by the `bibus » account » productLocale` property
- the description associated with the ISO 4217 currency code

This property

- is of type `string`
is encoded as type `xs:string`
- is read-only

currencyArrayProp

Defines the array property class for the `bibus » currency` class.

This class

- inherits properties from the `bibus » baseProp` class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type `bibus » currency`
is encoded as type `tns:currencyArray`

currencyProp

Defines the simple property class for the `bibus » currency` class.

This class

- inherits properties from the `bibus » baseProp` class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » currency](#)
is encoded as type `tns:currency`

dashboard

Contains information used to define an IBM Cognos Workspace workspace.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » personalization » base](#)
- [bibus » shortcut » target](#)

Container Information

Contains instances of the following classes

- [bibus » baseAgentDefinition](#)
- [bibus » baseDataIntegrationTask](#)
- [bibus » baseDataMovementTask](#)
- [bibus » basePowerPlay8Report](#)
- [bibus » basePowerPlayClass](#)
- [bibus » baseReport](#)
- [bibus » dashboard](#)
- [bibus » document](#)
- [bibus » folder](#)
- [bibus » jobDefinition](#)
- [bibus » package](#)
- [bibus » pagelet](#)
- [bibus » personalization](#)
- [bibus » planningTask](#)
- [bibus » shortcut](#)
- [bibus » URL](#)

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)
- [bibus » pageletFolder](#)

What's new

New in Version 8.4 — [“Dashboards” on page 1894](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)

has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

context

Specifies the metadata for this dashboard.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [xmlEncodedXML](#)

is encoded as type `tns:xmlEncodedXMLProp`

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus](#) » [addOptions](#) » [dataEncoding](#) property and the [bibus](#) » [queryOptions](#) » [dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`

is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type `bibus » deploymentReference`
 - is encoded as type `tns:deploymentReferenceArrayProp`

items

Contains the child objects for this object.

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method. The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

dataAdvisorService

Defines run-time configuration parameters for the data advisor service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

What's new

New in Version 10.1.0 – [“Updated Support for IBM Cognos Express” on page 1871](#)

This class was added.

Related information:

- [“Data advisor service” on page 7](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
is encoded as type `tns:anyTypeProp`

- can be acquired from a containing object

dasAuditLevel

Specifies the auditing level for the data advisor service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Updated Support for IBM Cognos Express” on page 1871

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

dataIntegrationService

Defines run-time configuration parameters for the [dataIntegrationService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Related information:

- [“Data integration service” on page 7](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

disAuditLevel

Specifies the auditing level for the data integration service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

disConnections

Specifies the maximum number of connections that a process of the data integration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

disPeakConnections

Specifies the number of connections that a data integration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

dataIntegrationServiceSpecification

Defines the type for the data integration service specifications.

This class

- inherits properties from the `bibus » asyncSpecification` class

References

Used by the following properties:

- `bibus » baseDataIntegrationTask » specification`
- `bibus » dataIntegrationServiceSpecificationArrayProp » value`
- `bibus » dataIntegrationServiceSpecificationProp » value`

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `dataIntegrationServiceSpecification` class.

<i>Table 156. Services and methods for the dataIntegrationServiceSpecification class.</i>			
Action	Mode	Service	Method
Run	All	<code>dataIntegrationService</code>	<code>async » runSpecification(specification, parameterValues, options)</code>

dataIntegrationServiceSpecificationArrayProp

Defines the array property class for the [bibus » dataIntegrationServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » dataIntegrationServiceSpecification](#)
is encoded as type `tns:dataIntegrationServiceSpecificationArray`

dataIntegrationServiceSpecificationProp

Defines the simple property class for the [bibus » dataIntegrationServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » baseDataIntegrationTask » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » dataIntegrationServiceSpecification](#)
is encoded as type `tns:dataIntegrationServiceSpecification`

dataIntegrationTaskOption

Defines the abstract base class for all data integration task option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » dataIntegrationTaskOptionBoolean](#)

Properties

This class has the following properties.

name

Identifies the data integration task option.

This property

- is of type [bibus » dataIntegrationTaskOptionEnum](#)
is encoded as type `tns:dataIntegrationTaskOptionEnum`

dataIntegrationTaskOptionBoolean

Defines boolean values for the data integration task options.

This class

- inherits properties from the [bibus » dataIntegrationTaskOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the data integration task option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

dataMovementService

Defines run-time configuration parameters for the [dataMovementService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Related information:

- [“Data movement service” on page 7](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

dmsAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsAuditLevel

Specifies the auditing level for the data movement service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

dmsConnections – deprecated

Specifies the maximum number of connections that a process of the data movement service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- is deprecated and will be removed in a future version of the product
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property is deprecated. Use the following properties instead:

- [bibus](#) » [dataMovementService](#) » [dmsAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsPeakNonAffineConnections](#) property

dmsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsPeakAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

New in Version 8.3 — [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsPeakMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1

- can be acquired from a containing object

dmsPeakNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsQueueLimit

Specifies the number of seconds that a request for the data movement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.3 — “Data Movement Service Configuration Parameters” on page 1917

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

dataMovementServiceSpecification

Defines the class for data movement service specifications.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

References

Used by the following properties:

- [bibus » dataMovementTask » specification](#)
- [bibus » dataMovementServiceSpecificationProp » value](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the dataMovementServiceSpecification class.

Action	Mode	Service	Method
Run	All	dataMovementService	asynch » runSpecification(specification, parameterValues, options)

dataMovementServiceSpecificationProp

Defines the simple property class for the [bibus » dataMovementServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » dataMovementTask » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » dataMovementServiceSpecification](#)
is encoded as type `tns:dataMovementServiceSpecification`

dataMovementTask

Defines a data movement task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseDataMovementTask](#) class

References

Used by the following properties:

- [bibus » dataMovementTaskAlias » base](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the dataMovementTask class.

Action	Mode	Service	Method
Run	All	dataMovementService	asynch » run(objectPath, parameterValues, options)

What's new

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This class is now derived from the [bibus » baseDataMovementTask](#) class.

Properties

This class has the following properties.

ownerPassport – obsolete

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property is obsolete and was removed. Use the [bibus » baseDataMovementTask » ownerPassport](#) property instead.

runAsOwner – obsolete

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property is obsolete and was removed. Use the [bibus » baseDataMovementTask » runAsOwner](#) property instead.

runWithOwnerCapabilities

Reserved.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

New in Version 10.1.0 – “Run with Owner Capabilities” on page 1871

This property was added.

specification

Specifies the specification for the task.

This property

- is of type [bibus » dataMovementServiceSpecification](#)
is encoded as type `tns:dataMovementServiceSpecificationProp`

dataMovementTaskAlias

Defines the customization of a data movement task.

All properties other than the referenced specification can be changed in the data movement task alias.

Important: The [bibus » baseDataMovementTask » runAsOwner](#) property is ignored for this class. It is valid only when used with an instance of [bibus » dataMovementTask](#) class.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseDataMovementTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `dataMovementTaskAlias` class.

Action	Mode	Service	Method
Run	All	dataMovementService	<code>asynch » run(objectPath, parameterValues, options)</code>

What's new

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This class was added.

Properties

This class has the following properties.

base

Refers to the [bibus » dataMovementTask](#) on which this data movement task alias is based.

If the [dataMovementTask](#) that the data movement task alias is based on is moved, the id-based search path that references the object is preserved.

If the [dataMovementTask](#) that the data movement task alias is based on is deleted, the alias can still exist but should be updated to reference a valid search path.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » dataMovementTask](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

dataSet

Defines the object that contains the information for the Line of Business (LOB) data.

This class

- inherits properties from the [bibus » uiClass](#) class

What's new

New in Version 10.2.2 – [My data sets](#)

This class was added.

Container Information

Contained by instances of the following classes

- [bibus » dataSetFolder](#)

Properties

This class has the following properties.

dataModificationTime

Specifies the last time the imported Line of Business (LOB) data was refreshed, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is searchable

dataSetSize

Specifies the size of the imported Line of Business (LOB) data in number of KBs.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `nonNegativeIntegerProp`
- is searchable

packageBase

Refers to the package which this `dataSet` was last published to. A `dataSet` can exist even when the package it is related to is deleted.

This property

- is an array of type [bibus » baseClass](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an ID-based search path
- must contain at most 1 item

specification

Reserved.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXMLMIMEProp`

dataSetFolder

Contains the dataSet objects created by a user.

This class

- inherits properties from the [bibus » uiClass](#) class

What's new

New in Version 10.2.2 – [My data sets](#)

This class was added.

Container Information

Contains instances of the following classes

- [bibus » dataSet](#)

Contained by instances of the following classes

- [bibus » account](#)

Properties

This class has the following properties.

items

Contains the child objects for this object.

This property

- has no specified type information
- has items that must be of class [bibus » dataSet](#) or [bibus » dataSetFolder](#)

dataSource

Identifies an originator of data used by IBM Cognos Analytics for reports and queries.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » dataSourceConnection](#)

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » package](#)

Related information:

- IBM Cognos *Administration and Security Guide*

Properties

This class has the following properties.

capabilities

Contains the set of capabilities that can be used with this data source.

This property

- is an array of type [bibus » dataSourceCapabilityEnum](#)
is encoded as type `tns:anyURIArrayProp`
- is searchable

New in Version 10.1.0 – “Personal Packages” on page 1863

This property was added.

closeConnectionCommands

Specifies the commands to be run when a connection is closed.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

closeSessionCommands

Specifies the commands to be run when a session is closed.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

connections

Specifies the connections for this data source.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

openConnectionCommands

Specifies the commands to be run when a connection is opened.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

openSessionCommands

Specifies the commands to be run when a session is opened.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

dataSourceCommandBlock

Defines the type for data source command blocks.

A data source command block is an XML document that is used to specify commands to be run by the database when the query engine performs specific actions, such as opening a connection or closing a user session.

This class

- inherits properties from the [bibus » xmlEncodedXML](#) class

References

Used by the following properties:

- [bibus » dataSource » closeConnectionCommands](#)
- [bibus » dataSourceConnection » closeConnectionCommands](#)
- [bibus » dataSource » closeSessionCommands](#)
- [bibus » dataSourceConnection » closeSessionCommands](#)
- [bibus » dataSource » openConnectionCommands](#)
- [bibus » dataSourceConnection » openConnectionCommands](#)
- [bibus » dataSource » openSessionCommands](#)
- [bibus » dataSourceConnection » openSessionCommands](#)
- [bibus » dataSourceCommandBlockProp » value](#)

Related information:

- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

dataSourceCommandBlockProp

Defines the simple property class for the [bibus » dataSourceCommandBlock](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » dataSource » closeConnectionCommands](#)
- [bibus » dataSourceConnection » closeConnectionCommands](#)

- [bibus » dataSource » closeSessionCommands](#)
- [bibus » dataSourceConnection » closeSessionCommands](#)
- [bibus » dataSource » openConnectionCommands](#)
- [bibus » dataSourceConnection » openConnectionCommands](#)
- [bibus » dataSource » openSessionCommands](#)
- [bibus » dataSourceConnection » openSessionCommands](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlock`

dataSourceConnection

Defines a connection to the physical database used as the data source for IBM Cognos Analytics reports and queries.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » baseDataIntegrationTask » connections](#)

Container Information

Contains instances of the following classes

- [bibus » dataSourceNameBinding](#)
- [bibus » dataSourceSignon](#)

Contained by instances of the following classes

- [bibus » dataSource](#)

Related information:

- IBM Cognos *Administration and Security Guide*

Properties

This class has the following properties.

bindings

Contains the list of name bindings for this data source connection.

closeConnectionCommands

Specifies the commands to be run when a connection is closed.

This property

- is of type [bibus » dataSourceCommandBlock](#)
 - is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

closeSessionCommands

Specifies the commands to be run when a session is closed.

This property

- is of type [bibus » dataSourceCommandBlock](#)
 - is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

connectionString

Specifies the connection string for the database. Use the format appropriate to your RDBMS implementation. For example, a typical database connection string for Microsoft® SQL Server is: `;LOCAL;OL;DBInfo_Type=MS;Provider=SQLOLEDB;User ID="";Password="";DataSource=localhost;Provider_String=Initial Catalog=CR1_G0sales`

For more information about connection strings used in data source connections, see the *IBM Cognos Analytics Administration and Security Guide*.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

credentialNamespaces

Refers to a [bibus » namespace](#) object that may provide the credentials required for the [bibus » dataSourceConnection](#).

The following steps are used to locate credentials for a [bibus » dataSourceConnection](#):

1. Content Manager searches for a [bibus » dataSourceSignon](#) object contained in the [bibus » dataSourceConnection](#). The [bibus » dataSourceSignon » consumers](#) property is used to determine the available [bibus » dataSourceSignon](#) objects. If one [bibus » dataSourceSignon](#) object is available, the credential from the object is used. If multiple [bibus » dataSourceSignon](#) objects are available, the user may be prompted to select one. The credential from the selected object is used.
2. If no [bibus » dataSourceSignon](#) object is available and this property references a [bibus » namespace](#), then Content Manager requests the credential from IBM Cognos Access Manager.
3. If a credential cannot be determined, the user may be prompted for credentials.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » namespace](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

isolationLevel

Specifies the minimum isolation level used with this connection. For multiple connections, the most strict level applies.

This property

- is of type [bibus » isolationLevelEnum](#)
is encoded as type `tns:nmtokenProp`

openConnectionCommands

Specifies the commands to be run when a connection is opened.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

openSessionCommands

Specifies the commands to be run when a session is opened.

This property

- is of type [bibus » dataSourceCommandBlock](#)
is encoded as type `tns:dataSourceCommandBlockProp`
- can be acquired from a containing object

signons

Contains the set of database signons for this data source.

dataSourceCredential

Stores the credentials required to sign on to the data source specified by the [dataSourceName](#) and [dataSourceConnectionName](#) properties.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » account](#)

What's new

New in Version 10.1.0 – “[Personal Data Source Credentials](#)” on page 1851

This class was added.

Properties

This class has the following properties.

credentials

Specifies the credentials for the data source.

Each `credential` element includes information required by the data source. Typically, this information is in the form of a username and password, but additional information may be required.

For a data source, IBM Cognos Analytics does not require a namespace element. The data source connection string identifies the data source as the authentication provider.

When a `dataSourceCredential` is stored, the information is encrypted as shown in the following example.

```
<credential>
  <encryptedValue>
    <!-- base64 encoded data -->
    AbcdEf1jKdkhuW8...
  </encryptedValue>
</credential>
```

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- is write-only

Related information:

IBM Cognos Analytics Administration and Security Guide

IBM Cognos Analytics Installation and Configuration Guide

dataSourceConnectionName

Specifies the name of the [bibus » dataSourceConnection](#) that can be accessed with the specified [credentials](#).

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is searchable

dataSourceName

Specifies the name of the [dataSource](#) that can be accessed with the specified [credentials](#).

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is searchable

dataSourceNameBinding

Defines a name binding rule for the connection to the database.

Use the object name to match the rule to the original name.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [dataSourceNameBinding](#)

Contained by instances of the following classes

- [bibus](#) » [dataSourceConnection](#)
- [bibus](#) » [dataSourceNameBinding](#)

Properties

This class has the following properties.

bindings

Contains the list of bindings that can be applied if the name binding matches.

qualifier

Specifies that a rule applies to a catalog, schema, table, view, or stored procedure.

This property

- is of type [bibus](#) » [bindingQualifierEnum](#)
is encoded as type `tns:nmtokenProp`

replacement

Specifies a string that is substituted if the name binding matches.

This property

- is of type `NCName`
is encoded as type `tns:ncnameProp`
- can contain at least 255 characters

dataSourceSignon

Defines signon credentials for a database.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [baseDataIntegrationTask](#) » [connections](#)

Container Information

Contained by instances of the following classes

- [bibus](#) » [dataSourceConnection](#)

Properties

This class has the following properties.

consumers

Refers to the accounts, groups, namespaces, and roles that can use this signon.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » group](#), [bibus » namespace](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

Related information:

IBM Cognos Administration and Security Guide

credentials

Specifies the credentials required for the database, in general an account name and password.

This property contains the same XML information as the [bibus » credential » credentials](#) property.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

dateTimeProp

Defines the simple property class for the `dateTime`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » history » actualCompletionTime](#)
- [bibus » history » actualExecutionTime](#)
- [bibus » reportVersion » asOfTime](#)
- [bibus » batchReportService » brsDataSourceChange](#)
- [bibus » configuration » brsDataSourceChange](#)
- [bibus » configurationFolder » brsDataSourceChange](#)
- [bibus » dispatcher » brsDataSourceChange](#)
- [bibus » baseClass » creationTime](#)
- [bibus » baseHistoryDetail » detailTime](#)
- [bibus » schedule » endDate](#)
- [bibus » agentOutputHotList » expirationTime](#)
- [bibus » documentVersion » expirationTime](#)
- [bibus » history » expirationTime](#)
- [bibus » model » expirationTime](#)
- [bibus » reportCache » expirationTime](#)
- [bibus » reportVersion » expirationTime](#)
- [bibus » shortcut » expirationTime](#)
- [bibus » configuration » lastConfigurationModificationTime](#)
- [bibus » baseClass » modificationTime](#)

- [bibus » history » requestedExecutionTime](#)
- [bibus » configuration » rsDataSourceChange](#)
- [bibus » configurationFolder » rsDataSourceChange](#)
- [bibus » dispatcher » rsDataSourceChange](#)
- [bibus » reportService » rsDataSourceChange](#)
- [bibus » schedule » startDate](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

decimalProp

Defines the simple property class for the `decimal`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » pageDefinition » height](#)
- [bibus » pageDefinition » width](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `decimal`
is encoded as type `xs:decimal`

deleteOptions

Defines the options you can specify for the [content » delete\(objects, options\)](#) method.

References

Used by the following method parameters:

- [content » delete\(objects, options\) » options](#)

Properties

This class has the following properties.

faultIfObjectReferenced

Specifies how references to objects being deleted or replaced are processed by Content Manager.

If this property is set to `false`, any reference to the objects being deleted are automatically deleted. If this property is set to `true`, Content Manager generates a fault if any object in the content store continues to reference any of the deleted objects.

During recursive delete, a fault is not generated if one deleted object refers to another object that is also deleted.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

force

Specifies whether Content Manager can consider the `setPolicy` permissions of a user when deciding if the user can delete a selected object.

If set to `true`, a selected object will be deleted if the current user has either `write` or `setPolicy` permission for the following:

- the selected object
- the parent of the selected object
- every descendant of the selected object

If not set, or if set to `false`, a selected object will be deleted if the current user has `write` permission for the following:

- the selected object
- the parent of the selected object
- every descendant of the selected object

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`

recursive

Specifies whether contained objects are deleted.

If set to `false`, the operation will fail if any object that is specified by the `content » delete(objects, options) » objects` parameter has a child object.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`

deliveryOption

Defines the abstract base class for all delivery option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the `bibus » option` class

Derived Classes

- [bibus » deliveryOptionAddressSMTPArray](#)
- [bibus » deliveryOptionChannel](#)
- [bibus » deliveryOptionMemoPart](#)
- [bibus » deliveryOptionSearchPathMultipleObjectArray](#)
- [bibus » deliveryOptionSearchPathSingleObject](#)
- [bibus » deliveryOptionString](#)

Properties

This class has the following properties.

name

Identifies the delivery option.

This property

- is of type [bibus » deliveryOptionEnum](#)
is encoded as type `tns:deliveryOptionEnum`

deliveryOptionAddressSMTPArray

Defines [bibus » addressSMTP](#) values for the delivery options.

This class

- inherits properties from the [bibus » deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the delivery option.

This property

- is an array of type [bibus » addressSMTP](#)
is encoded as type `tns:addressSMTPArray`

deliveryOptionChannel

Defines [bibus » deliveryChannelEnum](#) values for the delivery options.

This class

- inherits properties from the [bibus » deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the delivery option.

This property

- is of type [bibus » deliveryChannelEnum](#)

is encoded as type `tns:deliveryChannelEnum`

deliveryOptionMemoPart

Defines [bibus » memoPart](#) values for the delivery options.

This class

- inherits properties from the [bibus » deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the delivery option.

This property

- is of type [bibus » memoPart](#)
is encoded as type `tns:memoPart`

deliveryOptionSearchPathMultipleObjectArray

Defines [bibus » searchPathMultipleObject](#) values for the delivery options.

This class

- inherits properties from the [bibus » deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the delivery option.

This property

- is an array of type [bibus » searchPathMultipleObject](#)
is encoded as type `tns:searchPathMultipleObjectArray`

deliveryOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the delivery options.

This class

- inherits properties from the [bibus » deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the delivery option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

deliveryOptionString

Defines string values for the delivery options.

This class

- inherits properties from the [bibus](#) » [deliveryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the delivery option.

This property

- is of type `string`
is encoded as type `xs:string`

deliveryService

Defines run-time configuration parameters for the [deliveryService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

- can be acquired from a containing object

dsAuditLevel

Specifies the auditing level for the delivery service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

dsCompressAttachmentLimit

Specifies the maximum size, in MB, of an uncompressed email attachment. The delivery service will compress an attachment that is larger than the maximum size before sending it.

Use a value of 0 to disable email attachment compression.

Use a `nil` value to compress all email attachments.

Setting a non-`nil`, non-zero value improves performance when sending email messages with large attachments, such as report outputs.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — “[Email Delivery Enhancements](#)” on page 1916

This property was added.

dsConnections

Specifies the maximum number of connections that a process of the delivery service can use concurrently to execute requests.

For more information, see “[Tune Server Performance](#)” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

dsMaximumEMailSize

Specifies the maximum amount of data in MB that can be placed in an email by the delivery service.

Base this value on your mail server configuration. Please contact your mail server administrator to determine an appropriate value. If email sizes exceed that which your mail server is configured to support, consider a different transfer method.

If the uncompressed size of the email exceeds the specified configuration parameter value, the [deliveryService](#) service removes the largest attachments from the message until the total size of the message is lower than the specified configuration parameter value. If an attachment is removed,

an error message is created and added as a plain text attachment to the email indicating that the message size has exceeded the configured limit. If the uncompressed size of the email body (for example, an HTML report) exceeds the configured limit, the error message replaces the content of the body.

Use a value of 0 to specify that the total size of an email is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – [“New Email Configuration Parameters” on page 1868](#)

This property was added.

dsPeakConnections

Specifies the number of connections that a delivery service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – [“System Metrics” on page 1908](#)

This property was added.

deliveryServiceSpecification

Defines the type for delivery service specifications.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `deliveryServiceSpecification` class.

Action	Mode	Service	Method
Run	All	deliveryService	asynch » runSpecification(specification, parameterValues, options)

deploymentDetail

Defines information related to the deployment of an object.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

Container Information

Contains instances of the following classes

- [bibus » deploymentDetail](#)

Contained by instances of the following classes

- [bibus » deploymentDetail](#)

Properties

This class has the following properties.

deployedObject

Refers to the deployed object.

This property

- is an array of type [bibus » baseClass](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using a name- and type-based search path
- must have at most 1 item

deployedObjectAncestorDefaultNames

Specifies the [defaultName](#) of the ancestors of the deployed object.

This property

- is an array of type `token`
 - is encoded as type `tns:tokenArrayProp`

deployedObjectClass

Specifies the [objectClass](#) of the deployed object.

This property

- is of type [bibus](#) » [classEnum](#)
 - is encoded as type `tns:classEnumProp`
- must contain no more than 64 characters
- is searchable

deployedObjectName

Specifies the [defaultName](#) of the deployed object. This is the name that best matches the name expressed in the user's preferred language, as specified by the [bibus](#) » [account](#) » [contentLocale](#) property. The user selects this language in the user preferences.

This property

- is of type `token`
 - is encoded as type `tns:tokenProp`
- can contain at least 255 characters

deployedObjectStatus

Specifies the deployment status of the deployed object.

This property

- is of type [bibus](#) » [deploymentStatusEnum](#)
 - is encoded as type `tns:deploymentStatusEnumProp`
- is searchable

deployedObjectUsage

Specifies the [usage](#) of the deployed object.

This property

- is of type [bibus](#) » [usageEnum](#)
 - is encoded as type `tns:nmtokenProp`
- can contain at least 255 characters
- is searchable

details

Contains deployment details for objects contained within the deployed object.

hasMessage

Specifies that the [message](#) property contains a message.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only
- is searchable

message

Contains the message for the deployed object.

This property

- is an array of type [bibus](#) » [faultDetail](#)
 - is encoded as type `tns:faultDetailArrayProp`

- must have at most 1 item

deploymentImportRule

Defines how objects are imported into the content store from a deployment archive.

References

Used by the following properties:

- [bibus](#) » [deploymentOptionImportRuleArray](#) » [value](#)

What's new

New in Version 8.3 — “[Package Hierarchies](#)” on page 1913

This class was added. It replaces the following classes:

- [deploymentOptionPackageInfo](#)
- [packageDeploymentInfo](#)

Properties

This class has the following properties.

archiveSearchPath

Identifies the object in the deployment archive.

This property

- is of type [bibus](#) » [searchPathSingleObject](#)

is encoded as type `tns:searchPathSingleObject`

disabled

Specifies whether the object is disabled on deployment. Use this property to set the [disabled](#) property on the object.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

name

Specifies the name of the imported content store object when it is added or updated.

This property

- is of type `token`

is encoded as type `tns:multilingualTokenArray`

- is multilingual

parent

Identifies the parent of this content store object when it is added or updated.

This property

- is of type [bibus](#) » [searchPathSingleObject](#)

is encoded as type `tns:searchPathSingleObject`

deploymentObjectInformation

Contains information about objects in the deployment archive.

References

Used by the following properties:

- [bibus](#) » [deploymentOptionObjectInformationArray](#) » [value](#)

What's new

New in Version 8.3 — “Package Hierarchies” on page 1913

This class was added. It replaces the following classes:

- [bibus](#) » [deploymentOptionPackageInfo](#) class
- [bibus](#) » [packageDeploymentInfo](#) class

Properties

This class has the following properties.

ancestors

Contains information about the ancestors of the object.

The value of this property is consistent with the value returned by [bibus](#) » [baseClass](#) » [ancestors](#) property.

This property

- is an array of type [bibus](#) » [ancestorInfo](#)
is encoded as type `tns:ancestorInfoArray`

defaultName

Specifies the object name that best matches the name expressed in the user's preferred language.

The value of this property is consistent with the value returned by [bibus](#) » [baseClass](#) » [defaultName](#) property.

This property

- is of type `token`
is encoded as type `xs:string`

name

Specifies the name of the object.

The value of this property is consistent with the value returned by [bibus](#) » [baseClass](#) » [name](#) property.

This property

- is of type `token`
is encoded as type `tns:multilingualTokenArray`
- is multilingual

objectClass

Specifies the class of the object.

The value of this property is consistent with the value returned by [bibus](#) » [baseClass](#) » [objectClass](#) property.

This property

- is of type [bibus » classEnum](#)
is encoded as type `tns:classEnum`

searchPath

Specifies the search path of the object.

The value of this property is consistent with the value returned by [bibus » baseClass » searchPath](#) property.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

deploymentOption

Defines the abstract base class for all deployment option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » deploymentOptionAnyType](#)
- [bibus » deploymentOptionAuditLevel](#)
- [bibus » deploymentOptionBoolean](#)
- [bibus » deploymentOptionClassEnumArray](#)
- [bibus » deploymentOptionImportRuleArray](#)
- [bibus » deploymentOptionMultilingualString](#)
- [bibus » deploymentOptionObjectInformationArray](#)
- [bibus » deploymentOptionResolution](#)
- [bibus » deploymentOptionSearchPathSingleObjectArray](#)
- [bibus » deploymentOptionString](#)
- [bibus » deploymentOptionStringArray](#)

References

Used by the following properties:

- [bibus » deploymentOptionArrayProp » value](#)
- [bibus » deploymentOptionProp » value](#)

Properties

This class has the following properties.

name

Identifies the deployment option.

This property

- is of type [bibus » deploymentOptionEnum](#)
is encoded as type `tns:deploymentOptionEnum`

deploymentOptionAnyType

Defines anyType values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type anyType
is encoded as type `xs:string`

deploymentOptionArrayProp

Defines the array property class for the [bibus » deploymentOption](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » deploymentOption](#)
is encoded as type `tns:deploymentOptionArray`

deploymentOptionAuditLevel

Defines [bibus » auditLevelEnum](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

What's new

New in Version 8.3 — “[Deployment History](#)” on page 1915

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnum`

deploymentOptionBoolean

Defines boolean values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

deploymentOptionClassEnumArray

Defines [bibus » classEnum](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the deployment option.

This property

- is an array of type [bibus » classEnum](#)
is encoded as type `tns:classEnumArray`

deploymentOptionImportRuleArray

Defines [bibus » deploymentImportRule](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

What's new

New in Version 8.3 — “Package Hierarchies” on page 1913

This class was added. It replaces the following classes:

Properties

This class has the following properties.

value

Specifies the values for the deployment option.

This property

- is an array of type [bibus » deploymentImportRule](#)
is encoded as type `tns:deploymentImportRuleArray`

deploymentOptionMultilingualString

Defines multilingual string values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type `string`
is encoded as type `tns:multilingualStringArray`
- is multilingual

deploymentOptionObjectInformationArray

Defines [bibus » deploymentObjectInformation](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

What's new

New in Version 8.3 – “[Package Hierarchies](#)” on page 1913

This class was added. It replaces the following classes:

- [bibus » deploymentOptionPackageInfo](#) class
- [bibus » packageDeploymentInfo](#) class

Properties

This class has the following properties.

value

Specifies the values for the deployment option.

This property

- is an array of type [bibus » deploymentObjectInformation](#)
is encoded as type `tns:deploymentObjectInformationArray`

deploymentOptionProp

Defines the simple property class for the [bibus » deploymentOption](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » deploymentOption](#)
is encoded as type `tns:deploymentOption`

deploymentOptionResolution

Defines [bibus » conflictResolutionEnum](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type [bibus » conflictResolutionEnum](#)
is encoded as type `tns:conflictResolutionEnum`

deploymentOptionSearchPathSingleObjectArray

Defines [bibus » searchPathSingleObject](#) values for the deployment options.

This class

- inherits properties from the [bibus » deploymentOption](#) class

What's new

New in Version 8.3 — “[Package Hierarchies](#)” on page 1913

This class was added. It replaces the following classes:

- [bibus » deploymentOptionPackageInfo](#) class
- [bibus » packageDeploymentInfo](#) class

Properties

This class has the following properties.

value

Specifies the values for the deployment option.

This property

- is an array of type [bibus » searchPathSingleObject](#)

is encoded as type `tns:searchPathSingleObjectArray`

deploymentOptionString

Defines string values for the deployment options.

This class

- inherits properties from the [bibus](#) » [deploymentOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the deployment option.

This property

- is of type `string`
is encoded as type `xs:string`

deploymentOptionStringArray

Defines string values for the deployment options.

This class

- inherits properties from the [bibus](#) » [deploymentOption](#) class

What's new

New in Version 10.2.0 – [“Support for multi-tenancy” on page 1838](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the deployment option.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

deploymentReference

Defines a type used to maintain references to objects in the content store during deployments.

References

Used by the following properties:

- [bibus](#) » [dashboard](#) » [deploymentReferences](#)
- [bibus](#) » [drillPath](#) » [deploymentReferences](#)
- [bibus](#) » [launchable](#) » [deploymentReferences](#)
- [bibus](#) » [pageletInstance](#) » [deploymentReferences](#)

- [bibus » personalization » deploymentReferences](#)
- [bibus » portlet » deploymentReferences](#)
- [bibus » portletInstance » deploymentReferences](#)
- [bibus » deploymentReferenceArrayProp » value](#)
- [bibus » deploymentReferenceProp » value](#)

What's new

New in Version 8.3 — [“Package Hierarchies” on page 1913](#)

This class was added.

Properties

This class has the following properties.

name

Specifies the name assigned to the reference.

This property

- is of type `token`
- is encoded as type `tns:multilingualTokenArray`
- is multilingual

objects

Contains the object references

This property

- is an array of type [bibus » baseClass](#)
- is encoded as type `tns:baseClassArray`
- refers to other objects in the content store using an id-based search path

deploymentReferenceArrayProp

Defines the array property class for the [bibus » deploymentReference](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » dashboard » deploymentReferences](#)
- [bibus » drillPath » deploymentReferences](#)
- [bibus » launchable » deploymentReferences](#)
- [bibus » pageletInstance » deploymentReferences](#)
- [bibus » personalization » deploymentReferences](#)
- [bibus » portlet » deploymentReferences](#)
- [bibus » portletInstance » deploymentReferences](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » deploymentReference](#)
is encoded as type `tns:deploymentReferenceArray`

deploymentReferenceProp

Defines the simple property class for the [bibus » deploymentReference](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » deploymentReference](#)
is encoded as type `tns:deploymentReference`

deploymentStatusEnumProp

Defines the simple property class for the [bibus » deploymentStatusEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » deploymentDetail » deployedObjectStatus](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » deploymentStatusEnum](#)
is encoded as type `tns:deploymentStatusEnum`

dimensionManagementService

Defines run-time configuration parameters for the [dimensionManagementService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 8.4 — [“Dimension Management Service” on page 1896](#)

This class was added.

Related information:

- [“Dimension management service” on page 8](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

dimsAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsAuditLevel

Specifies the auditing level for the dimensionManagement service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 8.4 — “Dimension Management Service” on page 1896

This property was added.

dimsMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of `2`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus](#) » [dimensionManagementServiceSpecification](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “[Dimension Management Service](#)” on page 1896

This property was added.

dimsPeakNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus](#) » [dimensionManagementServiceSpecification](#) class.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “[Dimension Management Service](#)” on page 1896

This property was added.

dimsQueueLimit

Specifies the number of seconds that a request for the dimensionManagement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 – “[Dimension Management Service](#)” on page 1896

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

dimensionManagementServiceSpecification

Defines the class for dimension management service specifications.

This specification can describe a request to manipulate a dimension.

Refer to the IBM Cognos Business Viewpoint documentation for more information.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the dimensionManagementServiceSpecification class.

Action	Mode	Service	Method
Run	All	dimensionManagementService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.4 – “Dimension Management Service” on page 1896

This class was added.

directory

Defines the object that contains all directory information.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » namespace](#)
- [bibus » tenants](#)

Contained by instances of the following classes

- [bibus » root](#)

Properties

This class has the following properties.

namespaces

Contains the namespaces for the directory.

tenants

Contains the tenants in a multi-tenancy environment.

The tenant user profile is based on the default user profile defined in the Cognos namespace.

See the IBM Cognos Analytics *Administration and Security Guide* for more information about tenant administration.

This property

- must have at most 1 item

dispatcher

Defines run-time configuration parameters for the [dispatcher](#), as well as other services that are managed by the dispatcher.

Values for many of the properties of this class can be acquired from the parent object. Values for these properties can also be acquired by child objects. Property acquisition can be used to simplify the configuration of IBM Cognos Analytics installations. Using [bibus » configurationFolder](#) objects can simplify service administration while providing maximum flexibility.

A dispatcher creates an instance of this class in the content store when it starts, if it cannot locate an object containing a matching [bibus » dispatcher » dispatcherPath](#) property.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » configuration » jmxProxyHostDispatchers](#)

Container Information

Contains instances of the following classes

- [bibus » adaptiveAnalyticsService](#)
- [bibus » agentService](#)
- [bibus » annotationService](#)
- [bibus » batchReportService](#)
- [bibus » contentManagerCacheService](#)
- [bibus » contentManagerService](#)
- [bibus » dataAdvisorService](#)
- [bibus » dataIntegrationService](#)
- [bibus » dataMovementService](#)

- [bibus » deliveryService](#)
- [bibus » dimensionManagementService](#)
- [bibus » eventManagementService](#)
- [bibus » EVService](#)
- [bibus » graphicsService](#)
- [bibus » humanTaskService](#)
- [bibus » idVizService](#)
- [bibus » indexDataService](#)
- [bibus » indexSearchService](#)
- [bibus » indexUpdateService](#)
- [bibus » installedComponent](#)
- [bibus » jobService](#)
- [bibus » logService](#)
- [bibus » metadataService](#)
- [bibus » metricsManagerService](#)
- [bibus » migrationService](#)
- [bibus » mobileService](#)
- [bibus » monitorService](#)
- [bibus » planningAdministrationConsoleService](#)
- [bibus » planningDataService](#)
- [bibus » planningRuntimeService](#)
- [bibus » planningTaskService](#)
- [bibus » powerPlayService](#)
- [bibus » presentationService](#)
- [bibus » queryService](#)
- [bibus » relationalMetadataService](#)
- [bibus » reportDataService](#)
- [bibus » reportService](#)
- [bibus » repositoryService](#)
- [bibus » runTimeState](#)
- [bibus » saCAMService](#)
- [bibus » systemMetricThresholds](#)
- [bibus » systemService](#)

Contained by instances of the following classes

- [bibus » configuration](#)
- [bibus » configurationFolder](#)

Related information:

- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

Properties

This class has the following properties.

aasAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasAuditLevel

Specifies the auditing level for the adaptive analytics service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the adaptive analytics service, low affinity requests are used by client applications of IBM® Cognos® Analytic Applications to retrieve metadata and query information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a adaptive analytics service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the adaptive analytics service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakMaximumProcesses

Specifies the maximum number of adaptive analytics service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

aasPeakNonAffineConnections

Specifies the number of connections that a adaptive analytics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a adaptive analytics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

adaptiveAnalyticsService

Contains the adaptive analytics service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – “Adaptive Analytics Service” on page 1856

This property was added.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

agentService

Contains the agent service for this dispatcher.

This property

- must have at most 1 item

annotationService

Contains the annotation service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

ansAnnotationLifetime

Specifies the lifetime of the annotation in XML Schema 1.0 type `xs:duration` form.

The lifetime is the length of time in days after the entry associated with the annotation is deleted. For example, if the lifetime for an annotation is set to 60 days, the annotation is deleted 60 days after the associated report is deleted.

For an [annotation](#) object associated with a [reportVersion](#) object, the annotations are deleted when the report is deleted.

The default is 180 days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of `P180D`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

ansAuditLevel

Specifies the auditing level for the annotation service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Annotation Service” on page 1864

This property was added.

asAuditLevel

Specifies the auditing level for the agent service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

asConnections

Specifies the maximum number of connections that a process of the agent service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

asMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the agent service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

asPeakConnections

Specifies the number of connections that a agent service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

batchReportService

Contains the batch report service for this dispatcher.

This property

- must have at most 1 item

brsAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsAuditLevel

Specifies the auditing level for the batch report service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

brsAuditNativeQuery

Specifies whether to log native query information for the batch report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

brsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of `0` to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 — “Chart Hotspot Limit” on page 1909

This property was added.

brsDataSourceChange

Specifies the change time of the data source for the batch report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

brsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`

- has a default value of 0
- can be acquired from a containing object

brsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the batch report service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – [“New Email Configuration Parameters” on page 1868](#)

This property was added.

brsMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

brsNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

brsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the batch report service.

This property

- is of type [bibus » pdfCharacterEncodingEnum](#)
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of [auto](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPDFCompressionLevel

Specifies the compression level for PDF documents created by the batch report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » dispatcher » brsPDFCompressionType](#) property

brsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the batch report service.

This property

- is of type [bibus » pdfCompressionTypeEnum](#)
 - is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » dispatcher » brsPDFCompressionLevel](#) property

brsPDFEmbedFonts

Specifies whether the batch report service should embed fonts in generated PDF documents.

This property

- is of type `bibus » pdfFontEmbeddingEnum`
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of `allow`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

brsPeakAffineConnections

Specifies the number of connections that a batch report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a batch report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

brsPeakMaximumProcesses

Specifies the maximum number of batch report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

brsPeakNonAffineConnections

Specifies the number of connections that a batch report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a batch report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 4.

capacity

Specifies the processing capacity of a dispatcher.

Use with the `bibus » dispatcher` class to indicate the processing capacity of a dispatcher relative to other dispatchers.

For example, if the first dispatcher is twice as fast as the second, set the capacity of the first to 2.0 and the capacity of the second to 1.0. Incoming requests will be directed to these dispatchers in the same relative proportion (2-to-1). That is, the first dispatcher will receive two-thirds of the requests.

Use with the `bibus » configuration` class and the `bibus » configurationFolder` class to indicate the processing capacity of all dispatchers in the configuration folder.

When you add a dispatcher to the configuration folder, it automatically inherits the configuration settings of the folder. However, if you change the default values of a dispatcher or service, the dispatcher's properties will take precedence over those of the `bibus » configurationFolder` class.

This property

- is of type `float`
 - is encoded as type `tns:floatProp`
- has a default value of 1.0
- must contain a value greater than 0.0
- can be acquired from a containing object

cmcsAuditLevel

Specifies the auditing level for the content manager cache service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmcsHeapLimit

Specifies the upper limit of the cache size for the Content Manager cache service. This value is represented as a percentage of the JVM heap size.

The default is 10%.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- can be acquired from a containing object

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This property was added.

cmsAuditLevel

Specifies the auditing level for the content manager service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

cmsConnections

Specifies the maximum number of connections that a process of the content manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

cmsPeakConnections

Specifies the number of connections that a content manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

configuration – obsolete

contentManagerCacheService

This property

- must have at most 1 item

New in Version 10.1.0 – [“Content Manager Cache Service” on page 1862](#)

This property was added.

contentManagerService

Contains the Content Manager service for this dispatcher.

This property

- must have at most 1 item

dasAuditLevel

Specifies the auditing level for the data advisor service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Updated Support for IBM Cognos Express” on page 1871](#)

This property was added.

dataAdvisorService

Contains the data advisor service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – [“Updated Support for IBM Cognos Express” on page 1871](#)

This property was added.

dataIntegrationService

Contains the data integration service for this dispatcher.

This property

- must have at most 1 item

dataMovementService

Contains the data movement service for this dispatcher.

This property

- must have at most 1 item

deliveryService

Contains the delivery service for this dispatcher.

This property

- must have at most 1 item

dimensionManagementService

Reserved.

This property

- must have at most 1 item

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsAuditLevel

Specifies the auditing level for the dimensionManagement service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus](#) » [dimensionManagementServiceSpecification](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a dimensionManagement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

There are no secondary requests associated with the dimension management service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakMaximumProcesses

Specifies the maximum number of dimensionManagement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Dimension Management Service” on page 1896

This property was added.

dimsPeakNonAffineConnections

Specifies the number of connections that a dimensionManagement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a dimensionManagement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests are used to run a [bibus » dimensionManagementServiceSpecification](#) class.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — [“Dimension Management Service” on page 1896](#)

This property was added.

dimsQueueLimit

Specifies the number of seconds that a request for the dimensionManagement service can be queued before it exceeds the timeout period.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — [“Dimension Management Service” on page 1896](#)

This property was added.

disAuditLevel

Specifies the auditing level for the data integration service.

This property

- is of type `bibus » auditLevelEnum`

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

disConnections

Specifies the maximum number of connections that a process of the data integration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

dispatcherAuditLevel

Specifies the auditing level for the dispatcher(s) .

This property

- is of type `bibus » auditLevelEnum`

is encoded as type `tns:auditLevelEnumProp`

- has a default value of [minimal](#)
- can be acquired from a containing object

dispatcherID

Specifies the global unique identifier (GUID) for the dispatcher. The value of this property is assigned by Content Manager when the object is created.

The property value does not contain any sensitive configuration information, such as the server name, port number, or path of the dispatcher.

This property

- is of type [bibus » guid](#)

is encoded as type `tns:guidProp`

- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

dispatcherPath

Specifies the location (path) of the IBM Cognos Analytics dispatcher in the server environment.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

- can contain at least 255 characters
- is searchable

disPeakConnections

Specifies the number of connections that a data integration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

dmsAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

New in Version 8.3 – [“Data Movement Service Configuration Parameters” on page 1917](#)

This property was added.

dmsAuditLevel

Specifies the auditing level for the data movement service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

dmsConnections – deprecated

Specifies the maximum number of connections that a process of the data movement service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `4`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.3 – [“Data Movement Service Configuration Parameters” on page 1917](#)

This property is deprecated. Use the following properties instead:

- `bibus » dispatcher » dmsAffineConnections` property
- `bibus » dispatcher » dmsNonAffineConnections` property
- `bibus » dispatcher » dmsPeakAffineConnections` property
- `bibus » dispatcher » dmsPeakNonAffineConnections` property

dmsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsPeakAffineConnections

Specifies the number of connections that a data movement service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a data movement service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to cancel tasks, remove inactive requests from the service cache, and to notify the server that the client is still waiting for the task to complete.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsPeakMaximumProcesses

Specifies the maximum number of data movement service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

dmsPeakNonAffineConnections

Specifies the number of connections that a data movement service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a data movement service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute primary requests, such as requests to run a [bibus » dataMovementServiceSpecification](#) or a [bibus » dataMovementTask](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dmsQueueLimit

Specifies the number of seconds that a request for the data movement service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.3 — “Data Movement Service Configuration Parameters” on page 1917

This property was added.

dsAuditLevel

Specifies the auditing level for the delivery service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

dsCompressAttachmentLimit

Specifies the maximum size, in MB, of an uncompressed email attachment. The delivery service will compress an attachment that is larger than the maximum size before sending it.

Use a value of 0 to disable email attachment compression.

Use a `nil` value to compress all email attachments.

Setting a non-`nil`, non-zero value improves performance when sending email messages with large attachments, such as report outputs.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.3 — “Email Delivery Enhancements” on page 1916

This property was added.

dsConnections

Specifies the maximum number of connections that a process of the delivery service can use concurrently to execute requests.

For more information, see “Tune Server Performance” on page 89.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

dsMaximumEMailSize

Specifies the maximum amount of data in MB that can be placed in an email by the delivery service.

Base this value on your mail server configuration. Please contact your mail server administrator to determine an appropriate value. If email sizes exceed that which your mail server is configured to support, consider a different transfer method.

If the uncompressed size of the email exceeds the specified configuration parameter value, the `deliveryService` service removes the largest attachments from the message until the total size of the message is lower than the specified configuration parameter value. If an attachment is removed, an error message is created and added as a plain text attachment to the email indicating that the message size has exceeded the configured limit. If the uncompressed size of the email body (for example, an HTML report) exceeds the configured limit, the error message replaces the content of the body.

Use a value of 0 to specify that the total size of an email is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

dsPeakConnections

Specifies the number of connections that a delivery service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

edition

Specifies the edition of all services running under the control of this dispatcher.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

emsAuditLevel

Specifies the auditing level for the event management service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

eventManagementService

Contains the event management service for this dispatcher.

This property

- must have at most 1 item

evsAuditLevel

Specifies the auditing level for the EV service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was added.

EVService

Contains the EV service for this dispatcher.

This property

- must have at most 1 item

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was added.

graphicsService

Contains the graphics service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

gsAuditLevel

Specifies the auditing level for the graphics service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

gsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
- is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

gsNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the graphics service, low affinity requests process requests to generate graphics (charts) for report run requests.

This property

- is of type `int`
- is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

New in Version 10.2.1 – [“Performance enhancements” on page 1830](#)

Changed the default value to 50.

gsPeakNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsQueueLimit

Specifies the number of seconds that a request for the graphics service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

htsAuditLevel

Specifies the auditing level for the human task service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

htsCompletedTaskLifetime

Specifies the lifetime of completed human tasks in XML Schema 1.0 type `xs:duration` form. If the lifetime is set to `P90D`, the human task will be deleted after 90 days if all linked reports or dashboards are deleted.

The default lifetime is ninety days.

This property

- is of type `duration`

is encoded as type `tns:durationProp`

- has a default value of `P90D`
- can be acquired from a containing object

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

humanTaskService

Contains the human task service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

idsAuditLevel

Specifies the auditing level for the index data service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

idsConnections

Specifies the maximum number of connections that a process of the index data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

idsPeakConnections

Specifies the number of connections that a index data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

idVizAuditLevel

Specifies the auditing level for the idViz service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.2.0 – “Support for interactive discovery and visualization” on page 1839

This property was added.

idVizService

Reserved.

This property

- must have at most 1 item

New in Version 10.2.0 – “Support for interactive discovery and visualization” on page 1839

This property was added.

indexDataService

Contains the index data service for this dispatcher.

This property

- must have at most 1 item

indexSearchService

Contains the index search service for this dispatcher.

This property

- must have at most 1 item

indexUpdateService

Contains the index update service for this dispatcher.

This property

- must have at most 1 item

installedComponents

Contains the components installed with this dispatcher.

issAuditLevel

Specifies the auditing level for the index search service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

issConnections

Specifies the maximum number of connections that a process of the index search service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

issPeakConnections

Specifies the number of connections that a index search service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

iusAuditLevel

Specifies the auditing level for the index update service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

iusConnections

Specifies the maximum number of connections that a process of the index update service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

iusPeakConnections

Specifies the number of connections that a index update service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

jobService

Contains the job service for this dispatcher.

This property

- must have at most 1 item

jsAuditLevel

Specifies the auditing level for the job service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

jsConnections

Specifies the maximum number of connections that a process of the job service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

jsmNonPeakDemandBeginHour – obsolete

jsmNonPeakDemandMaximumJobs – obsolete

jsmPeakDemandBeginHour – obsolete

jsmPeakDemandMaximumJobs – obsolete

jsPeakConnections

Specifies the number of connections that a job service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

loadBalancingMode

Specifies the load balancing mode for a dispatcher or group of dispatchers.

Use this property to indicate which load balancing algorithm should be used for

- a specific [bibus » dispatcher](#)
- a group of dispatchers contained within a [bibus » configurationFolder](#)
- all the dispatchers contained within a [bibus » configuration](#) object

This property

- is of type [bibus » loadBalancingModeEnum](#)
 - is encoded as type `tns:loadBalancingModeEnumProp`
- has a default value of [weightedRoundRobin](#)
- can be acquired from a containing object

logService

Contains the log service for this dispatcher.

This property

- must have at most 1 item

IsAuditAdminLevel – obsolete

IsAuditLevel – obsolete

IsAuditNativeQuery – obsolete

IsAuditOtherLevel – obsolete

IsAuditUsageLevel – obsolete

mbsAuditLevel

Specifies the auditing level for the mobile service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mbsConnections

Specifies the maximum number of connections that a process of the mobile service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

mbsPeakConnections

Specifies the number of connections that a mobile service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

mdsAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The `metadataService` does not use this property as this service only processes synchronous requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “[Lineage Metadata](#)” on page 1891

This property was added.

mdsAuditLevel

Specifies the auditing level for the metadata service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “[Lineage Metadata](#)” on page 1891

This property was added.

mdsMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “[Lineage Metadata](#)” on page 1891

This property was added.

mdsNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests process lineage requests that process metadata in published Framework Manager models as well as in the query section of a report specification.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Lineage Metadata” on page 1891](#)

This property was added.

mdsPeakAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Lineage Metadata” on page 1891](#)

This property was added.

mdsPeakMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Lineage Metadata” on page 1891](#)

This property was added.

mdsPeakNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no

benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsQueueLimit

Specifies the number of seconds that a request for the metadata service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

metadataService

Contains the metadata service for this dispatcher.

This property

- must have at most 1 item

metricsManagerService

Contains the IBM Cognos Metric Studio service for this dispatcher.

This property

- must have at most 1 item

migrationService

Contains the migration service for this dispatcher.

This property

- must have at most 1 item

New in Version 8.4 — “Migration Service” on page 1883

This property was added.

misAuditLevel

Specifies the auditing level for the migration service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`

- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was added.

misConnections

Specifies the maximum number of connections that a process of the migration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was added.

misPeakConnections

Specifies the number of connections that a migration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was added.

mmsAuditLevel

Specifies the auditing level for the metrics manager service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mmsConnections

Specifies the maximum number of connections that a process of the metrics manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

mmsPeakConnections

Specifies the number of connections that a metrics manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

mobileService

Contains the mobile service for this dispatcher.

This property

- must have at most 1 item

monitorService

Contains the monitor service for this dispatcher.

This property

- must have at most 1 item

msAuditLevel

Specifies the auditing level for the monitor service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

nonPeakDemandBeginHour

Specifies the hour of the day at which the non-peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 18
- must contain a value greater than or equal to 0

- must contain a value less than or equal to 23
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

pacsauditLevel

Specifies the auditing level for the planning administration console service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

pacsaConnections

Specifies the maximum number of connections that a process of the planning administration console service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

pacsaPeakConnections

Specifies the number of connections that a planning administration console service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

pdsaAuditLevel

Specifies the auditing level for the planning data service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)

- can be acquired from a containing object

pdsConnections

Specifies the maximum number of connections that a process of the planning data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsEListAccessCacheLimit

Specifies the number of seconds that an e-list access rights cache entry can remain in the cache before it must be recalculated.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 3600
- must contain a value greater than or equal to 0
- can be acquired from a containing object

pdsMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

pdsPeakConnections

Specifies the number of connections that a planning data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

pdsPeakMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

pdsShowCellAnnotations

Specifies whether authoring studios should display cell annotations.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

peakDemandBeginHour

Specifies the hour of the day at which the peak demand time begins.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 7
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 23
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added. It replaces the `bibus » dispatcher » msPeakDemandBeginHour` property.

planningAdministrationConsoleService

Reserved.

This property

- must have at most 1 item

planningDataService

Contains the planning data service for this dispatcher.

This property

- must have at most 1 item

planningRuntimeService

Reserved.

This property

- must have at most 1 item

planningTaskService

Reserved.

This property

- must have at most 1 item

powerPlayService

Contains the PowerPlay service for this dispatcher.

This property

- must have at most 1 item

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to retrieve saved output, and also when a request must go to a particular instance of [powerPlayService](#) service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsAuditLevel

Specifies the auditing level for the PowerPlay service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

ppsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

ppsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the PowerPlay service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

ppsNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute a report, and also when a request can be sent to any instance of `powerPlayService` service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

ppsPeakAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

ppsPeakNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsQueueLimit

Specifies the number of seconds that a request for the PowerPlay service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

presentationService

Contains the presentation service for this dispatcher.

This property

- must have at most 1 item

prsAuditLevel

Specifies the auditing level for the planning runtime service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

prsConnections

Specifies the maximum number of connections that a process of the planning runtime service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

prsPeakConnections

Specifies the number of connections that a planning runtime service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

psAuditLevel

Specifies the auditing level for the presentation service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

ptsAuditLevel

Specifies the auditing level for the planning task service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

ptsConnections

Specifies the maximum number of connections that a process of the planning task service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

ptsPeakConnections

Specifies the number of connections that a planning task service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – [“Improve Batch Processing” on page 1902](#)

This property was added.

qsAdditionalJVMArguments

Specifies additional arguments that control the JVM. The arguments may vary depending on the JVM.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsAuditLevel

Specifies the auditing level for the query service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsDiagnosticsEnabled – deprecated

Specifies whether diagnostics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsDisableQueryPlanCache – deprecated

Specifies whether query plans are cached for possible re-use. A query plan represents a set of transformations applied to query data objects to obtain desired query results.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product

- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.1.1 – [“Deprecation of `qsDisableQueryPlanCache`” on page 1849](#)

This property is deprecated.

`qsDisableVerboseGCLogging`

Specifies whether garbage collection information is recorded in a log file.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – [“New `queryService` configuration options” on page 1837](#)

This property was added.

`qsDumpModelToFile`

Specifies whether the model is dumped to a file when a query is run.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

`qsGCPolicy`

Specifies the garbage collection policy used to manage JVM heap storage.

Default: [Generational](#)

This property

- is of type `bibus » gcPolicyEnum`
is encoded as type `tns:anyURIProp`
- can be acquired from a containing object

New in Version 10.2.0 – [“New `queryService` configuration options” on page 1837](#)

This property was added.

`qsGenerateCommentsInNativeSQL`

Specifies whether comments in native SQL are recorded in a log file.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`

- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsIdleConnectionTimeout

Specifies the timeout period, in seconds, for an idle database connection.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 300
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsInitialJVMHeapSize

Specifies the initial size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsInitialJVMMemorySize

Specifies the initial nursery size, in MB, of the JVM. A value of zero indicates that the initial nursery size is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsJVMHeapSizeLimit

Specifies the maximum size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0

- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsJVMMemorySizeLimit

Specifies the maximum size, in MB, of the JVM nursery. A value of zero indicates that the nursery size limit is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- must contain a value greater than or equal to `0`
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsManualCubeStart

Specifies whether ROLAP cubes are started manually.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsMetricsEnabled – deprecated

Specifies whether metrics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsMultiDimensionalQuerySizeLimit

Specifies memory size, in MB, allotted to each locally executed multi-dimensional query. A value of 0 means the memory size is limited to available system memory.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836](#)

This property was added.

qsQueryExecutionTrace

Specifies whether information tracing the execution of queries is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsQueryPlanningTrace

Specifies whether information tracing the development of query plans is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsResultSetCacheQueryTimeThreshold

Specifies the minimum time, in milliseconds, that must elapse during the construction of a result set before it is considered for caching.

Important: The unit of measurement for this configuration parameter is *milliseconds* (1/1000th of a second) whereas other configuration parameters that specify a duration use *seconds*.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 1000
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPCubeAdministrationCommandTimeout

Specifies the time, in seconds, that the `queryService` will wait for internal resources to become available while executing a ROLAP cube administration command. If the specified timeout period has elapsed without resources becoming available, the `queryService` will time out.

Use a value of 0 when you do not want the `queryService` to time out.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 120
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPCubeConfigurations

Defines configuration data for ROLAP cubes.

This property

- is an array of type `bibus » baseROLAPCubeConfiguration`
 - is encoded as type `tns:baseROLAPCubeConfigurationArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsROLAPMemberCacheAliasRoot

Reserved.

This property

- is of type `token`
 - is encoded as type `tns:tokenProp`
- can be acquired from a containing object

New in Version 10.2.0 – [“IBM Cognos Dynamic Cubes Administration” on page 1836](#)

This property was added.

qsVerboseGCLogLimit

Specifies the maximum number of JVM garbage collection cycles that are logged when `bibus » dispatcher » qsDisableVerboseGCLogging` is enabled.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1000
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

queryService

This property

- must have at most 1 item

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

rdsAuditLevel

Specifies the auditing level for the report data service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

rdsGatewayMappings

Contains the mappings between the internal and external PowerPlay gateways.

This property

- is an array of type [bibus](#) » [gatewayMapping](#)
 - is encoded as type `tns:gatewayMappingArrayProp`
- can be acquired from a containing object

rdsMaximumDataSize

Specifies the maximum amount of data that can be read from a content provider in MB.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- must contain a value greater than or equal to 1
- can be acquired from a containing object

relationalMetadataService

Contains the relational metadata service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

reportDataService

Contains the report data service for this dispatcher.

This property

- must have at most 1 item

reportService

Contains the report service for this dispatcher.

This property

- must have at most 1 item

reposAuditLevel

Specifies the auditing level for the repository service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

reposCacheObjTTL

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of `1200`
- can be acquired from a containing object

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

repositoryService

Reserved.

This property

- must have at most 1 item

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

reposNumObjDisk

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of `1000`
- can be acquired from a containing object

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

reposNumObjMem

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`

- has a default value of 100
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

rmdsAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsAuditLevel

Specifies the auditing level for the relational metadata service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsConnections

Specifies the maximum number of connections that a process of the relational metadata service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – [“Relational metadata service” on page 1835](#)

This property was added.

rmdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Writer comment

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsPeakAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

rmdsPeakConnections

Specifies the number of connections that a relational metadata service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Relational metadata service” on page 1835

This property was added.

rmdsPeakNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

rsAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsAuditLevel

Specifies the auditing level for the report service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

rsAuditNativeQuery

Specifies whether to log native query information for the report service.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

rsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of 0 to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 – “Chart Hotspot Limit” on page 1909

This property was added.

rsDataSourceChange

Specifies the change time of the data source for the report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

rsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

rsMaximumEmailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

rsMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

rsNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the report service.

This property

- is of type `bibus » pdfCharacterEncodingEnum`
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of `auto`
- can be acquired from a containing object

New in Version 8.3 – “PDF Configuration Parameters” on page 1915

This property was added.

rsPDFCompressionLevel

Specifies the compression level for PDF documents created by the report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`

- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus](#) » [dispatcher](#) » [rsPDFCompressionType](#) property

rsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the report service.

This property

- is of type [bibus](#) » [pdfCompressionTypeEnum](#)
is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus](#) » [dispatcher](#) » [rsPDFCompressionLevel](#) property

rsPDFEmbedFonts

Specifies whether the report service should embed fonts in generated PDF documents.

This property

- is of type [bibus](#) » [pdfFontEmbeddingEnum](#)
is encoded as type `tns:pdfFontEmbeddingEnumProp`
- has a default value of [allow](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

rsPeakAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsPeakMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

rsPeakNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsQueueLimit

Specifies the number of seconds that a request for the report service can be queued before it exceeds the timeout period.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

saCAMAuditLevel

Specifies the auditing level for the saCAM service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1838](#)

This property was added.

New in Version 10.2.1 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1832](#)

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

saCAMService

Contains the standalone CAM service for this dispatcher.

This property

- must have at most 1 item

New in Version 10.2.0 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1838](#)

This property was added.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

serverGroup

Specifies the server group associated with this object.

Use this property to identify groups of dispatchers in different application server clusters. Dispatchers that share the same value for this property act as a single cluster.

By default, all dispatchers registered in the same content store act as a single, load-balancing cluster. If your installation uses application server clusters, use this property to identify members of different clusters. For example, if you need some of your dispatchers to operate in [weightedRoundRobin](#) mode while others operate in [clusterCompatible](#) mode, use the `serverGroup` property to distinguish these groupings. When you group your dispatchers using this property, requests are only forwarded to a dispatcher that is part of the same server group.

You must use this setting in conjunction with the values you have set in the `bibus » loadBalancingModeEnum` enumeration set to properly contain your dispatchers in groups that use the same load-balancing algorithm. Failure to meet this requirement will result in unpredictable forwarding of requests by dispatchers.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- can be acquired from a containing object

ssAuditLevel

Specifies the auditing level for the system service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

systemService

Contains the system service for this dispatcher.

This property

- must have at most 1 item

dispatcherTransportVar

Defines the class used by the dispatcher to store information concerning the processing of the request.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [dispatcherTransportVars](#)

Properties

This class has the following properties.

name

Specifies the name of the dispatcher transport mechanism.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the value assigned to the dispatcher transport mechanism.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

displayObject

Defines a template used to create a control in an HTML page.

References

Used by the following properties:

- [bibus](#) » [promptInfo](#) » [displayObjects](#)

Properties

This class has the following properties.

caption

Contains the label text associated with the prompt. For example, if a text box appears, the caption element contains the label for the text box.

This property

- is of type `string`
is encoded as type `xs:string`

name

Identifies the form variable used to store the prompt value.

This property

- is of type `string`
is encoded as type `xs:string`

promptOptions

Specifies the possible prompt choices for this displayed object.

This property

- is an array of type `bibus » promptOption`
is encoded as type `tns:promptOptionArray`

type

Specifies the type of user interface control used to prompt the user for information, such as a list box or a text box.

This property

- is of type `string`
is encoded as type `xs:string`

value

Contains the string value used when the display type is hidden.

This property

- is of type `string`
is encoded as type `xs:string`

distributionList

Defines a set of security objects.

If you want to distribute reports to more than one recipient, you can create a distribution list.

If a recipient is not part of the IBM Cognos security system, you can create a contact for that person and add the contact to a distribution list.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- `bibus » adminFolder » contact`
- `bibus » aliasRoot » contact`
- `bibus » annotation » contact`
- `bibus » annotationFolder » contact`
- `bibus » archiveLocation » contact`
- `bibus » baseAgentDefinition » contact`
- `bibus » baseDataIntegrationTask » contact`
- `bibus » baseDataMovementTask » contact`
- `bibus » basePowerPlay8Report » contact`
- `bibus » basePowerPlayClass » contact`

- [bibus » baseReport » contact](#)
- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » group » distributionMembers](#)
- [bibus » role » distributionMembers](#)

- [bibus » distributionList » members](#)
- [bibus » group » members](#)
- [bibus » role » members](#)
- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)

Container Information

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Related information:

- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

members

Refers to the members of this distribution list.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

document

This class defines the structure of documents created by IBM Cognos Analytics for Microsoft® Office.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » shortcutRSSTask » link](#)
- [bibus » shortcut » target](#)

Container Information

Contains instances of the following classes

- [bibus » documentVersion](#)

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)
- [bibus » periodical](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

documentType

This property specifies the media type of the document. The media type is expressed as `type/subtype`. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- can contain at least 255 characters

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

<i>Table 162. Rules for a new document object</i>			
Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » documentVersion class		1	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

versions

Manages the document versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » documentVersion](#)

documentContent

This class contains the document content.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contains instances of the following classes

- [bibus » graphic](#)
- [bibus » page](#)

Contained by instances of the following classes

- [bibus » documentVersion](#)

Properties

This class has the following properties.

burstID

Identifies the values for the burst keys that produced the output.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- can contain at least 32767 characters
- is searchable

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

burstKey

Specifies the label for the burst key values that produced this document content.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- can contain at least 4095 characters

New in Version 8.3 — “Conditional Subscriptions” on page 1899

The documentation for this property was updated.

context

Specifies the context data for the document.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the `bibus » addOptions » dataEncoding` property and the `bibus » queryOptions » dataEncoding` property for additional details.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `bibus » xmlEncodedXML`

is encoded as type `tns:xmlEncodedXMLMIMEProp`

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This property was added.

contextBlockCount

Specifies the number of context blocks stored in the `context` property.

This property

- is of type `int`

is encoded as type `tns:intProp`

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This property was added.

data

Contains the MIME data. The data format is specified in the `dataType` property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to MIME. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

format

Specifies the rendering format.

When used in the [bibus » account](#) class or the [bibus » contact](#) class, this property specifies the preferred output format of reports for the account or contact.

When used in the [bibus » documentContent](#) class or [bibus » output](#) class this property specifies the format of the data contained in the object.

This property

- is of type [bibus » outputFormatEnum](#)
is encoded as type `tns:nmtokenProp`
- must contain no more than 10 characters
- is searchable

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

images

Contains the images for the document content.

lastPage

Specifies the page number for the last page of the document.

This property

- is of type NMTOKEN
 - is encoded as type `tns:nmtokenProp`
- can contain at least 255 characters

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

locale

Specifies the locale for this object. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

When used by [bibus](#) » [contact](#), this property determines the language and data format of the returned content.

Use the appropriate language so that users understand object names and search paths. Use the appropriate region so that date, time, and currency values are presented in the proper localized format.

This property

- is of type Language
 - is encoded as type `tns:languageProp`
- must contain no more than 64 characters
- is searchable

pages

Contains the pages of HTML output for the document.

The first page of the document is stored within this object to decrease the time needed to access the first page.

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

recipients

Refers to the burst document content recipients.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

recipientsEMail

Contains the set of email recipients for the burst document content.

This property

- is an array of type string

is encoded as type `tns:stringArrayProp`

documentVersion

This class contains the document content.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » documentContent](#)

Contained by instances of the following classes

- [bibus » document](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

content

Contains the content for this document version.

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`

is encoded as type `tns:dateTimeProp`

- is read-only
- is searchable

New in Version 8.4 – “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

drillOption

Defines the abstract base class for all drill option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » drillOptionParameterValues](#)

Properties

This class has the following properties.

name

Identifies the drill option.

This property

- is of type [bibus » drillOptionEnum](#)
is encoded as type `tns:drillOptionEnum`

drillOptionParameterValues

Defines [bibus » parameterValue](#) values for the drill options.

This class

- inherits properties from the [bibus » drillOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the drill option.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

drillPath

Defines a drill-through path between a source and a target.

Drill-through paths defined in a [bibus » package](#) object are used by all reports that are based on that package.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [asynchDetailDrillPathObject](#) » [drillPath](#)

Used by the following method parameters:

- [drillThrough](#) » [addDrillPath\(parentPath, object, options\)](#) » [object](#)
- [drillThrough](#) » [updateDrillPath\(object, options\)](#) » [object](#)

Used by the following method return values:

- [drillThrough](#) » [addDrillPath\(parentPath, object, options\)](#) » [result](#)
- [drillThrough](#) » [updateDrillPath\(object, options\)](#) » [result](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `drillPath` class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus](#) » [package](#)

What's new

New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889

This class can now be used with the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method.

Properties

This class has the following properties.

action

Specifies the action for the target object.

This property

- is of type [bibus](#) » [baseReportActionEnum](#)
 - is encoded as type `tns:baseReportActionEnumProp`
- has a default value of [viewOutput](#)

bookmarkItem

Specifies the data item from the source to be used to determine the target bookmark within the drill path target.

This property

- is of type [bibus » metadataModelItemName](#)
is encoded as type `tns:metadataModelItemNameProp`

bookmarkText

Specifies the target bookmark within the drill path target.

This property

- is of type `string`
is encoded as type `tns:stringProp`

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)
is encoded as type `tns:deploymentReferenceArrayProp`

New in Version 8.4 — “Supporting New Drill-through Targets” on page 1890

This property was added. It replaces the [bibus » drillPath » target](#) property.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameterAssignments

Contains the parameter assignments for the drill path.

This property

- is an array of type [bibus » baseParameterAssignment](#)

is encoded as type `tns:baseParameterAssignmentArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArrayProp`

scope

Specifies the source domain from which this drill path may be used.

This property

- is an array of type `bibus » metadataModelItemName`
is encoded as type `tns:metadataModelItemNameArrayProp`

specification

Specifies the information required to perform a drill-through operation between two resources, including information about the target and the action to be performed on the target. The specification may also include mapping between source and target parameters.

This property

- is of type `bibus » reportServiceDrillThroughSpecification`
is encoded as type `tns:reportServiceDrillThroughSpecificationProp`

New in Version 8.4 — “Supporting New Drill-through Targets” on page 1890

This property was added. It replaces the `bibus » drillPath » target` property.

target

Specifies the drill-through target.

If the target was moved, deleted, or renamed since the drill path was specified, the drill-through will fail.

This property

- is an array of type `bibus » baseClass`
has items that must be of class `bibus » analysis`, `bibus » query`, `bibus » report`, or `bibus » reportView`
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 8.4 — “Supporting New Drill-through Targets” on page 1890

This property is deprecated. Use the `bibus » drillPath » deploymentReferences` property instead.

targetOptions

Contains the set of options to be passed to the drill-through target to perform the drill-through. Applicable options vary depending on the class of the target object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type `bibus » option`

is encoded as type `tns:optionArrayProp`

New in Version 8.4 – “Supporting New Drill-through Targets” on page 1890

This property was added.

targetParameters

Contains the set of parameter values to be passed to the drill-through target to perform the drill-through.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)

is encoded as type `tns:parameterValueArrayProp`

New in Version 8.4 – “Supporting New Drill-through Targets” on page 1890

This property was added.

drillThroughAction

Defines the class that contains information about drill-through actions.

Properties

This class has the following properties.

name

Specifies the name of the drill-through action.

This property

- is of type `string`

is encoded as type `xs:string`

uri

Specifies the URI that triggers the action.

This property

- is of type `anyURI`

is encoded as type `xs:string`

drillThroughOption

Defines the abstract base class for all drill-through option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » drillThroughOptionAnyURI](#)
- [bibus » drillThroughOptionBoolean](#)
- [bibus » drillThroughOptionXMLEncodedXML](#)

Properties

This class has the following properties.

name

Identifies the drill-through option.

This property

- is of type [bibus » drillThroughOptionEnum](#)

is encoded as type `tns:drillThroughOptionEnum`

drillThroughOptionAnyURI

Defines anyURI values for the drill-through options.

This class

- inherits properties from the [bibus » drillThroughOption](#) class

What's new

New in Version 8.4 — “Drill-Through Improvements” on page 1889

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the drill-through option.

This property

- is of type anyURI

is encoded as type `xs:string`

drillThroughOptionBoolean

Defines boolean values for the drill-through options.

This class

- inherits properties from the [bibus » drillThroughOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the drill-through option.

This property

- is of type boolean

is encoded as type `xs:boolean`

drillThroughOptionXMLEncodedXML

Defines xmlEncodedXML values for the drill-through options.

This class

- inherits properties from the [bibus » drillThroughOption](#) class

What's new

New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the drill-through option.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

drillThroughPath

Defines the class that contains information about drill-through paths in the content store.

Properties

This class has the following properties.

name

Specifies the name of the drill-through path.

This property

- is of type `string`
is encoded as type `xs:string`

searchPath

Specifies the search path of the drill-through path in the content store.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

durationProp

Defines the simple property class for the duration.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » annotationService » ansAnnotationLifetime](#)
- [bibus » configuration » ansAnnotationLifetime](#)
- [bibus » configurationFolder » ansAnnotationLifetime](#)

- [bibus » dispatcher » ansAnnotationLifetime](#)
- [bibus » configuration » htsCompletedTaskLifetime](#)
- [bibus » configurationFolder » htsCompletedTaskLifetime](#)
- [bibus » dispatcher » htsCompletedTaskLifetime](#)
- [bibus » humanTaskService » htsCompletedTaskLifetime](#)
- [bibus » schedule » intradayRecurrenceInterval](#)
- [bibus » configuration » periodicalDocumentVersionRetentionAge](#)
- [bibus » configuration » temporaryObjectLifetime](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `duration`
- is encoded as type `xs:string`

environmentVar

Defines the class used to store information about an environment variable.

References

Used by the following properties:

- [bibus » hdrSession » environmentVars](#)

Properties

This class has the following properties.

name

Specifies the name of the Web server gateway environment variable.

This property

- is of type `string`
- is encoded as type `xs:string`

value

Specifies the value assigned to the Web server gateway environment variable.

This property

- is of type `string`
- is encoded as type `xs:string`

eventManagementService

Defines run-time configuration parameters for the [eventManagementService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

emsAuditLevel

Specifies the auditing level for the event management service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

eventManagementServiceSpecification

Defines the type for event management service specifications.

An event specification specifies filtering information required to query future events. The [bibus » eventManagementService](#) manages future events that are in a scheduled or cancelled state.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `eventManagementServiceSpecification` class.

Action	Mode	Service	Method
Run	All	eventManagementService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.3 – “Schedule Management” on page 1907

This class was added.

eventRecord

Defines an event.

An event occurs when a scheduled task runs.

References

Used by the following method parameters:

- [event » updateEvents\(events\) » events](#)

Properties

This class has the following properties.

eventID

Identifies the event.

This property

- is of type `token`
 - is encoded as type `xs:string`
- is read-only

- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase

history

Refers to the history for the event.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [history](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

priority

Specifies the event priority. Higher priority tasks run before lower priority tasks. Priority values range from 1 to 5. 1 is the lowest priority; 5 is the highest.

This property

- is of type `int`
 - is encoded as type `xs:int`
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 5

New in Version 8.3 – “Schedule Priority” on page 1912

This property was added.

requestedStartTime

Specifies the requested start time for the event, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `xs:dateTime`
- is read-only

runnable

Refers to the object that runs during the event.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [baseAgentDefinition](#), [bibus](#) » [baseDataIntegrationTask](#), [bibus](#) » [baseDataMovementTask](#), [bibus](#) » [basePowerPlay8Report](#), [bibus](#) » [baseReport](#), [bibus](#) » [baseRSSTask](#), [bibus](#) » [contentTask](#), [bibus](#) » [exportDeployment](#), [bibus](#) » [humanTask](#), [bibus](#) » [importDeployment](#), [bibus](#) » [indexUpdateTask](#), [bibus](#) » [jobDefinition](#), [bibus](#) » [memo](#), [bibus](#) » [migrationTask](#), [bibus](#) » [planningMacroTask](#), [bibus](#) » [planningTask](#), [bibus](#) » [queryServiceTask](#), [bibus](#) » [storedProcedureTask](#), or [bibus](#) » [webServiceTask](#)
 - is encoded as type `tns:baseClassArray`
- is read-only
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property was extended to allow instances of the [bibus » dataMovementTaskAlias](#) class to be referenced by instances of this property.

New in Version 8.4 – “Migration Service” on page 1883

This property was extended to allow instances of the [bibus » migrationTask](#) class to be referenced by instances of this property.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

This property was extended to allow instances of the [bibus » queryServiceTask](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Human Task” on page 1862

This property was extended to allow instances of the [bibus » humanTask](#) class to be referenced by instances of this property.

status

Specifies the run status of the event.

This property

- is of type [bibus » runStatusEnum](#)
 - is encoded as type `xs:string`
- is read-only

eventTypeEnumArrayProp

Defines the array property class for the [bibus » eventTypeEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » agentTaskDefinition » eventTypes](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » eventTypeEnum](#)
 - is encoded as type `tns:eventTypeEnumArray`

eventTypeEnumProp

Defines the simple property class for the [bibus » eventTypeEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » eventTypeEnum](#)
is encoded as type `tns:eventTypeEnum`

EVService

Defines run-time configuration parameters for the EVService .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

This class was added.

Related information:

- [“EV service” on page 8](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

evsAuditLevel

Specifies the auditing level for the EV service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

exportDeployment

Defines the specification for a particular export deployment.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- `bibus » asynchDetailEventRecord » runnable`
- `bibus » eventRecord » runnable`
- `bibus » jobStepDefinition » stepObject`

- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `exportDeployment` class.

<i>Table 165. Services and methods for the <code>exportDeployment</code> class.</i>			
Action	Mode	Service	Method
Run	All	contentManagerService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus](#) » [history](#)
- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [adminFolder](#)

Properties

This class has the following properties.

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » <u>history</u> class		1	bibus » <u>baseClass</u> » <u>creationTime</u> property

This property

- is an array of type bibus » retentionRule
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

faultDetail

Defines the type for the Simple Object Access Protocol (SOAP) detail element.

References

Used by the following properties:

- [bibus » deploymentDetail » message](#)
- [bibus » historyDetailMigrationService » message](#)
- [bibus » asynchDetailMessages » messages](#)
- [bibus » faultDetailArrayProp » value](#)
- [bibus » faultDetailProp » value](#)

Properties

This class has the following properties.

errorCode

Identifies the error that caused the fault.

This property

- is of type `string`
is encoded as type `xs:string`

message

Contains the messages for the fault.

This property

- is an array of type [bibus » faultDetailMessage](#)
is encoded as type `tns:faultDetailMessageArray`

severity

Specifies the severity of the fault.

This property

- is of type `string`
is encoded as type `xs:string`

faultDetailArrayProp

Defines the array property class for the [bibus » faultDetail](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » deploymentDetail » message](#)
- [bibus » historyDetailMigrationService » message](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [faultDetail](#)
is encoded as type `tns: faultDetailArray`

faultDetailMessage

Defines the type for messages in a Simple Object Access Protocol (SOAP) fault.

References

Used by the following properties:

- [bibus](#) » [faultDetail](#) » [message](#)

Properties

This class has the following properties.

message

Specifies details about the SOAP fault or error, in the form of a readable text message.

This property

- is of type `string`
is encoded as type `xs:string`

nestingLevel

Specifies the nesting level for the SOAP fault message.

This property

- is of type `int`
is encoded as type `xs:int`

faultDetailProp

Defines the simple property class for the [bibus](#) » [faultDetail](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [faultDetail](#)
is encoded as type `tns: faultDetail`

favoritesFolder

Contains objects that refer to the user's favorites.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » favoritesFolder](#)
- [bibus » shortcut](#)
- [bibus » URL](#)

Contained by instances of the following classes

- [bibus » account](#)
- [bibus » favoritesFolder](#)

What's new

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

items

Contains the favourite objects for the account.

floatProp

Defines the simple property class for the `float`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » configuration » capacity](#)
- [bibus » configurationFolder » capacity](#)
- [bibus » dispatcher » capacity](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `float`
is encoded as type `xs:float`

folder

Contains the set of application content objects, such as queries, reports, job definitions, and other folders.

Use instances of this class to organize objects in the web portal, so they are easier to find and manipulate.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » baseRSSTask » channel](#)
- [bibus » shortcutRSSTask » link](#)
- [bibus » account » portalPages](#)
- [bibus » shortcut » target](#)

Container Information

Contains instances of the following classes

- [bibus » baseAgentDefinition](#)
- [bibus » baseDataIntegrationTask](#)
- [bibus » baseDataMovementTask](#)
- [bibus » basePowerPlay8Report](#)
- [bibus » basePowerPlayClass](#)
- [bibus » baseReport](#)
- [bibus » dashboard](#)
- [bibus » document](#)
- [bibus » folder](#)

- [bibus » jobDefinition](#)
- [bibus » launchable](#)
- [bibus » package](#)
- [bibus » pagelet](#)
- [bibus » planningTask](#)
- [bibus » shortcut](#)
- [bibus » URL](#)

Contained by instances of the following classes

- [bibus » account](#)
- [bibus » content](#)
- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

effectiveUserCapabilities

Contains the current user's capabilities for the object. This value is determined by intersecting the set of user capabilities granted by [bibus » folder » userCapabilityPolicies](#) property with the set of user capabilities granted to the user globally, as returned by [bibus » session » userCapabilities](#) property.

In order for a capability to be effective for the object, it must be granted to the user both globally and for the object.

This property

- is an array of type [bibus » userCapabilityEnum](#)
 - is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

New in Version 10.1.0 – “Object Capabilities Properties” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

items

Contains the child objects for this object.

New in Version 8.3 – “Package Hierarchies” on page 1913

This property was extended to allow instances of the bibus » package class to be contained by instances of this property.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the bibus » basePowerPlay8Report class to be contained by instances of this property.

New in Version 8.4 – “Dashboards” on page 1894

This property was extended to allow instances of the bibus » dashboard class to be contained by instances of this property.

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was extended to allow instances of the bibus » launchable class to be contained by instances of this property.

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property was extended to allow instances of the bibus » dataMovementTaskAlias class to be contained by instances of this property.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the bibus » baseClass » searchPath property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the bibus » baseClass » ancestors property instead.

powerPlay8Configuration

Specifies the PowerPlay configuration data for this object.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was added.

repositoryRules

This property

- is an array of type [bibus » repositoryRule](#)
is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The [bibus » account](#), [bibus » namespace](#), and [bibus » namespaceFolder](#) classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

userCapabilities

Contains the current user's capabilities for the object, as defined by the [userCapabilityPolicies](#) property.

This property

- is an array of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

New in Version 10.1.0 – “Object Capabilities Properties” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

userCapabilityPolicies

Contains the user capability policy for the object.

This property

- is an array of type [bibus](#) » [userCapabilityPolicy](#)
is encoded as type `tns:userCapabilityPolicyArrayProp`
- can be acquired from a containing object

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

formFieldVar

Defines the class used to store information about an HTML form field variable or a CGI query parameter.

For example, instances of this class can be used to store the user name and password form fields from the IBM Cognos Analytics logon form.

References

Used by the following properties:

- [bibus](#) » [hdrSession](#) » [formFieldVars](#)

Properties

This class has the following properties.

format

Specifies whether the information in the HTML form field or CGI query parameter is encrypted.

This property

- is of type [bibus](#) » [formatEnum](#)
is encoded as type `tns:formatEnum`
- has a default value of [not_encrypted](#)

name

Specifies the name of the HTML form field or CGI query parameter.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the value of the HTML form field or CGI query parameter.

This property

- is of type `string`

is encoded as type `xs:string`

gatewayMapping

Defines mappings between the internal and external PowerPlay gateways.

References

Used by the following properties:

- [bibus » configuration » rdsGatewayMappings](#)
- [bibus » configurationFolder » rdsGatewayMappings](#)
- [bibus » dispatcher » rdsGatewayMappings](#)
- [bibus » reportDataService » rdsGatewayMappings](#)
- [bibus » gatewayMappingArrayProp » value](#)
- [bibus » gatewayMappingProp » value](#)

Properties

This class has the following properties.

externalURI

Specifies the external PowerPlay URI.

This is the URI of the user-accessible PowerPlay gateway outside the firewall.

This property

- is of type `anyURI`
is encoded as type `xs:string`

internalURI

Specifies the internal PowerPlay URI.

This is the URI of the PowerPlay gateway inside the firewall. This URI corresponds to the URI specified by [externalURI](#) used by the [bibus » reportDataService](#) service.

This property

- is of type `anyURI`
is encoded as type `xs:string`

gatewayMappingArrayProp

Defines the array property class for the [bibus » gatewayMapping](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » configuration » rdsGatewayMappings](#)
- [bibus » configurationFolder » rdsGatewayMappings](#)
- [bibus » dispatcher » rdsGatewayMappings](#)
- [bibus » reportDataService » rdsGatewayMappings](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » gatewayMapping](#)
is encoded as type `tns:gatewayMappingArray`

gatewayMappingProp

Defines the simple property class for the [bibus » gatewayMapping](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » gatewayMapping](#)
is encoded as type `tns:gatewayMapping`

genericOption

Defines the abstract base class for all generic option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » genericOptionAnyURI](#)
- [bibus » genericOptionBoolean](#)
- [bibus » genericOptionStringArray](#)
- [bibus » genericOptionXMLEncodedXML](#)

What's new

New in Version 10.1.0 – [“IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858](#)

This class was added.

New in Version 10.1.0 – [“Content Manager Cache Service” on page 1862](#)

This class was added.

New in Version 10.1.0 – [“Query Modes” on page 1865](#)

This class was added.

Properties

This class has the following properties.

name

Identifies the generic option.

This property

- is of type `anyURI`
is encoded as type `xs:string`

genericOptionAnyURI

Defines `anyURI` values for the generic options.

This class

- inherits properties from the [bibus » genericOption](#) class

What's new

New in Version 10.1.0 – [“Query Modes” on page 1865](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the generic option.

This property

- is of type `anyURI`
is encoded as type `xs:string`

genericOptionBoolean

Defines `boolean` values for the generic options.

This class

- inherits properties from the [bibus » genericOption](#) class

What's new

New in Version 10.1.0 – [“IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the generic option.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

genericOptionStringArray

Defines string values for the generic options.

This class

- inherits properties from the [bibus » genericOption](#) class

What's new

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the generic option.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

genericOptionXMLEncodedXML

Defines [bibus » xmlEncodedXML](#) values for the generic options.

This class

- inherits properties from the [bibus » genericOption](#) class

What's new

New in Version 10.1.0 – “Content Manager Cache Service” on page 1862

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the generic option.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

governor

Defines the base class for the governors used by IBM Cognos Analytics.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus](#) » [governorBoolean](#)
- [bibus](#) » [governorInt](#)

References

Used by the following properties:

- [bibus](#) » [group](#) » [governors](#)
- [bibus](#) » [role](#) » [governors](#)
- [bibus](#) » [governorArrayProp](#) » [value](#)
- [bibus](#) » [governorProp](#) » [value](#)

Properties

This class has the following properties.

name

Identifies the governor.

This property

- is of type [bibus](#) » [governorEnum](#)
is encoded as type `tns:governorEnum`

[governorArrayProp](#)

Defines the array property class for the [bibus](#) » [governor](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [group](#) » [governors](#)
- [bibus](#) » [role](#) » [governors](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [governor](#)
is encoded as type `tns:governorArray`

[governorBoolean](#)

Defines the type for governors that have a value that is type `boolean`.

This class

- inherits properties from the [bibus](#) » [governor](#) class

Properties

This class has the following properties.

value

Specifies the value of the governor.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

governorInt

Defines the type for governors that have a value that is type `int`.

This class

- inherits properties from the [bibus » governor](#) class

Properties

This class has the following properties.

value

Specifies the integer value of the governor.

This property

- is of type `int`

is encoded as type `xs:int`

governorProp

Defines the simple property class for the [bibus » governor](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » governor](#)

is encoded as type `tns:governor`

graphic

Contains a graphic image used in a report or query.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » cacheOutput](#)
- [bibus » documentContent](#)
- [bibus » output](#)
- [bibus » session](#)

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
 - is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains a URI specifying the location of binary data for this object. The URI can specify a location in the content store or the location of an external repository.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is read-only

New in Version 10.2.1 – “[External object store for report archiving](#)” on page 1829

This property was added.

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property `data`. The media type is expressed as a `type/subtype` pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

- can contain at least 255 characters

graphicsService

Defines run-time configuration parameters for the graphics service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This class was added.

Related information:

- [“Graphics service” on page 8](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

gsAuditLevel

Specifies the auditing level for the graphics service.

This property

- is of type [bibus » auditLevelEnum](#)

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

gsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

gsNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For the graphics service, low affinity requests process requests to generate graphics (charts) for report run requests.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Graphics Service” on page 1860](#)

This property was added.

New in Version 10.2.1 – [“Performance enhancements” on page 1830](#)

Changed the default value to 50.

gsPeakNonAffineConnections

Specifies the number of connections that a graphics service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a graphics service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

New in Version 10.2.1 – “Performance enhancements” on page 1830

Changed the default value to 50.

gsQueueLimit

Specifies the number of seconds that a request for the graphics service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 10.1.0 – “Graphics Service” on page 1860

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

group

Contains group information, as stored in an external security provider or as defined in the Cognos namespace.

Groups represent collections of users or accounts that perform similar tasks. Accounts are specified in an external security provider with the exception of the Anonymous account, which is specified in the Cognos namespace.

You can create groups in the Cognos namespace and modify their properties.

If groups are created based on information in a security provider, IBM Cognos Analytics may constrain your ability to manipulate these objects. For example, you may not be able to change the value of a property, or the property may not return a value when queried.

The name of an instance may match the name of a sibling. In this case, a different search strategy must be used to distinguish between objects having the same name.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » dataSourceSignon » consumers](#)
- [bibus » adminFolder » contact](#)
- [bibus » aliasRoot » contact](#)
- [bibus » annotation » contact](#)
- [bibus » annotationFolder » contact](#)
- [bibus » archiveLocation » contact](#)
- [bibus » baseAgentDefinition » contact](#)
- [bibus » baseDataIntegrationTask » contact](#)
- [bibus » baseDataMovementTask » contact](#)
- [bibus » basePowerPlay8Report » contact](#)
- [bibus » basePowerPlayClass » contact](#)
- [bibus » baseReport » contact](#)
- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)

- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)
- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » group » distributionMembers](#)
- [bibus » role » distributionMembers](#)
- [bibus » session » identity](#)
- [bibus » distributionList » members](#)
- [bibus » group » members](#)
- [bibus » role » members](#)
- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)
- [bibus » group » securityMembers](#)
- [bibus » role » securityMembers](#)

Container Information

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Properties

This class has the following properties.

distributionMembers

Refers to the members of this group for the purposes of distribution.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » distributionList](#), or [bibus » group](#)
is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path

New in Version 8.3 — “New Properties for Group and Role Classes” on page 1919

This property was added.

governors

Contains the set of governors for this object.

This property

- is an array of type [bibus » governor](#)
is encoded as type `tns:governorArrayProp`

members

Refers to the members of this group .

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » distributionList](#), or [bibus » group](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

New in Version 8.3 — “New Properties for Group and Role Classes” on page 1919

This property was extended to allow instances of the [bibus » distributionList](#) class to be referenced by instances of this property.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property can now be acquired.

securityMembers

Refers to the members of this group for the purposes of security.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » account` or `bibus » group`
 - is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path

New in Version 8.3 — “New Properties for Group and Role Classes” on page 1919

This property was added.

guid

Defines the global unique identifier (GUID) type.

This class

- inherits properties from the `token`

References

Used by the following properties:

- `bibus » planningApplication » applicationGUID`
- `bibus » dispatcher » dispatcherID`
- `bibus » history » dispatcherID`
- `bibus » asynchDetailIndexData » externalGUID`
- `bibus » archiveDescriptor » reportStoreID`
- `bibus » archiveDescriptor » reportViewStoreID`
- `bibus » ancestorInfo » storeID`
- `bibus » baseClass » storeID`
- `bibus » guidArrayProp » value`
- `bibus » guidProp » value`

guidArrayProp

Defines the array property class for the `bibus » guid` class.

This class

- inherits properties from the `bibus » baseProp` class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » guid](#)
is encoded as type `tns:guidArray`

guidProp

Defines the simple property class for the [bibus » guid](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » planningApplication » applicationGUID](#)
- [bibus » dispatcher » dispatcherID](#)
- [bibus » history » dispatcherID](#)
- [bibus » baseClass » storeID](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » guid](#)
is encoded as type `tns:guid`

hdrSession

Defines the class for information related to the user's session.

Information related to the current user's session includes the following:

- the environment variables for the gateway that received the request
- the HTML form fields or CGI query parameters
- the browser cookies for the request and the response

References

Used by the following properties:

- [bibus » biBusHeader » hdrSession](#)

Properties

This class has the following properties.

cookieVars

Specifies the browser cookies for the request.

This property

- is an array of type [bibus](#) » [cookieVar](#)
is encoded as type `tns:cookieVarArray`

environmentVars

Specifies the environment variables of the Web server gateway that received the request, such as the portal state variables.

This property

- is an array of type [bibus](#) » [environmentVar](#)
is encoded as type `tns:environmentVarArray`

formFieldVars

Specifies the HTML form fields or CGI query parameters.

This property

- is an array of type [bibus](#) » [formFieldVar](#)
is encoded as type `tns:formFieldVarArray`

setCookieVars

Specifies the browser cookies that will be set in the response.

This property

- is an array of type [bibus](#) » [setCookieVar](#)
is encoded as type `tns:setCookieVarArray`

hierarchicalParmValueItem

Use this class to define hierarchical parameter values.

Examples of hierarchical levels are cities within countries and brands within product lines.

This class

- inherits properties from the [bibus](#) » [parmValueItem](#) class

References

Used by the following properties:

- [bibus](#) » [hierarchicalParmValueItem](#) » [subNodes](#)

Properties

This class has the following properties.

subNodes

Contains the values for a lower level in the hierarchy for this value.

This property

- is an array of type [bibus](#) » [hierarchicalParmValueItem](#)
is encoded as type `tns:hierarchicalParmValueItemArray`

value

Specifies the value of a member at a particular level in the hierarchy, such as the country Canada or the city Ottawa.

This property

- is of type [bibus » simpleParmValueItem](#)
is encoded as type `tns:simpleParmValueItem`

history

Provides information about the running of a task.

The object that was run is the [parent](#) of the [bibus » history](#) object.

The [monitorService](#) creates and updates [bibus » history](#) objects when it is called to run a task.

While the [bibus » history](#) class defines information common to all tasks, each service may record additional details in instances of the [bibus » baseHistoryDetail](#) class.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asynchDetailEventRecord » history](#)
- [bibus » eventRecord » history](#)
- [bibus » historyDetailRelatedHistory » related](#)

Container Information

Contains instances of the following classes

- [bibus » baseHistoryDetail](#)

Contained by instances of the following classes

- [bibus » baseAgentDefinition](#)
- [bibus » baseDataIntegrationTask](#)
- [bibus » baseReport](#)
- [bibus » baseRSSTask](#)
- [bibus » contentTask](#)
- [bibus » exportDeployment](#)
- [bibus » humanTask](#)
- [bibus » importDeployment](#)
- [bibus » indexUpdateTask](#)
- [bibus » jobDefinition](#)
- [bibus » memo](#)
- [bibus » migrationTask](#)
- [bibus » planningMacroTask](#)
- [bibus » planningTask](#)

- [bibus » storedProcedureTask](#)
- [bibus » webServiceTask](#)

Properties

This class has the following properties.

actualCompletionTime

Specifies the time that the task completed, or the time the task was purged or canceled, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
- is encoded as type `tns:dateTimeProp`

actualExecutionTime

Specifies the actual start time of the task, or the time the task was purged, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
- is encoded as type `tns:dateTimeProp`

details

Contains additional information about the running of the task. The type of information varies based on the type of task.

dispatcherID

Identifies the dispatcher that handled the run task request.

SDK applications should not change this property.

This property

- is of type [bibus » guid](#)
- is encoded as type `tns:guidProp`
- is searchable

New in Version 8.3 — “Schedule Management” on page 1907

This property was added.

eventID

Specifies the unique ID for the task that created this history object.

This property

- is of type `string`
- is encoded as type `tns:stringProp`
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase
- is searchable

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`
is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable

New in Version 8.4 — “[On Demand Refresh of Prompt Cache](#)” on page 1884

This property was added.

maximumDetailSeverity

Specifies the maximum severity of all contained [bibus » baseHistoryDetail](#) objects. SDK applications should not change this property.

This property

- is of type [bibus » severityEnum](#)
is encoded as type `tns:severityEnumProp`

New in Version 8.3 — “[Schedule Management](#)” on page 1907

This property was added.

ownerEventID

Specifies the unique ID of the originating task, when that task, such as an agent or job, launches other tasks during its run.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase
- is searchable

requestedExecutionTime

Specifies the requested start time for the task, in Coordinated Universal Time (UTC). The task may have started later than the requested run time.

This property

- is of type `dateTime`
is encoded as type `tns:dateTimeProp`
- is searchable

restartEventID

If this object was created as a result of retrying a failed task, this property specifies the [eventID](#) of the failed run that was restarted.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase
- is searchable

New in Version 8.3 — “[Task Retry](#)” on page 1909

This property was added.

scheduleTriggerName

Specifies the value from [bibus » schedule » triggerName](#) property when the task was scheduled.

If the task was not run as a scheduled task or the value of [bibus » schedule » type](#) property was not [trigger](#) then this property will be NIL.

SDK applications should not change this property.

This property

- is of type `string`
is encoded as type `tns:stringProp`

New in Version 8.3 — “Schedule Management” on page 1907

This property was added.

scheduleType

Specifies the value from [bibus » schedule » type](#) property when the task was scheduled.

If the task was not a scheduled task, this property will be NIL.

SDK applications should not change this property.

This property

- is of type [bibus » scheduleTypeEnum](#)
is encoded as type `tns:nmtokenProp`

New in Version 8.3 — “Schedule Management” on page 1907

This property was added.

status

Specifies the current run status of a task.

This property

- is of type [bibus » runStatusEnum](#)
is encoded as type `tns:nmtokenProp`
- must contain no more than 9 characters
- contains a value that can be counted
- is searchable

user

Contains information about the account used to run the task.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

historyDetail

Provides a message generated during the running of a task.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

Container Information

Contained by instances of the following classes

- [bibus » historyDetailDataMovementService](#)

historyDetailAgentService

Provides the location of the [bibus » agentOutputHotList](#).

Instances of this class are created by the [agentService](#) during the running of an agent.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

Properties

This class has the following properties.

output

Contains a reference to the [bibus » agentOutputHotList](#) object produced during the run.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » agentOutputHotList](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

historyDetailDataMovementService

Provides information related to the execution of a data movement task.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

Container Information

Contains instances of the following classes

- [bibus » historyDetail](#)
- [bibus » historyDetailDataMovementService](#)

Contained by instances of the following classes

- [bibus » historyDetailDataMovementService](#)

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
 - is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property `data`. The media type is expressed as a `type/subtype` pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

details

Contains data movement details for sub-tasks.

historyDetailDeploymentSummary

Provides summary information related to deployments.

This includes the number of objects added, deleted, replaced, and updated during a deployment. Instances of this class are created by the [contentManagerService](#).

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

What's new

New in Version 8.3 – [“Deployment History” on page 1915](#)

This class was added.

Properties

This class has the following properties.

addedObjectCount

Specifies the number of objects added to the archive or content store during the deployment.

This property

- is of type `int`
is encoded as type `tns:intProp`

deletedObjectCount

Specifies the number of objects deleted from the content store during the deployment.

This property

- is of type `int`
is encoded as type `tns:intProp`

replacedObjectCount

Specifies the number of objects replaced in the content store during the deployment.

This property

- is of type `int`
is encoded as type `tns:intProp`

updatedObjectCount

Specifies the number of objects updated in the content store during the deployment.

This property

- is of type `int`
is encoded as type `tns:intProp`

historyDetailMigrationService

Defines information related to the execution of a migration task.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus](#) » [baseHistoryDetail](#) class

What's new

New in Version 8.4 – [“Migration Service” on page 1883](#)

This class was added.

Properties

This class has the following properties.

hasMessage

Specifies that the message property contains a message.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only
- is searchable

message

Contains the messages related to the migrated object.

This property

- is an array of type bibus » [faultDetail](#)
 - is encoded as type `tns:faultDetailArrayProp`

migratedObject

Refers to the migrated object.

This property

- is an array of type bibus » [baseClass](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using a name- and type-based search path
- must have at most 1 item

historyDetailRelatedHistory

Provides an association between the bibus » [history](#) instance of an invoked task with the bibus » [history](#) of the parent task.

Tasks such as agents and jobs invoke other tasks, such as reports or Web service methods. When each task is run, the `monitorService` creates a bibus » [history](#) object to record the execution details for that task. For example, if an agent runs a report, the history for the agent will contain an instance of this class. This instance refers to the history for the report.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the bibus » [baseHistoryDetail](#) class

Properties

This class has the following properties.

related

Refers to other bibus » [history](#) instances created during the running of a task.

This property

- is an array of type bibus » [baseClass](#)
 - has items that must be of class bibus » [history](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

historyDetailRelatedReports

Provides the information required to run a report that supplies the history for the task.

This class is created in the [bibus](#) » [history](#) object of tasks that record details about the run in a relational database.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus](#) » [baseHistoryDetail](#) class

Properties

This class has the following properties.

linkPaths

Contains a reference to the associated [bibus](#) » [report](#) that can be run to obtain details about the running of a task.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [report](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

parameters

Contains the set of parameter values to be used with objects referenced by the [linkPaths](#) property.

This property

- is an array of type [bibus](#) » [parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

historyDetailReportService

Provides the location of the report output created in the content store during the running of a report.

Instances of this class are created by the [reportService](#) and [batchReportService](#).

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus](#) » [baseHistoryDetail](#) class

Properties

This class has the following properties.

output

Contains a reference to the [bibus](#) » [reportVersion](#) object produced during the run.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [reportVersion](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path

historyDetailRequestArguments

Provides the options and parameter values used to run the task.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseHistoryDetail](#) class

Properties

This class has the following properties.

options

Contains the set of options specified for the execution of the task.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values specified for the execution of the task.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

humanTask

Defines the class that describes a task that requires human interaction.

A task is defined as a notification, request, or other user-defined action, related to an issue.

Tasks can be assigned to users manually by another user, or automatically. Automatic tasks are generated as a result of certain conditions being met when a report is run.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the humanTask class.

Action	Mode	Service	Method
Run	All	agentService	<code>asynch » run(objectPath, parameterValues, options)</code>

Container Information

Contains instances of the following classes

- [bibus » history](#)

Contained by instances of the following classes

- [bibus » authoredAgentDefinition](#)

What's new

New in Version 10.1.0 – [“Human Task” on page 1862](#)

This class was added.

Properties

This class has the following properties.

bindingName

Specifies the binding that defines the operation. The binding provides protocol-specific information for an operation.

This property

- is of type `string`

is encoded as type `tns:stringProp`

bulkEvents

Specifies that multiple events can be passed to the human task service in a single call.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

inputMessageName

Specifies the name of the input message for the operation.

This property

- is of type `string`

is encoded as type `tns:stringProp`

operationName

Specifies the name of the human task service operation.

This property

- is of type `string`
is encoded as type `tns:stringProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type `bibus » option`
is encoded as type `tns:optionArrayProp`

outputMessageName

Specifies the name of the output message for the operation.

This property

- is of type `string`
is encoded as type `tns:stringProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type `bibus » parameterValue`
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

<i>Table 168. Rules for a new humanTask object</i>			
Versioned Class	Maximum Age	Maximum Count	Ordered By
<code>bibus » history class</code>		5	<code>bibus » baseClass » creationTime property</code>

This property

- is an array of type `bibus » retentionRule`
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

serviceName

Specifies the name of the service.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

uri

Specifies the URI of the human task service to call.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`

humanTaskService

Defines run-time configuration parameters for the human task service.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contains instances of the following classes

- `bibus » runTimeState`

- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Human Task Service” on page 1854](#)

This class was added.

Related information:

- [“Human task service” on page 8](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

htsAuditLevel

Specifies the auditing level for the human task service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.1.0 – [“Human Task Service” on page 1854](#)

This property was added.

htsCompletedTaskLifetime

Specifies the lifetime of completed human tasks in XML Schema 1.0 type `xs:duration` form. If the lifetime is set to `P90D`, the human task will be deleted after 90 days if all linked reports or dashboards are deleted.

The default lifetime is ninety days.

This property

- is of type `duration`
 - is encoded as type `tns:durationProp`
- has a default value of `P90D`
- can be acquired from a containing object

New in Version 10.1.0 – “Human Task Service” on page 1854

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

identity

Represents the user identities. It specifies the account and the set of groups, roles, and namespaces associated with the user session. For Multi-tenant install, it also includes the user's tenancy.

What's new

New in Version 10.2.2 – Full tenant impersonation capability for system administrators

This class was added.

Properties

This class has the following properties.

self

Identifies the account and the set of groups, roles, and namespaces associated with the user session.

This property

- is an array of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`
- is read-only

tenancy

Identifies the user's tenancy for the current session such as the true tenantID, readTenantIDs and the writeTenantID

This property

- is of type [bibus](#) » [tenancy](#)
is encoded as type `tns:tenancy`
- is read-only

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.2.0 – [“Support for interactive discovery and visualization” on page 1839](#)

This class was added.

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

idVizAuditLevel

Specifies the auditing level for the idViz service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.2.0 – [“Support for interactive discovery and visualization” on page 1839](#)

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

importDeployment

Defines the specification for a particular import deployment.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the importDeployment class.

Action	Mode	Service	Method
Run	All	contentManagerService	async » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)

Properties

This class has the following properties.

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		1	bibus » baseClass » creationTime property

This property

- is an array of type [bibus](#) » [retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the [content](#) » [determineRouting\(objectPaths\)](#) method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

indexDataService

Defines run-time configuration parameters for the index data service .

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

idsAuditLevel

Specifies the auditing level for the index data service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

idsConnections

Specifies the maximum number of connections that a process of the index data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

idsPeakConnections

Specifies the number of connections that an index data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

indexOption

Defines the abstract base class for all index option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus](#) » [option](#) class

Derived Classes

- [bibus](#) » [indexOptionBoolean](#)
- [bibus](#) » [indexOptionInt](#)
- [bibus](#) » [indexOptionSearchPathMultipleObjectArray](#)

What's new

New in Version 8.3 – “Search – For Internal Use Only” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

name

Identifies the index option.

This property

- is of type [bibus](#) » [indexOptionEnum](#)
is encoded as type `tns:indexOptionEnum`

indexOptionBoolean

Defines boolean values for the index options.

This class

- inherits properties from the [bibus » indexOption](#) class

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

value

Specifies the value for the index option.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

indexOptionInt

Defines integer values for the index options.

This class

- inherits properties from the [bibus » indexOption](#) class

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

value

Specifies the value for the index option.

This property

- is of type `int`

is encoded as type `xs:int`

indexOptionSearchPathMultipleObjectArray

Defines [bibus » searchPathMultipleObject](#) values for the index options.

This class

- inherits properties from the [bibus » indexOption](#) class

What's new

New in Version 8.4 – “Index Options” on page 1893

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the index option.

This property

- is an array of type [bibus » searchPathMultipleObject](#)
is encoded as type `tns:searchPathMultipleObjectArray`

indexSearchService

Defines run-time configuration parameters for the [indexSearchService](#).

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

issAuditLevel

Specifies the auditing level for the index search service.

This property

- is of type [bibus » auditLevelEnum](#)

is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

issConnections

Specifies the maximum number of connections that a process of the index search service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

issPeakConnections

Specifies the number of connections that a index search service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

indexSearchServiceSpecification

Reserved.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the indexSearchServiceSpecification class.

Action	Mode	Service	Method
Run	All	indexSearchService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.3 — “Search — For Internal Use Only” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

indexTerm

Reserved.

References

Used by the following properties:

- [bibus » asynchDetailIndexTerms » childTerms](#)
- [bibus » asynchDetailIndexTerms » exampleTerms](#)
- [bibus » asynchDetailIndexTerms » parentTerms](#)
- [bibus » asynchDetailIndexTerms » siblingTerms](#)
- [bibus » asynchDetailIndexData » terms](#)
- [bibus » indexTermOptionIndexTermArray » value](#)

Used by the following method parameters:

- [indexTerm » addTermAssociation\(term, parameterValues, options\) » term](#)
- [indexTerm » deleteTermAssociation\(term, parameterValues, options\) » term](#)

What's new

New in Version 8.3 — “Search — For Internal Use Only” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

value

This property

- is an array of type [bibus » multilingualString](#)
is encoded as type `tns:multilingualStringArray`

indexTermOption

Defines the abstract base class for all index term option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » indexTermOptionBoolean](#)
- [bibus » indexTermOptionIndexTermArray](#)

What's new

New in Version 8.3 — [“Search — For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

name

Identifies the index term option.

This property

- is of type [bibus » indexTermOptionEnum](#)
is encoded as type `tns:indexTermOptionEnum`

indexTermOptionBoolean

Defines boolean values for the index term options.

This class

- inherits properties from the [bibus » indexTermOption](#) class

What's new

New in Version 8.3 — [“Search — For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

value

Specifies the value for the index term option.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

indexTermOptionIndexTermArray

Defines [bibus » indexTerm](#) class values for the index term options.

This class

- inherits properties from the [bibus » indexTermOption](#) class

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

value

Specifies the values for the index term option.

This property

- is an array of type [bibus » indexTerm](#)

is encoded as type `tns:indexTermArray`

indexUpdateService

Defines run-time configuration parameters for the [indexUpdateService](#).

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

iusAuditLevel

Specifies the auditing level for the index update service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

iusConnections

Specifies the maximum number of connections that a process of the index update service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

iusPeakConnections

Specifies the number of connections that an index update service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

indexUpdateServiceSpecification

Reserved.

This class

- inherits properties from the [bibus](#) » [asynchSpecification](#) class

References

Used by the following properties:

- [bibus](#) » [indexUpdateTask](#) » [specification](#)
- [bibus](#) » [indexUpdateServiceSpecificationProp](#) » [value](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `indexUpdateServiceSpecification` class.

Action	Mode	Service	Method
Run	All	indexUpdateService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.3 – “Search – For Internal Use Only” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

indexUpdateServiceSpecificationProp

Defines the simple property class for the [bibus » indexUpdateServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » indexUpdateTask » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » indexUpdateServiceSpecification](#)
is encoded as type `tns:indexUpdateServiceSpecification`

indexUpdateTask

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `indexUpdateTask` class.

Action	Mode	Service	Method
Run	All	indexUpdateService	<code>asynch » run(objectPath, parameterValues, options)</code>

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)

What's new

New in Version 8.3 — [“Search — For Internal Use Only” on page 1920](#)

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

specification

This property

- is of type [bibus](#) » [indexUpdateServiceSpecification](#)

is encoded as type `tns:indexUpdateServiceSpecificationProp`

installedComponent

Contains information about an installed IBM Cognos Analytics component.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

Properties

This class has the following properties.

componentID

Identifies the component.

This property

- is of type [bibus](#) » [installedComponentEnum](#)

is encoded as type `tns:installedComponentEnumProp`

- is searchable

installedComponentEnumProp

Defines the simple property class for the [bibus](#) » [installedComponentEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [installedComponent](#) » [componentID](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [installedComponentEnum](#)
is encoded as type `tns:installedComponentEnum`

interactiveReport

Contains information that defines an IBM Cognos Active Report specification.

IBM Cognos Analytics - Reporting usually creates instances of this class.

An IBM Cognos Active Report allows users to work offline with a local copy of their data. They can explore and analyze data, make changes to their reports, and filter and sort data while disconnected from the network. When they re-connect, they can synchronize their changes with the live data source.

The [reportService](#) service and [batchReportService](#) service identify an IBM Cognos Active Report type by the object type specified in a request. If the report is persisted in the content store, the object type is [bibus](#) » [interactiveReport](#) class and the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method initiates the request. If the report is not persisted, the object type is [bibus](#) » [reportServiceInteractiveReportSpecification](#) class and the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method initiates the request.

An IBM Cognos Active Report is identified in the report specification by the attribute value of `application="true"` for the root `report` element. For more information about report specifications, see [Chapter 23, "Using report specifications,"](#) on page 1457.

See the *IBM Cognos Analytics - Reporting User Guide* for more information.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [authoredReport](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `interactiveReport` class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

What's new

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This class was added.

intProp

Defines the simple property class for the `int`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » adaptiveAnalyticsService » aasAffineConnections](#)
- [bibus » configuration » aasAffineConnections](#)
- [bibus » configurationFolder » aasAffineConnections](#)
- [bibus » dispatcher » aasAffineConnections](#)
- [bibus » adaptiveAnalyticsService » aasExecutionTimeLimit](#)
- [bibus » configuration » aasExecutionTimeLimit](#)
- [bibus » configurationFolder » aasExecutionTimeLimit](#)
- [bibus » dispatcher » aasExecutionTimeLimit](#)
- [bibus » adaptiveAnalyticsService » aasMaximumProcesses](#)
- [bibus » configuration » aasMaximumProcesses](#)
- [bibus » configurationFolder » aasMaximumProcesses](#)
- [bibus » dispatcher » aasMaximumProcesses](#)
- [bibus » adaptiveAnalyticsService » aasNonAffineConnections](#)
- [bibus » configuration » aasNonAffineConnections](#)
- [bibus » configurationFolder » aasNonAffineConnections](#)
- [bibus » dispatcher » aasNonAffineConnections](#)
- [bibus » adaptiveAnalyticsService » aasPeakAffineConnections](#)
- [bibus » configuration » aasPeakAffineConnections](#)
- [bibus » configurationFolder » aasPeakAffineConnections](#)
- [bibus » dispatcher » aasPeakAffineConnections](#)
- [bibus » adaptiveAnalyticsService » aasPeakMaximumProcesses](#)
- [bibus » configuration » aasPeakMaximumProcesses](#)
- [bibus » configurationFolder » aasPeakMaximumProcesses](#)
- [bibus » dispatcher » aasPeakMaximumProcesses](#)
- [bibus » adaptiveAnalyticsService » aasPeakNonAffineConnections](#)
- [bibus » configuration » aasPeakNonAffineConnections](#)
- [bibus » configurationFolder » aasPeakNonAffineConnections](#)
- [bibus » dispatcher » aasPeakNonAffineConnections](#)
- [bibus » historyDetailDeploymentSummary » addedObjectCount](#)
- [bibus » agentService » asConnections](#)
- [bibus » configuration » asConnections](#)
- [bibus » configurationFolder » asConnections](#)

- [bibus » dispatcher » asConnections](#)
- [bibus » agentService » asMaximumEMailAttachmentSize](#)
- [bibus » configuration » asMaximumEMailAttachmentSize](#)
- [bibus » configurationFolder » asMaximumEMailAttachmentSize](#)
- [bibus » dispatcher » asMaximumEMailAttachmentSize](#)
- [bibus » agentService » asPeakConnections](#)
- [bibus » configuration » asPeakConnections](#)
- [bibus » configurationFolder » asPeakConnections](#)
- [bibus » dispatcher » asPeakConnections](#)
- [bibus » batchReportService » brsAffineConnections](#)
- [bibus » configuration » brsAffineConnections](#)
- [bibus » configurationFolder » brsAffineConnections](#)
- [bibus » dispatcher » brsAffineConnections](#)
- [bibus » batchReportService » brsChartHotspotLimit](#)
- [bibus » configuration » brsChartHotspotLimit](#)
- [bibus » configurationFolder » brsChartHotspotLimit](#)
- [bibus » dispatcher » brsChartHotspotLimit](#)
- [bibus » batchReportService » brsExecutionTimeLimit](#)
- [bibus » configuration » brsExecutionTimeLimit](#)
- [bibus » configurationFolder » brsExecutionTimeLimit](#)
- [bibus » dispatcher » brsExecutionTimeLimit](#)
- [bibus » batchReportService » brsMaximumEMailAttachmentSize](#)
- [bibus » configuration » brsMaximumEMailAttachmentSize](#)
- [bibus » configurationFolder » brsMaximumEMailAttachmentSize](#)
- [bibus » dispatcher » brsMaximumEMailAttachmentSize](#)
- [bibus » batchReportService » brsNonAffineConnections](#)
- [bibus » configuration » brsNonAffineConnections](#)
- [bibus » configurationFolder » brsNonAffineConnections](#)
- [bibus » dispatcher » brsNonAffineConnections](#)
- [bibus » batchReportService » brsPDFCompressionLevel](#)
- [bibus » configuration » brsPDFCompressionLevel](#)
- [bibus » configurationFolder » brsPDFCompressionLevel](#)
- [bibus » dispatcher » brsPDFCompressionLevel](#)
- [bibus » batchReportService » brsPeakAffineConnections](#)
- [bibus » configuration » brsPeakAffineConnections](#)
- [bibus » configurationFolder » brsPeakAffineConnections](#)
- [bibus » dispatcher » brsPeakAffineConnections](#)
- [bibus » batchReportService » brsPeakMaximumProcesses](#)
- [bibus » configuration » brsPeakMaximumProcesses](#)
- [bibus » configurationFolder » brsPeakMaximumProcesses](#)
- [bibus » dispatcher » brsPeakMaximumProcesses](#)
- [bibus » batchReportService » brsPeakNonAffineConnections](#)
- [bibus » configuration » brsPeakNonAffineConnections](#)

- [bibus » configurationFolder » brsPeakNonAffineConnections](#)
- [bibus » dispatcher » brsPeakNonAffineConnections](#)
- [bibus » configuration » cmcsHeapLimit](#)
- [bibus » configurationFolder » cmcsHeapLimit](#)
- [bibus » contentManagerCacheService » cmcsHeapLimit](#)
- [bibus » dispatcher » cmcsHeapLimit](#)
- [bibus » configuration » cmsConnections](#)
- [bibus » configurationFolder » cmsConnections](#)
- [bibus » contentManagerService » cmsConnections](#)
- [bibus » dispatcher » cmsConnections](#)
- [bibus » configuration » cmsPeakConnections](#)
- [bibus » configurationFolder » cmsPeakConnections](#)
- [bibus » contentManagerService » cmsPeakConnections](#)
- [bibus » dispatcher » cmsPeakConnections](#)
- [bibus » documentContent » contextBlockCount](#)
- [bibus » output » contextBlockCount](#)
- [bibus » packageConfiguration » crosstabItemDisplayCountDefault](#)
- [bibus » packageConfiguration » crosstabItemDisplayCountLimit](#)
- [bibus » historyDetailDeploymentSummary » deletedObjectCount](#)
- [bibus » configuration » dimsAffineConnections](#)
- [bibus » configurationFolder » dimsAffineConnections](#)
- [bibus » dimensionManagementService » dimsAffineConnections](#)
- [bibus » dispatcher » dimsAffineConnections](#)
- [bibus » configuration » dimsExecutionTimeLimit](#)
- [bibus » configurationFolder » dimsExecutionTimeLimit](#)
- [bibus » dimensionManagementService » dimsExecutionTimeLimit](#)
- [bibus » dispatcher » dimsExecutionTimeLimit](#)
- [bibus » configuration » dimsMaximumProcesses](#)
- [bibus » configurationFolder » dimsMaximumProcesses](#)
- [bibus » dimensionManagementService » dimsMaximumProcesses](#)
- [bibus » dispatcher » dimsMaximumProcesses](#)
- [bibus » configuration » dimsNonAffineConnections](#)
- [bibus » configurationFolder » dimsNonAffineConnections](#)
- [bibus » dimensionManagementService » dimsNonAffineConnections](#)
- [bibus » dispatcher » dimsNonAffineConnections](#)
- [bibus » configuration » dimsPeakAffineConnections](#)
- [bibus » configurationFolder » dimsPeakAffineConnections](#)
- [bibus » dimensionManagementService » dimsPeakAffineConnections](#)
- [bibus » dispatcher » dimsPeakAffineConnections](#)
- [bibus » configuration » dimsPeakMaximumProcesses](#)
- [bibus » configurationFolder » dimsPeakMaximumProcesses](#)
- [bibus » dimensionManagementService » dimsPeakMaximumProcesses](#)
- [bibus » dispatcher » dimsPeakMaximumProcesses](#)

- [bibus » configuration » dimsPeakNonAffineConnections](#)
- [bibus » configurationFolder » dimsPeakNonAffineConnections](#)
- [bibus » dimensionManagementService » dimsPeakNonAffineConnections](#)
- [bibus » dispatcher » dimsPeakNonAffineConnections](#)
- [bibus » configuration » dimsQueueLimit](#)
- [bibus » configurationFolder » dimsQueueLimit](#)
- [bibus » dimensionManagementService » dimsQueueLimit](#)
- [bibus » dispatcher » dimsQueueLimit](#)
- [bibus » configuration » disConnections](#)
- [bibus » configurationFolder » disConnections](#)
- [bibus » dataIntegrationService » disConnections](#)
- [bibus » dispatcher » disConnections](#)
- [bibus » configuration » disPeakConnections](#)
- [bibus » configurationFolder » disPeakConnections](#)
- [bibus » dataIntegrationService » disPeakConnections](#)
- [bibus » dispatcher » disPeakConnections](#)
- [bibus » configuration » dmsAffineConnections](#)
- [bibus » configurationFolder » dmsAffineConnections](#)
- [bibus » dataMovementService » dmsAffineConnections](#)
- [bibus » dispatcher » dmsAffineConnections](#)
- [bibus » configuration » dmsConnections](#)
- [bibus » configurationFolder » dmsConnections](#)
- [bibus » dataMovementService » dmsConnections](#)
- [bibus » dispatcher » dmsConnections](#)
- [bibus » configuration » dmsExecutionTimeLimit](#)
- [bibus » configurationFolder » dmsExecutionTimeLimit](#)
- [bibus » dataMovementService » dmsExecutionTimeLimit](#)
- [bibus » dispatcher » dmsExecutionTimeLimit](#)
- [bibus » configuration » dmsMaximumProcesses](#)
- [bibus » configurationFolder » dmsMaximumProcesses](#)
- [bibus » dataMovementService » dmsMaximumProcesses](#)
- [bibus » dispatcher » dmsMaximumProcesses](#)
- [bibus » configuration » dmsNonAffineConnections](#)
- [bibus » configurationFolder » dmsNonAffineConnections](#)
- [bibus » dataMovementService » dmsNonAffineConnections](#)
- [bibus » dispatcher » dmsNonAffineConnections](#)
- [bibus » configuration » dmsPeakAffineConnections](#)
- [bibus » configurationFolder » dmsPeakAffineConnections](#)
- [bibus » dataMovementService » dmsPeakAffineConnections](#)
- [bibus » dispatcher » dmsPeakAffineConnections](#)
- [bibus » configuration » dmsPeakMaximumProcesses](#)
- [bibus » configurationFolder » dmsPeakMaximumProcesses](#)
- [bibus » dataMovementService » dmsPeakMaximumProcesses](#)

- [bibus » dispatcher » dmsPeakMaximumProcesses](#)
- [bibus » configuration » dmsPeakNonAffineConnections](#)
- [bibus » configurationFolder » dmsPeakNonAffineConnections](#)
- [bibus » dataMovementService » dmsPeakNonAffineConnections](#)
- [bibus » dispatcher » dmsPeakNonAffineConnections](#)
- [bibus » configuration » dmsQueueLimit](#)
- [bibus » configurationFolder » dmsQueueLimit](#)
- [bibus » dataMovementService » dmsQueueLimit](#)
- [bibus » dispatcher » dmsQueueLimit](#)
- [bibus » configuration » dsCompressAttachmentLimit](#)
- [bibus » configurationFolder » dsCompressAttachmentLimit](#)
- [bibus » deliveryService » dsCompressAttachmentLimit](#)
- [bibus » dispatcher » dsCompressAttachmentLimit](#)
- [bibus » configuration » dsConnections](#)
- [bibus » configurationFolder » dsConnections](#)
- [bibus » deliveryService » dsConnections](#)
- [bibus » dispatcher » dsConnections](#)
- [bibus » configuration » dsMaximumEMailSize](#)
- [bibus » configurationFolder » dsMaximumEMailSize](#)
- [bibus » deliveryService » dsMaximumEMailSize](#)
- [bibus » dispatcher » dsMaximumEMailSize](#)
- [bibus » configuration » dsPeakConnections](#)
- [bibus » configurationFolder » dsPeakConnections](#)
- [bibus » deliveryService » dsPeakConnections](#)
- [bibus » dispatcher » dsPeakConnections](#)
- [bibus » configuration » gsExecutionTimeLimit](#)
- [bibus » configurationFolder » gsExecutionTimeLimit](#)
- [bibus » dispatcher » gsExecutionTimeLimit](#)
- [bibus » graphicsService » gsExecutionTimeLimit](#)
- [bibus » configuration » gsNonAffineConnections](#)
- [bibus » configurationFolder » gsNonAffineConnections](#)
- [bibus » dispatcher » gsNonAffineConnections](#)
- [bibus » graphicsService » gsNonAffineConnections](#)
- [bibus » configuration » gsPeakNonAffineConnections](#)
- [bibus » configurationFolder » gsPeakNonAffineConnections](#)
- [bibus » dispatcher » gsPeakNonAffineConnections](#)
- [bibus » graphicsService » gsPeakNonAffineConnections](#)
- [bibus » configuration » gsQueueLimit](#)
- [bibus » configurationFolder » gsQueueLimit](#)
- [bibus » dispatcher » gsQueueLimit](#)
- [bibus » graphicsService » gsQueueLimit](#)
- [bibus » account » horizontalElementsRenderingLimit](#)
- [bibus » contact » horizontalElementsRenderingLimit](#)

- [bibus » configuration » idsConnections](#)
- [bibus » configurationFolder » idsConnections](#)
- [bibus » dispatcher » idsConnections](#)
- [bibus » indexDataService » idsConnections](#)
- [bibus » configuration » idsPeakConnections](#)
- [bibus » configurationFolder » idsPeakConnections](#)
- [bibus » dispatcher » idsPeakConnections](#)
- [bibus » indexDataService » idsPeakConnections](#)
- [bibus » configuration » issConnections](#)
- [bibus » configurationFolder » issConnections](#)
- [bibus » dispatcher » issConnections](#)
- [bibus » indexSearchService » issConnections](#)
- [bibus » configuration » issPeakConnections](#)
- [bibus » configurationFolder » issPeakConnections](#)
- [bibus » dispatcher » issPeakConnections](#)
- [bibus » indexSearchService » issPeakConnections](#)
- [bibus » configuration » iusConnections](#)
- [bibus » configurationFolder » iusConnections](#)
- [bibus » dispatcher » iusConnections](#)
- [bibus » indexUpdateService » iusConnections](#)
- [bibus » configuration » iusPeakConnections](#)
- [bibus » configurationFolder » iusPeakConnections](#)
- [bibus » dispatcher » iusPeakConnections](#)
- [bibus » indexUpdateService » iusPeakConnections](#)
- [bibus » configuration » jsConnections](#)
- [bibus » configurationFolder » jsConnections](#)
- [bibus » dispatcher » jsConnections](#)
- [bibus » jobService » jsConnections](#)
- [bibus » configuration » jsPeakConnections](#)
- [bibus » configurationFolder » jsPeakConnections](#)
- [bibus » dispatcher » jsPeakConnections](#)
- [bibus » jobService » jsPeakConnections](#)
- [bibus » configuration » mbsConnections](#)
- [bibus » configurationFolder » mbsConnections](#)
- [bibus » dispatcher » mbsConnections](#)
- [bibus » mobileService » mbsConnections](#)
- [bibus » configuration » mbsPeakConnections](#)
- [bibus » configurationFolder » mbsPeakConnections](#)
- [bibus » dispatcher » mbsPeakConnections](#)
- [bibus » mobileService » mbsPeakConnections](#)
- [bibus » configuration » mdsAffineConnections](#)
- [bibus » configurationFolder » mdsAffineConnections](#)
- [bibus » dispatcher » mdsAffineConnections](#)

- [bibus » metadataService » mdsAffineConnections](#)
- [bibus » configuration » mdsExecutionTimeLimit](#)
- [bibus » configurationFolder » mdsExecutionTimeLimit](#)
- [bibus » dispatcher » mdsExecutionTimeLimit](#)
- [bibus » metadataService » mdsExecutionTimeLimit](#)
- [bibus » configuration » mdsMaximumProcesses](#)
- [bibus » configurationFolder » mdsMaximumProcesses](#)
- [bibus » dispatcher » mdsMaximumProcesses](#)
- [bibus » metadataService » mdsMaximumProcesses](#)
- [bibus » configuration » mdsNonAffineConnections](#)
- [bibus » configurationFolder » mdsNonAffineConnections](#)
- [bibus » dispatcher » mdsNonAffineConnections](#)
- [bibus » metadataService » mdsNonAffineConnections](#)
- [bibus » configuration » mdsPeakAffineConnections](#)
- [bibus » configurationFolder » mdsPeakAffineConnections](#)
- [bibus » dispatcher » mdsPeakAffineConnections](#)
- [bibus » metadataService » mdsPeakAffineConnections](#)
- [bibus » configuration » mdsPeakMaximumProcesses](#)
- [bibus » configurationFolder » mdsPeakMaximumProcesses](#)
- [bibus » dispatcher » mdsPeakMaximumProcesses](#)
- [bibus » metadataService » mdsPeakMaximumProcesses](#)
- [bibus » configuration » mdsPeakNonAffineConnections](#)
- [bibus » configurationFolder » mdsPeakNonAffineConnections](#)
- [bibus » dispatcher » mdsPeakNonAffineConnections](#)
- [bibus » metadataService » mdsPeakNonAffineConnections](#)
- [bibus » configuration » mdsQueueLimit](#)
- [bibus » configurationFolder » mdsQueueLimit](#)
- [bibus » dispatcher » mdsQueueLimit](#)
- [bibus » metadataService » mdsQueueLimit](#)
- [bibus » packageConfiguration » memberDisplayCountDefault](#)
- [bibus » packageConfiguration » memberDisplayCountLimit](#)
- [bibus » configuration » misConnections](#)
- [bibus » configurationFolder » misConnections](#)
- [bibus » dispatcher » misConnections](#)
- [bibus » migrationService » misConnections](#)
- [bibus » configuration » misPeakConnections](#)
- [bibus » configurationFolder » misPeakConnections](#)
- [bibus » dispatcher » misPeakConnections](#)
- [bibus » migrationService » misPeakConnections](#)
- [bibus » configuration » mmsConnections](#)
- [bibus » configurationFolder » mmsConnections](#)
- [bibus » dispatcher » mmsConnections](#)
- [bibus » metricsManagerService » mmsConnections](#)

- [bibus » configuration » mmsPeakConnections](#)
- [bibus » configurationFolder » mmsPeakConnections](#)
- [bibus » dispatcher » mmsPeakConnections](#)
- [bibus » metricsManagerService » mmsPeakConnections](#)
- [bibus » configuration » nonPeakDemandBeginHour](#)
- [bibus » configurationFolder » nonPeakDemandBeginHour](#)
- [bibus » dispatcher » nonPeakDemandBeginHour](#)
- [bibus » configuration » pacsConnections](#)
- [bibus » configurationFolder » pacsConnections](#)
- [bibus » dispatcher » pacsConnections](#)
- [bibus » planningAdministrationConsoleService » pacsConnections](#)
- [bibus » configuration » pacsPeakConnections](#)
- [bibus » configurationFolder » pacsPeakConnections](#)
- [bibus » dispatcher » pacsPeakConnections](#)
- [bibus » planningAdministrationConsoleService » pacsPeakConnections](#)
- [bibus » configuration » pdsConnections](#)
- [bibus » configurationFolder » pdsConnections](#)
- [bibus » dispatcher » pdsConnections](#)
- [bibus » planningDataService » pdsConnections](#)
- [bibus » configuration » pdsEListAccessCacheLimit](#)
- [bibus » configurationFolder » pdsEListAccessCacheLimit](#)
- [bibus » dispatcher » pdsEListAccessCacheLimit](#)
- [bibus » planningDataService » pdsEListAccessCacheLimit](#)
- [bibus » configuration » pdsMaximumProcesses](#)
- [bibus » configurationFolder » pdsMaximumProcesses](#)
- [bibus » dispatcher » pdsMaximumProcesses](#)
- [bibus » planningDataService » pdsMaximumProcesses](#)
- [bibus » configuration » pdsPeakConnections](#)
- [bibus » configurationFolder » pdsPeakConnections](#)
- [bibus » dispatcher » pdsPeakConnections](#)
- [bibus » planningDataService » pdsPeakConnections](#)
- [bibus » configuration » pdsPeakMaximumProcesses](#)
- [bibus » configurationFolder » pdsPeakMaximumProcesses](#)
- [bibus » dispatcher » pdsPeakMaximumProcesses](#)
- [bibus » planningDataService » pdsPeakMaximumProcesses](#)
- [bibus » configuration » peakDemandBeginHour](#)
- [bibus » configurationFolder » peakDemandBeginHour](#)
- [bibus » dispatcher » peakDemandBeginHour](#)
- [bibus » configuration » periodicalDocumentVersionRetentionCount](#)
- [bibus » configuration » ppsAffineConnections](#)
- [bibus » configurationFolder » ppsAffineConnections](#)
- [bibus » dispatcher » ppsAffineConnections](#)
- [bibus » powerPlayService » ppsAffineConnections](#)

- [bibus » configuration » ppsExecutionTimeLimit](#)
- [bibus » configurationFolder » ppsExecutionTimeLimit](#)
- [bibus » dispatcher » ppsExecutionTimeLimit](#)
- [bibus » powerPlayService » ppsExecutionTimeLimit](#)
- [bibus » configuration » ppsMaximumEMailAttachmentSize](#)
- [bibus » configurationFolder » ppsMaximumEMailAttachmentSize](#)
- [bibus » dispatcher » ppsMaximumEMailAttachmentSize](#)
- [bibus » powerPlayService » ppsMaximumEMailAttachmentSize](#)
- [bibus » configuration » ppsNonAffineConnections](#)
- [bibus » configurationFolder » ppsNonAffineConnections](#)
- [bibus » dispatcher » ppsNonAffineConnections](#)
- [bibus » powerPlayService » ppsNonAffineConnections](#)
- [bibus » configuration » ppsPeakAffineConnections](#)
- [bibus » configurationFolder » ppsPeakAffineConnections](#)
- [bibus » dispatcher » ppsPeakAffineConnections](#)
- [bibus » powerPlayService » ppsPeakAffineConnections](#)
- [bibus » configuration » ppsPeakNonAffineConnections](#)
- [bibus » configurationFolder » ppsPeakNonAffineConnections](#)
- [bibus » dispatcher » ppsPeakNonAffineConnections](#)
- [bibus » powerPlayService » ppsPeakNonAffineConnections](#)
- [bibus » configuration » ppsQueueLimit](#)
- [bibus » configurationFolder » ppsQueueLimit](#)
- [bibus » dispatcher » ppsQueueLimit](#)
- [bibus » powerPlayService » ppsQueueLimit](#)
- [bibus » schedule » priority](#)
- [bibus » configuration » prsConnections](#)
- [bibus » configurationFolder » prsConnections](#)
- [bibus » dispatcher » prsConnections](#)
- [bibus » planningRuntimeService » prsConnections](#)
- [bibus » configuration » prsPeakConnections](#)
- [bibus » configurationFolder » prsPeakConnections](#)
- [bibus » dispatcher » prsPeakConnections](#)
- [bibus » planningRuntimeService » prsPeakConnections](#)
- [bibus » configuration » ptsConnections](#)
- [bibus » configurationFolder » ptsConnections](#)
- [bibus » dispatcher » ptsConnections](#)
- [bibus » planningTaskService » ptsConnections](#)
- [bibus » configuration » ptsPeakConnections](#)
- [bibus » configurationFolder » ptsPeakConnections](#)
- [bibus » dispatcher » ptsPeakConnections](#)
- [bibus » planningTaskService » ptsPeakConnections](#)
- [bibus » configuration » qsIdleConnectionTimeout](#)
- [bibus » configurationFolder » qsIdleConnectionTimeout](#)

- [bibus » dispatcher » qsIdleConnectionTimeout](#)
- [bibus » queryService » qsIdleConnectionTimeout](#)
- [bibus » configuration » qsInitialJVMHeapSize](#)
- [bibus » configurationFolder » qsInitialJVMHeapSize](#)
- [bibus » dispatcher » qsInitialJVMHeapSize](#)
- [bibus » queryService » qsInitialJVMHeapSize](#)
- [bibus » configuration » qsInitialJVMNurserySize](#)
- [bibus » configurationFolder » qsInitialJVMNurserySize](#)
- [bibus » dispatcher » qsInitialJVMNurserySize](#)
- [bibus » queryService » qsInitialJVMNurserySize](#)
- [bibus » configuration » qsJVMHeapSizeLimit](#)
- [bibus » configurationFolder » qsJVMHeapSizeLimit](#)
- [bibus » dispatcher » qsJVMHeapSizeLimit](#)
- [bibus » queryService » qsJVMHeapSizeLimit](#)
- [bibus » configuration » qsJVMNurserySizeLimit](#)
- [bibus » configurationFolder » qsJVMNurserySizeLimit](#)
- [bibus » dispatcher » qsJVMNurserySizeLimit](#)
- [bibus » queryService » qsJVMNurserySizeLimit](#)
- [bibus » configuration » qsMultiDimensionalQuerySizeLimit](#)
- [bibus » configurationFolder » qsMultiDimensionalQuerySizeLimit](#)
- [bibus » dispatcher » qsMultiDimensionalQuerySizeLimit](#)
- [bibus » queryService » qsMultiDimensionalQuerySizeLimit](#)
- [bibus » configuration » qsResultSetCacheQueryTimeThreshold](#)
- [bibus » configurationFolder » qsResultSetCacheQueryTimeThreshold](#)
- [bibus » dispatcher » qsResultSetCacheQueryTimeThreshold](#)
- [bibus » queryService » qsResultSetCacheQueryTimeThreshold](#)
- [bibus » configuration » qsROLAPCubeAdministrationCommandTimeout](#)
- [bibus » configurationFolder » qsROLAPCubeAdministrationCommandTimeout](#)
- [bibus » dispatcher » qsROLAPCubeAdministrationCommandTimeout](#)
- [bibus » queryService » qsROLAPCubeAdministrationCommandTimeout](#)
- [bibus » configuration » qsVerboseGCLogLimit](#)
- [bibus » configurationFolder » qsVerboseGCLogLimit](#)
- [bibus » dispatcher » qsVerboseGCLogLimit](#)
- [bibus » queryService » qsVerboseGCLogLimit](#)
- [bibus » configuration » rdsMaximumDataSize](#)
- [bibus » configurationFolder » rdsMaximumDataSize](#)
- [bibus » dispatcher » rdsMaximumDataSize](#)
- [bibus » reportDataService » rdsMaximumDataSize](#)
- [bibus » historyDetailDeploymentSummary » replacedObjectCount](#)
- [bibus » configuration » rmdsAffineConnections](#)
- [bibus » configurationFolder » rmdsAffineConnections](#)
- [bibus » dispatcher » rmdsAffineConnections](#)
- [bibus » relationalMetadataService » rmdsAffineConnections](#)

- [bibus » configuration » rmdsConnections](#)
- [bibus » configurationFolder » rmdsConnections](#)
- [bibus » dispatcher » rmdsConnections](#)
- [bibus » relationalMetadataService » rmdsConnections](#)
- [bibus » configuration » rmdsExecutionTimeLimit](#)
- [bibus » configurationFolder » rmdsExecutionTimeLimit](#)
- [bibus » dispatcher » rmdsExecutionTimeLimit](#)
- [bibus » relationalMetadataService » rmdsExecutionTimeLimit](#)
- [bibus » configuration » rmdsNonAffineConnections](#)
- [bibus » configurationFolder » rmdsNonAffineConnections](#)
- [bibus » dispatcher » rmdsNonAffineConnections](#)
- [bibus » relationalMetadataService » rmdsNonAffineConnections](#)
- [bibus » configuration » rmdsPeakAffineConnections](#)
- [bibus » configurationFolder » rmdsPeakAffineConnections](#)
- [bibus » dispatcher » rmdsPeakAffineConnections](#)
- [bibus » relationalMetadataService » rmdsPeakAffineConnections](#)
- [bibus » configuration » rmdsPeakConnections](#)
- [bibus » configurationFolder » rmdsPeakConnections](#)
- [bibus » dispatcher » rmdsPeakConnections](#)
- [bibus » relationalMetadataService » rmdsPeakConnections](#)
- [bibus » configuration » rmdsPeakNonAffineConnections](#)
- [bibus » configurationFolder » rmdsPeakNonAffineConnections](#)
- [bibus » dispatcher » rmdsPeakNonAffineConnections](#)
- [bibus » relationalMetadataService » rmdsPeakNonAffineConnections](#)
- [bibus » configuration » rsAffineConnections](#)
- [bibus » configurationFolder » rsAffineConnections](#)
- [bibus » dispatcher » rsAffineConnections](#)
- [bibus » reportService » rsAffineConnections](#)
- [bibus » configuration » rsChartHotspotLimit](#)
- [bibus » configurationFolder » rsChartHotspotLimit](#)
- [bibus » dispatcher » rsChartHotspotLimit](#)
- [bibus » reportService » rsChartHotspotLimit](#)
- [bibus » configuration » rsExecutionTimeLimit](#)
- [bibus » configurationFolder » rsExecutionTimeLimit](#)
- [bibus » dispatcher » rsExecutionTimeLimit](#)
- [bibus » reportService » rsExecutionTimeLimit](#)
- [bibus » configuration » rsMaximumEMailAttachmentSize](#)
- [bibus » configurationFolder » rsMaximumEMailAttachmentSize](#)
- [bibus » dispatcher » rsMaximumEMailAttachmentSize](#)
- [bibus » reportService » rsMaximumEMailAttachmentSize](#)
- [bibus » configuration » rsNonAffineConnections](#)
- [bibus » configurationFolder » rsNonAffineConnections](#)
- [bibus » dispatcher » rsNonAffineConnections](#)

- [bibus » reportService » rsNonAffineConnections](#)
- [bibus » configuration » rsPDFCompressionLevel](#)
- [bibus » configurationFolder » rsPDFCompressionLevel](#)
- [bibus » dispatcher » rsPDFCompressionLevel](#)
- [bibus » reportService » rsPDFCompressionLevel](#)
- [bibus » configuration » rsPeakAffineConnections](#)
- [bibus » configurationFolder » rsPeakAffineConnections](#)
- [bibus » dispatcher » rsPeakAffineConnections](#)
- [bibus » reportService » rsPeakAffineConnections](#)
- [bibus » configuration » rsPeakMaximumProcesses](#)
- [bibus » configurationFolder » rsPeakMaximumProcesses](#)
- [bibus » dispatcher » rsPeakMaximumProcesses](#)
- [bibus » reportService » rsPeakMaximumProcesses](#)
- [bibus » configuration » rsPeakNonAffineConnections](#)
- [bibus » configurationFolder » rsPeakNonAffineConnections](#)
- [bibus » dispatcher » rsPeakNonAffineConnections](#)
- [bibus » reportService » rsPeakNonAffineConnections](#)
- [bibus » historyDetailDeploymentSummary » updatedObjectCount](#)
- [bibus » account » verticalElementsRenderingLimit](#)
- [bibus » contact » verticalElementsRenderingLimit](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `int`

is encoded as type `xs:int`

jobDefinition

Describes a job as a series of job steps and defines parameters for those job steps.

You can set the same schedule for multiple reports by creating a job. A job identifies a collection of reports, report views, and other jobs that are scheduled together and share the same schedule settings. When a scheduled job runs, all the reports in the job run.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)

- [bibus » jobStepDefinition » stepObject](#)
- [bibus » shortcut » target](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the jobDefinition class.

Table 176. Services and methods for the jobDefinition class.			
Action	Mode	Service	Method
Run	All	jobService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » jobStepDefinition](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)
- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)

Related information:

- IBM Cognos *Administration and Security Guide*

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)

is encoded as type `tns:optionArrayProp`

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus](#) » [baseClass](#) » [searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
- is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus](#) » [baseClass](#) » [ancestors](#) property instead.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)

is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Table 177. Rules for a new jobDefinition object

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

sequencing

Specifies whether the job steps can run at the same time or one after the other.

This property

- is of type [bibus » sequencingEnum](#)

is encoded as type `tns:nmtokenProp`

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

steps

Specifies the steps contained by the job definition.

jobOption – deprecated

Defines the abstract base class for all job option classes.

This class

- is deprecated and will be removed in a future version of the product
- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » jobOptionBoolean](#)

Properties

This class has the following properties.

name

Identifies the job option.

This property

- is of type [bibus » jobOptionEnum](#)
 - is encoded as type `tns:jobOptionEnum`

jobOptionBoolean – deprecated

Defines boolean values for the job options.

This class

- is deprecated and will be removed in a future version of the product
- inherits properties from the [bibus » jobOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the job option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

jobService

Defines run-time configuration parameters for the [jobService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

jsAuditLevel

Specifies the auditing level for the job service.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

jsConnections

Specifies the maximum number of connections that a process of the job service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

jsPeakConnections

Specifies the number of connections that a job service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “[System Metrics](#)” on page 1908

This property was added.

jobStepDefinition

Specifies an object to be executed as part of a job.

Instances of this class can also specify options and parameter values used when executing the objects included in the job step.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus](#) » [jobDefinition](#)

Properties

This class has the following properties.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus](#) » [baseClass](#) » [searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus](#) » [baseClass](#) » [ancestors](#) property instead.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

stepObject

Refers to an object that runs as part of the job.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay8Report](#), [bibus » baseReport](#), [bibus » contentTask](#), [bibus » exportDeployment](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » migrationTask](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), or [bibus » queryServiceTask](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

New in Version 10.1.0 – “Variable Support for Data Movement Tasks” on page 1854

This property was extended to allow instances of the [bibus » dataMovementTaskAlias](#) class to be referenced by instances of this property.

New in Version 8.4 – “Migration Service” on page 1883

This property was extended to allow instances of the [bibus » migrationTask](#) class to be referenced by instances of this property.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be referenced by instances of this property.

New in Version 10.1.0 – “Query Service Administration Task” on page 1869

This property was extended to allow instances of the [bibus » queryServiceTask](#) class to be referenced by instances of this property.

languageProp

Defines the simple property class for the language.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » account » contentLocale](#)
- [bibus » cacheOutput » locale](#)
- [bibus » contact » locale](#)
- [bibus » documentContent » locale](#)
- [bibus » output » locale](#)
- [bibus » account » productLocale](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `language`

is encoded as type `xs:string`

languageArrayProp

Defines the array property class for the `language`.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type `language`
is encoded as type `tns:languageArray`

launchable

Defines a generic object that can be manipulated by a UI component.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » shortcut » target](#)

Container Information

Contained by instances of the following classes

- [bibus » content](#)
- [bibus » folder](#)
- [bibus » package](#)

What's new

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

This class was added.

New in Version 10.1.0 – [“Multi-Instance IBM Cognos Connection” on page 1863](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)

is encoded as type `tns:deploymentReferenceArrayProp`

launchableType

Specifies the type of launchable object.

This property is used to determine which UI component should be invoked to act on the object.

The following types are defined:

application/vnd.ibm.cognos.ev

IBM

application/vnd.ibm.cognos.instance

IBM Cognos Connection remote content reference object

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 10.1.0 – “[Updated Support for IBM Cognos Express](#)” on page 1871

Added the value `application/vnd.ibm.cognos.ev.canvas` to support IBM Cognos Express Data Advisor clients.

specification

Specifies the information required to execute the object.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type [bibus » xmlEncodedXML](#)

is encoded as type `tns:xmlEncodedXMLMIMEProp`

listTenantsOptions

Defines the options for the [content](#) » [listTenants\(options\)](#) method.

References

Used by the following method parameters:

- [content](#) » [listTenants\(options\)](#) » [options](#)

What's new

New in Version 10.2.0 – “Support for multi-tenancy” on page 1838

This class was added.

Properties

This class has the following properties.

maxObjects

Specifies the maximum number of objects in the result set. Use this property in conjunction with `skipObjects` to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. When specifying a value of zero or less, all selected objects are returned.

This property

- is of type `int`
 - is encoded as type `xs:int`
- has a default value of 0

skipObjects

Specifies the number of objects to be skipped in the result set. Use this property in conjunction with `maxObjects` to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. Specify a negative value to indicate the number of objects to skip from the end of the result set.

This property

- is of type `int`
 - is encoded as type `xs:int`
- has a default value of 0

loadBalancingModeEnumProp

Defines the simple property class for the [bibus](#) » [loadBalancingModeEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [loadBalancingMode](#)
- [bibus](#) » [configurationFolder](#) » [loadBalancingMode](#)

- [bibus](#) » [dispatcher](#) » [loadBalancingMode](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [loadBalancingModeEnum](#)
is encoded as type `tns:loadBalancingModeEnum`

locale

Defines information about the user's locale.

When you specify a locale, data is rendered in a format that takes into account the linguistic and cultural preferences of the user.

All locales are predefined. You cannot define additional locales.

References

Used by the following properties:

- [bibus](#) » [configurationData](#) » [serverLocale](#)
- [bibus](#) » [configurationData](#) » [supportedContentLocales](#)
- [bibus](#) » [configurationData](#) » [supportedProductLocales](#)
- [bibus](#) » [localeArrayProp](#) » [value](#)
- [bibus](#) » [localeProp](#) » [value](#)

Properties

This class has the following properties.

description

Specifies the locale description, in the language of the locale identified in the header for the request. Determined by the authenticated user's account, this language can be changed using the SDK.

This property

- is of type `string`
is encoded as type `xs:string`
- is read-only

locale

Specifies the locale. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

Use this property to set the language for IBM Cognos Analytics object names, and the appropriate localized format for date, time, and currency values.

This property

- is of type `language`
is encoded as type `xs:string`
- is read-only

localeArrayProp

Defines the array property class for the [bibus](#) » [locale](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [locale](#)
is encoded as type `tns:localeArray`

localeMapEntry

Provides the key-value pair used to define locale mappings.

References

Used by the following properties:

- [bibus](#) » [configurationData](#) » [contentLocaleMap](#)
- [bibus](#) » [configurationData](#) » [productLocaleMap](#)
- [bibus](#) » [localeMapEntryArrayProp](#) » [value](#)
- [bibus](#) » [localeMapEntryProp](#) » [value](#)

Properties

This class has the following properties.

key

Specifies the key or lookup string used for the map entry.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the locale value for the map entry.

This property

- is of type `language`
is encoded as type `xs:string`

localeMapEntryArrayProp

Defines the array property class for the [bibus](#) » [localeMapEntry](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [localeMapEntry](#)
is encoded as type `tns:localeMapEntryArray`

localeMapEntryProp

Defines the simple property class for the [bibus](#) » [localeMapEntry](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [localeMapEntry](#)
is encoded as type `tns:localeMapEntry`

localeProp

Defines the simple property class for the [bibus](#) » [locale](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [locale](#)
is encoded as type `tns:locale`

logService

Defines run-time configuration parameters for the log service .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Related information:

- [“Log service” on page 9](#)
- [“Set the Logging Level” on page 88](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

IsAuditAdminLevel – obsolete

IsAuditLevel – obsolete

IsAuditNativeQuery – obsolete

IsAuditOtherLevel – obsolete

IsAuditUsageLevel – obsolete

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

mapOption

Defines the abstract base class for all map option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Properties

This class has the following properties.

name

Identifies the map option.

This property

- is of type [bibus » mapOptionEnum](#)
is encoded as type `tns:mapOptionEnum`

memo

Defines the class for email messages that can be sent as part of a job or by an agent.

The content of the email is included in a [bibus » memoPart](#) object that is specified in a delivery option for the memo.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asynchDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the memo class.

Table 178. Services and methods for the memo class.

Action	Mode	Service	Method
Run	All	deliveryService	<code>asynch » run(objectPath, parameterValues, options)</code>

Container Information

Contains instances of the following classes

- [bibus » history](#)

Contained by instances of the following classes

- [bibus » authoredAgentDefinition](#)

Properties

This class has the following properties.

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Table 179. Rules for a new memo object

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus](#) » [retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

memoPart

Defines the abstract base class for classes that define the content of an email message.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus](#) » [memoPartAgentObject](#)
- [bibus](#) » [memoPartAgentObjectLinks](#)

- [bibus » memoPartComposite](#)
- [bibus » memoPartMIMEAttachment](#)
- [bibus » memoPartObject](#)
- [bibus » memoPartObjectLinks](#)
- [bibus » memoPartString](#)

References

Used by the following properties:

- [bibus » memoPartComposite » parts](#)
- [bibus » deliveryOptionMemoPart » value](#)

Properties

This class has the following properties.

contentDisposition

Specifies the SMTP content disposition for this part of the delivered message. This affects the way this part of the message is displayed. For example, to display an image as part of the message instead of simply attaching it, use the value [inline](#).

This property

- is of type [bibus » smtpContentDispositionEnum](#)
is encoded as type `tns:smtpContentDispositionEnum`

name

Specifies the name of the [bibus » memoPart](#) in the delivered message.

This property

- is of type `string`
is encoded as type `xs:string`

memoPartAgentObject

Defines the [bibus » memoPart](#) class by specifying that objects produced by the referenced [bibus » agentTaskDefinition](#) object should be included in the message.

This [bibus » memoPart](#) type is processed by the [agentService](#). During processing, this object is replaced by a set of [bibus » memoPartObject](#) objects. Each of these new objects references a content store object to add to the message.

To show that these parts are related in the message, create an instance of this class in a [bibus » memoPartComposite](#) object and set the [contentType](#) property to [mixed](#).

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

format

Specifies the filter value for the output format.

Any outputs generated by the associated agent task must have the specified format before they can be included in any memo sent by the containing task. If the format is not specified, the output objects are not filtered by format.

This property

- is of type [bibus » outputFormatEnum](#)
 - is encoded as type `xs:string`
- must contain no more than 10 characters
- is searchable

locale

Specifies the filter value for the output locale.

Any outputs generated by the associated agent task must have the specified locale before they can be included in any memo sent by the containing task. If the locale not specified, the output objects are not filtered by locale.

This property

- is of type `Language`
 - is encoded as type `xs:string`
- must contain no more than 64 characters
- is searchable

searchPath

Specifies the location of the [bibus » agentTaskDefinition](#) object in the content store.

This property

- is of type [bibus » searchPathSingleObject](#)
 - is encoded as type `tns:searchPathSingleObject`

memoPartAgentObjectLinks

Defines the [bibus » memoPart](#) class by specifying that links to objects produced by the referenced [bibus » agentTaskDefinition](#) objects should be included in the message.

This [bibus » memoPart](#) type is processed by the `agentService`. During processing, this object is replaced by a [bibus » memoPartComposite](#) object. This new memo part contains the following additional parts:

- a [bibus » memoPartString](#) object that contains text
- a [bibus » memoPartMIMEAttachment](#) object that contains HTML

Each part contains a list of links to the content store objects produced by the referenced tasks during the execution of the agent.

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

links

Specifies the link information for each [bibus » agentTaskDefinition](#) object, which is used to determine the set of links to add to the message.

This property

- is an array of type [bibus » objectLink](#)
is encoded as type `tns:objectLinkArray`
- must have at least 1 item

memoPartComposite

Defines the [bibus » memoPart](#) class that contains other [bibus » memoPart](#) objects.

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

contentType

Specifies the SMTP content type for this memo part in the delivered message. The SMTP content type describes how the contained parts are related to each other.

This property

- is of type [bibus » smtpContentTypeEnum](#)
is encoded as type `tns:smtpContentTypeEnum`
- has a default value of [mixed](#)

parts

Specifies the [bibus » memoPart](#) objects that are contained by this memo part in the delivered message.

This property

- is an array of type [bibus » memoPart](#)
is encoded as type `tns:memoPartArray`

memoPartMIMEAttachment

Defines the [bibus » memoPart](#) class used to include a MIME attachment with the message, such as an image file.

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

contentID

Specifies the globally unique Content - ID for this memo part in the delivered message. Other parts of the message can use this ID to reference the image or data stored in this memo part.

This property

- is of type `string`
is encoded as type `xs:string`

contentLocation

Specifies the URL to use as the Content - Location for this memo part in the delivered message. Other parts of the message can use this URL to link to the image or data that stored in this memo part.

This property

- is of type `string`
is encoded as type `xs:string`

data

Contains the MIME data. The data format is specified in the `dataType` property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the `bibus » addOptions » dataEncoding` property and the `bibus » queryOptions » dataEncoding` property for additional details.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
is encoded as type `xs:integer`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
is encoded as type `xs:string`
- can contain at least 255 characters

memoPartObject

Defines the `bibus » memoPart` class specifying that the referenced content store object should be included in the message.

This class

- inherits properties from the `bibus » memoPart` class

Properties

This class has the following properties.

searchPath

Specifies the location of the object in the content store.

This property

- is of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObject`

memoPartObjectLinks

Defines the [bibus » memoPart](#) class by specifying that links to the referenced objects should be included in the message.

During processing, this object is replaced by a [bibus » memoPartComposite](#) object. This new memo part contains the following additional parts:

- a [bibus » memoPartString](#) object that contains text
- a [bibus » memoPartMIMEAttachment](#) object that contains HTML

Each part contains a list of links to the specified objects.

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

links

Specifies the information required to construct a link in the message for each referenced content store object.

This property

- is an array of type [bibus » objectLink](#)
 - is encoded as type `tns:objectLinkArray`
- must have at least 1 item

memoPartString

Defines the [bibus » memoPart](#) class that contains plain text.

This class

- inherits properties from the [bibus » memoPart](#) class

Properties

This class has the following properties.

text

Specifies the text for this part of the message.

This property

- is of type `string`
 - is encoded as type `xs:string`
- can contain at least 32767 characters

message

Defines the class used to specify the message text and nesting level of a message.

References

Used by the following properties:

- [bibus » CAMException » messages](#)

Properties

This class has the following properties.

messageString

Provides the message text.

This property

- is of type `string`
is encoded as type `xs:string`

nestingLevel

Specifies a number that represents the nesting level for the error message. Each level may append an additional error message, depending on the processing context.

This property

- is of type `integer`
is encoded as type `xs:integer`

metadataModelExpression

Defines the type for an expression that conforms to the IBM Cognos Framework Manager expression syntax.

This class

- inherits properties from the `string`

References

Used by the following properties:

- [bibus](#) » [parameter](#) » [useValueExpression](#)

metadataModelItemName

Defines the type for names of items in the metadata model.

This class

- inherits properties from the `string`

References

Used by the following properties:

- [bibus](#) » [drillPath](#) » [bookmarkItem](#)
- [bibus](#) » [parameterAssignmentDataItem](#) » [dataItemName](#)
- [bibus](#) » [agentTaskDefinition](#) » [filterDataItemName](#)
- [bibus](#) » [drillPath](#) » [scope](#)
- [bibus](#) » [storedProcedureTask](#) » [storedProcedureName](#)
- [bibus](#) » [metadataModelItemNameArrayProp](#) » [value](#)
- [bibus](#) » [metadataModelItemNameProp](#) » [value](#)

metadataModelItemNameArrayProp

Defines the array property class for the [bibus](#) » [metadataModelItemName](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » drillPath » scope](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » metadataModelItemName](#)
is encoded as type `tns:metadataModelItemNameArray`

metadataModelItemNameProp

Defines the simple property class for the [bibus » metadataModelItemName](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » drillPath » bookmarkItem](#)
- [bibus » agentTaskDefinition » filterDataItemName](#)
- [bibus » storedProcedureTask » storedProcedureName](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » metadataModelItemName](#)
is encoded as type `tns:metadataModelItemName`

metadataService

Defines run-time configuration parameters for the metadata service.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

mdsAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The [metadataService](#) does not use this property as this service only processes synchronous requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsAuditLevel

Specifies the auditing level for the metadata service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`

- has a default value of `minimal`
- can be acquired from a containing object

mdsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of 0 when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

For this service, low affinity requests process lineage requests that process metadata in published Framework Manager models as well as in the query section of a report specification.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1

- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsPeakAffineConnections

Specifies the number of connections that a metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsPeakMaximumProcesses

Specifies the maximum number of metadata service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “Lineage Metadata” on page 1891

This property was added.

mdsPeakNonAffineConnections

Specifies the number of connections that a metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4

- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

mdsQueueLimit

Specifies the number of seconds that a request for the metadata service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 – “Lineage Metadata” on page 1891

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

metadataServiceLineageSpecification

Defines the class for metadata service lineage specifications.

This class

- inherits properties from the `bibus » metadataServiceSpecification` class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `metadataServiceLineageSpecification` class.

Table 180. Services and methods for the `metadataServiceLineageSpecification` class.

Action	Mode	Service	Method
Run	All	metadataService	<code>asynch » runSpecification(specification, parameterValues, options)</code>
Run	Batch	batchReportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>
Run	Interactive	reportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>

What's new

New in Version 8.4 – “Lineage Metadata” on page 1891

This class was added.

New in Version 10.1.0 – “Support for Lineage Requests” on page 1855

The [reportService](#) and [batchReportService](#) services now support lineage requests.

metadataServiceModelInformationSpecification

This class

- inherits properties from the `bibus » metadataServiceSpecification` class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `metadataServiceModelInformationSpecification` class.

Table 181. Services and methods for the `metadataServiceModelInformationSpecification` class.

Action	Mode	Service	Method
Run	All	metadataService	<code>asynch » runSpecification(specification, parameterValues, options)</code>

What's new

New in Version 8.4 – “Package Data Sources” on page 1895

This class was added.

metadataServiceSpecification

Defines the class for metadata service specifications.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the `bibus » asynchSpecification` class

Derived Classes

- `bibus » metadataServiceLineageSpecification`

- [bibus » metadataServiceModelInformationSpecification](#)

What's new

New in Version 8.4 – “Lineage Metadata” on page 1891

This class was added.

metricsDataSourceETLTask

Defines the metrics data source ETL task.

This class

- inherits properties from the [bibus » baseDataIntegrationTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the metricsDataSourceETLTask class.

<i>Table 182. Services and methods for the metricsDataSourceETLTask class.</i>			
Action	Mode	Service	Method
Run	All	dataIntegrationService	asynch » run(objectPath, parameterValues, options)

metricsExportTask

Defines the metrics export task.

This class

- inherits properties from the [bibus » baseDataIntegrationTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the metricsExportTask class.

<i>Table 183. Services and methods for the metricsExportTask class.</i>			
Action	Mode	Service	Method
Run	All	dataIntegrationService	asynch » run(objectPath, parameterValues, options)

metricsFileImportTask

Defines the metrics file import task.

This class

- inherits properties from the [bibus » baseDataIntegrationTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the metricsFileImportTask class.

<i>Table 184. Services and methods for the metricsFileImportTask class.</i>			
Action	Mode	Service	Method
Run	All	dataIntegrationService	asynch » run(objectPath, parameterValues, options)

metricsMaintenanceTask

Defines the metrics maintenance task.

This class

- inherits properties from the [bibus](#) » [baseDataIntegrationTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the metricsMaintenanceTask class.

<i>Table 185. Services and methods for the metricsMaintenanceTask class.</i>			
Action	Mode	Service	Method
Run	All	dataIntegrationService	asynch » run(objectPath, parameterValues, options)

metricsManagerService

Defines run-time configuration parameters for the metrics manager service .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

mmsAuditLevel

Specifies the auditing level for the metrics manager service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

mmsConnections

Specifies the maximum number of connections that a process of the metrics manager service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

mmsPeakConnections

Specifies the number of connections that a metrics manager service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

migrationCubeMapping

Defines a migration task mapping for a PowerCube.

This class is not used.

What's new

New in Version 8.4 — “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

source

Specifies the name of the cube in the source object.

This property

- is of type `string`
is encoded as type `xs:string`

target

Specifies the [bibus » package](#) that replaces the PowerCube in the target object.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

migrationMapping

Defines a migration task mapping.

This class is not used.

References

Used by the following properties:

- [bibus](#) » [migrationTaskOptionMappingArray](#) » [value](#)

What's new

New in Version 8.4 – “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

source

Identifies the source class to be migrated.

This property

- is of type [bibus](#) » [classEnum](#)
is encoded as type `tns:classEnum`

target

Specifies the target class for migrated objects.

This property

- is of type [bibus](#) » [classEnum](#)
is encoded as type `tns:classEnum`

migrationService

Defines run-time configuration parameters for the [migrationService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

What's new

New in Version 8.4 – [“Migration Service” on page 1883](#)

This class was added.

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

misAuditLevel

Specifies the auditing level for the migration service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was added.

misConnections

Specifies the maximum number of connections that a process of the migration service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“Migration Service” on page 1883](#)

This property was added.

misPeakConnections

Specifies the number of connections that a migration service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – “Migration Service” on page 1883

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

migrationServiceSpecification

Defines the migration service specification type.

This class

- inherits properties from the `bibus » asynchSpecification` class

References

Used by the following properties:

- `bibus » migrationTask » specification`
- `bibus » migrationServiceSpecificationProp » value`

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `migrationServiceSpecification` class.

<i>Table 186. Services and methods for the migrationServiceSpecification class.</i>			
Action	Mode	Service	Method
Run	All	<code>migrationService</code>	<code>asynch » runSpecification(specification, parameterValues, options)</code>

What's new

New in Version 8.4 — “Migration Service” on page 1883

This class was added.

migrationServiceSpecificationProp

Defines the simple property class for the [bibus](#) » [migrationServiceSpecification](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [migrationTask](#) » [specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [migrationServiceSpecification](#)
is encoded as type `tns:migrationServiceSpecification`

migrationTask

Defines a migration task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [eventRecord](#) » [runnable](#)
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#)
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the migrationTask class.

Table 187. Services and methods for the migrationTask class.

Action	Mode	Service	Method
Run	All	migrationService	<code>asynch » run(objectPath, parameterValues, options)</code>

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)

What's new

New in Version 8.4 — “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

<i>Table 188. Rules for a new migrationTask object</i>			
Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point (!)
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

specification

Specifies the specification for the task.

This property

- is of type [bibus » migrationServiceSpecification](#)

is encoded as type `tns:migrationServiceSpecificationProp`

migrationTaskOption

Defines the abstract base class for all migration task option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » migrationTaskOptionMappingArray](#)
- [bibus » migrationTaskOptionResolution](#)
- [bibus » migrationTaskOptionSearchPathSingleObject](#)
- [bibus » migrationTaskOptionSearchPathSingleObjectArray](#)

What's new

New in Version 8.4 — “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

name

Identifies the migration task option.

This property

- is of type [bibus » migrationTaskOptionEnum](#)

is encoded as type `tns:migrationTaskOptionEnum`

migrationTaskOptionMappingArray

Defines [bibus » migrationMapping](#) values for the migration task options.

This class

- inherits properties from the [bibus » migrationTaskOption](#) class

What's new

New in Version 8.4 – “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the migration task option.

This property

- is an array of type [bibus » migrationMapping](#)
is encoded as type `tns:migrationMappingArray`

migrationTaskOptionResolution

Defines [bibus » conflictResolutionEnum](#) values for the migration task options.

This class

- inherits properties from the [bibus » migrationTaskOption](#) class

What's new

New in Version 8.4 – “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the migration task option.

This property

- is of type [bibus » conflictResolutionEnum](#)
is encoded as type `tns:conflictResolutionEnum`

migrationTaskOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the migration task options.

This class

- inherits properties from the [bibus » migrationTaskOption](#) class

What's new

New in Version 8.4 – “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the migration task option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

migrationTaskOptionSearchPathSingleObjectArray

Defines [bibus » searchPathSingleObject](#) values for the migration task options.

This class

- inherits properties from the [bibus » migrationTaskOption](#) class

What's new

New in Version 8.4 – “Migration Service” on page 1883

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the migration task option.

This property

- is an array of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

mobileOption

Defines the abstract base class for all mobile option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » mobileOptionSearchPathMultipleObjectArray](#)

What's new

New in Version 8.4 – “Mobile Service” on page 1891

This class was added.

Properties

This class has the following properties.

name

Identifies the mobile option.

This property

- is of type [bibus » mobileOptionEnum](#)
is encoded as type `tns:mobileOptionEnum`

mobileOptionSearchPathMultipleObjectArray

Defines [bibus » searchPathMultipleObject](#) values for the mobile options.

This class

- inherits properties from the [bibus » mobileOption](#) class

What's new

New in Version 8.4 — “Mobile Service” on page 1891

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the mobile option.

This property

- is an array of type [bibus » searchPathMultipleObject](#)
is encoded as type `tns:searchPathMultipleObjectArray`

mobileService

Defines run-time configuration parameters for the [mobileService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Related information:

- *IBM Cognos Administration and Security Guide*

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

mbsAuditLevel

Specifies the auditing level for the mobile service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

mbsConnections

Specifies the maximum number of connections that a process of the mobile service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

mbsPeakConnections

Specifies the number of connections that a mobile service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “[System Metrics](#)” on page 1908

This property was added.

model

Contains a published IBM Cognos Framework Manager model.

Administrators can set a distinct security policy for this object that differs from the security policy set for the package in which it is contained.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the `bibus » uiClass` class

References

Used by the following properties:

- `bibus » authoredPowerPlay8Report » metadataModel`
- `bibus » authoredReport » metadataModel`
- `bibus » baseDataIntegrationTask » metadataModel`
- `bibus » reportCache » metadataModel`
- `bibus » reportVersion » metadataModel`
- `bibus » storedProcedureTask » metadataModel`

Container Information

Contains instances of the following classes

- [bibus](#) » [modelView](#)

Contained by instances of the following classes

- [bibus](#) » [baseROLAPDataSource](#)
- [bibus](#) » [package](#)

Related information:

- *IBM Cognos Framework Manager Developer Guide*

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

dataSetBase

This property references the [dataSet](#) object that this model is based on.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must contain at most 1 item

New in Version 10.2.2 – [My data sets](#)

This property was added.

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`

- is read-only
- is searchable

New in Version 8.4 – “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

model

Specifies the published Framework Manager model. When retrieved as an attachment, the content type of the attachment is `application/xml`.

Updates to this property that use an attachment must be of the same content type.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeMIMEProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

New in Version 10.1.0 – “Query Modes” on page 1865

This property was added.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

views

Contains the views for this Framework Manager model.

Administrators can set a distinct security policy for each `modelView` object that differs from the security policy of other views, or of the model in which they are contained.

modelView

Instances of this class are used to associate security policies with views stored in the IBM Cognos Framework Manager model.

Administrators can set a distinct security policy for each modelView object that differs from the security policy of other views, or of the model that contains them.

This distinct security policy implements the viewing permissions for a particular modelView object at run time.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » model](#)

Related information:

- *IBM Cognos Framework Manager Developer Guide*

monitorOption

Defines the abstract base class for all Monitor Service option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » monitorOptionBoolean](#)

Properties

This class has the following properties.

name

Identifies the Monitor Service option.

This property

- is of type [bibus » monitorOptionEnum](#)
is encoded as type `tns:monitorOptionEnum`

monitorOptionBoolean

Defines boolean values for the Monitor Service options.

This class

- inherits properties from the [bibus » monitorOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the Monitor Service option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

monitorService

Defines run-time configuration parameters for the [monitorService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus](#) » [dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [runTimeState](#)
- [bibus](#) » [systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus](#) » [dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

msAuditLevel

Specifies the auditing level for the monitor service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)

is encoded as type `tns:auditLevelEnumProp`

- has a default value of [minimal](#)
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

monitorServiceSpecification

Defines the type for monitor service specifications.

An event specification specifies filtering information required to query currently running events. The [bibus](#) » [monitorService](#) manages events that are in an executing, inactive, suspended, or pending state.

This class

- inherits properties from the [bibus](#) » [asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `monitorServiceSpecification` class.

Action	Mode	Service	Method
Run	All	monitorService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.3 — “Schedule Management” on page 1907

This class was added.

moveOptions

Defines the options associated with the [content](#) » [move\(objects, targetPath, options\)](#) method request.

References

Used by the following method parameters:

- [content](#) » [move\(objects, targetPath, options\)](#) » [options](#)
- [content](#) » [moveRename\(objects, targetPath, newNames, options\)](#) » [options](#)

Properties

This class has the following properties.

faultIfObjectReferenced

Specifies how references to objects being deleted or replaced are processed by Content Manager.

If this property is set to `false`, any reference to the objects being deleted are automatically deleted. If this property is set to `true`, Content Manager generates a fault if any object in the content store continues to reference any of the deleted objects.

During recursive delete, a fault is not generated if one deleted object refers to another object that is also deleted.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

updateAction

Specifies what happens when an object already exists at the object's new location in the content store: either the properties of the object are updated, or the object and all its descendants are deleted and a new object is created with the properties specified in the request.

This property

- is of type [bibus](#) » [updateActionEnum](#)
 - is encoded as type `tns:updateActionEnum`

mruFolder

Contains shortcuts and URL objects that refer to the most recently used (MRU) objects.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [shortcut](#)
- [bibus](#) » [URL](#)

Contained by instances of the following classes

- [bibus](#) » [account](#)

What's new

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

Shortcuts contained in this folder are no longer constrained to refer only to packages.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

items

Contains shortcuts and URLs to the most recently used (MRU) objects.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » shortcut](#), and [bibus » URL](#)

New in Version 8.3 – “[mruFolder Retention Rules](#)” on page 1920

The retention rule for this property has been changed to use the [bibus » baseClass » modificationTime](#) property instead of the [bibus » baseClass » creationTime](#) property.

New in Version 10.1.0 – “[Content Pane Improvements](#)” on page 1862

The retention rules for this property now allow up to 15 objects of each class to be retained.

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » shortcut class		15	bibus » baseClass » modificationTime property
bibus » URL class		15	bibus » baseClass » modificationTime property

This property

- is an array of type [bibus » retentionRule](#)
 - is encoded as type `tns:retentionRuleArrayProp`

multilingualString

Use this class to associate a locale with a string value to specify property values in multiple locales. Instances of this class are always used in an array.

References

Used by the following properties:

- [bibus » baseParameter » caption](#)
- [bibus » asynchDetailIndexData » displayName](#)
- [bibus » deploymentOptionMultilingualString » value](#)
- [bibus » indexTerm » value](#)
- [bibus » multilingualStringProp » value](#)
- [bibus » runOptionMultilingualString » value](#)

Properties

This class has the following properties.

locale

Specifies the locale for the string value.

This property

- is of type `language`
is encoded as type `xs:string`

value

Specifies the string value.

This property

- is of type `string`
is encoded as type `xs:string`

multilingualStringProp

Defines the property class for arrays of type `multilingualString`.

This class is used with multilingual properties.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » uiClass » description](#)
- [bibus » uiClass » screenTip](#)
- [bibus » baseRSSTask » title](#)
- [bibus » authoredReport » triggerDescription](#)

Properties

This class has the following properties.

value

Contains the values of type `multilingualString`.

This property

- is an array of type `bibus » multilingualString`
is encoded as type `tns:multilingualStringArray`

multilingualToken

Use this class to associate a locale with a token value to specify property values in multiple locales.

Instances of this class are always used in an array.

References

Used by the following properties:

- `bibus » deploymentImportRule » name`
- `bibus » deploymentObjectInformation » name`
- `bibus » deploymentReference » name`
- `bibus » navigationPath » name`
- `bibus » powerPlay8OptionSaveAs » objectName`
- `bibus » runOptionSaveAs » objectName`
- `bibus » multilingualTokenProp » value`

Properties

This class has the following properties.

locale

Specifies the locale for the value.

This property

- is of type `language`
is encoded as type `xs:string`

value

Specifies the value.

This property

- is of type `token`
is encoded as type `xs:string`

multilingualTokenProp

Defines the property class for arrays of type `multilingualToken`.

This class is used with multilingual properties.

This class

- inherits properties from the `bibus » baseProp` class

References

Used by the following properties:

- [bibus](#) » [baseClass](#) » [name](#)

Properties

This class has the following properties.

value

Contains the values of type `multilingualToken`.

This property

- is an array of type [bibus](#) » [multilingualToken](#)
is encoded as type `tns:multilingualTokenArray`

namespace

Contains the information used to define a namespace.

A namespace is a collection of accounts, users, groups, and roles from a security provider. IBM Cognos Analytics also uses a built-in namespace, the Cognos namespace, that contains the objects, such as groups, roles, data sources, distribution lists, and contacts.

Since most instances are based on information in a security provider, IBM Cognos Analytics may constrain your ability to manipulate these objects. For example, you may not be able to change the value of a property, or the property may not return a value when queried.

The name of an instance may match the name of a sibling. In this case, a different search strategy should be used to distinguish between objects having the same name.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [dataSourceSignon](#) » [consumers](#)
- [bibus](#) » [dataSourceConnection](#) » [credentialNamespaces](#)
- [bibus](#) » [session](#) » [identity](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [account](#)
- [bibus](#) » [baseROLAPDataSource](#)
- [bibus](#) » [contact](#)
- [bibus](#) » [dataSource](#)
- [bibus](#) » [distributionList](#)
- [bibus](#) » [group](#)
- [bibus](#) » [namespaceFolder](#)
- [bibus](#) » [printer](#)
- [bibus](#) » [resource](#)

- [bibus » role](#)

Contained by instances of the following classes

- [bibus » directory](#)

Related information:

- [Chapter 4, “Managing security,” on page 45](#)

Properties

This class has the following properties.

active

Specifies whether the namespace has been configured for use by IBM Cognos Analytics.

A namespace that has not been configured has a [bibus » namespace » active](#) property value of `false` and can be deleted from the content store. Only System Administrators can delete a namespace.

A namespace that has been configured has a [bibus » namespace » active](#) property value of `true` and cannot be deleted.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only

capabilities

Contains the set of capabilities that are supported by this namespace.

This property

- is an array of type [bibus » namespaceCapabilityEnum](#)
 - is encoded as type `tns:nmtokenArrayProp`
- is read-only

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

items

Contains the security-related objects in this namespace instance.

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

This property was extended to allow instances of the [bibus » resource](#) class to be contained by instances of this property.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was extended to allow instances of the [bibus » baseROLAPDataSource](#) class to be contained by instances of this property.

namespaceFormat

Specifies the encoding used for the namespace properties, such as the CAM ID.

This property

- is of type anyURI
 - is encoded as type `tns:anyURIProp`
- is read-only

repositoryRules

This property

- is an array of type [bibus » repositoryRule](#)
 - is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The [bibus » account](#), [bibus » namespace](#), and [bibus » namespaceFolder](#) classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

namespaceFolder

Contains folders or organizational units, as stored in an external security directory.

Since most instances are based on information in a security system, IBM Cognos Analytics may constrain your ability to manipulate these objects. For example, you may not be able to change the value of a property, or the property may not return a value when queried.

The name of an instance may match the name of a sibling. In this case, a different search strategy should be used to distinguish between objects having the same name.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » account](#)
- [bibus » contact](#)
- [bibus » distributionList](#)
- [bibus » group](#)
- [bibus » namespaceFolder](#)
- [bibus » resource](#)
- [bibus » role](#)

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Properties

This class has the following properties.

items

Contains the security-related objects for this instance of a folder or organizational unit.

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

This property was extended to allow instances of the [bibus » resource](#) class to be contained by instances of this property.

repositoryRules

This property

- is an array of type [bibus » repositoryRule](#)
is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The [bibus » account](#), [bibus » namespace](#), and [bibus » namespaceFolder](#) classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

nameValue

Contains a named value.

References

Used by the following properties:

- [bibus](#) » [runOptionNameValueArray](#) » [value](#)

Properties

This class has the following properties.

name

Specifies the name of the value.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the value.

This property

- is of type `string`
is encoded as type `xs:string`

navigationPath

Defines a navigation path between a source and a target, to support drill-through between reports and queries.

References

Used by the following properties:

- [bibus](#) » [authoredReport](#) » [paths](#)
- [bibus](#) » [navigationPathArrayProp](#) » [value](#)
- [bibus](#) » [navigationPathProp](#) » [value](#)

Properties

This class has the following properties.

name

Specifies the name of the navigation path.

This property

- is of type `token`
is encoded as type `tns:multilingualTokenArray`
- is multilingual

target

Specifies the drill-through target.

If the target was moved, deleted, or renamed since the navigation path was specified, the drill-through will fail.

This property

- is an array of type [bibus](#) » [baseClass](#)

has items that must be of class [bibus » baseReport](#)

is encoded as type `tns:baseClassArray`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

navigationPathArrayProp

Defines the array property class for the [bibus » navigationPath](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » authoredReport » paths](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » navigationPath](#)
is encoded as type `tns:navigationPathArray`

navigationPathProp

Defines the simple property class for the [bibus » navigationPath](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » navigationPath](#)
is encoded as type `tns:navigationPath`

ncnameProp

Defines the simple property class for the NCName.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [dataSourceNameBinding](#) » [replacement](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `NCName`
is encoded as type `xs:string`

nil

Defines an empty class, for situations in which the referenced object no longer exists in the content store, or cannot be retrieved due to security restrictions.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [baseClass](#) class

nmtokenArrayProp

Defines the array property class for the NMTOKEN.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [namespace](#) » [capabilities](#)
- [bibus](#) » [baseReport](#) » [defaultOutputFormat](#)
- [bibus](#) » [baseClass](#) » [permissions](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type `NMTOKEN`
is encoded as type `tns:nmtokenArray`

nmtokenProp

Defines the simple property class for the NMTOKEN.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » schedule » dailyPeriod](#)
- [bibus » deploymentDetail » deployedObjectUsage](#)
- [bibus » schedule » endType](#)
- [bibus » account » format](#)
- [bibus » contact » format](#)
- [bibus » documentContent » format](#)
- [bibus » output » format](#)
- [bibus » dataSourceConnection » isolationLevel](#)
- [bibus » documentContent » lastPage](#)
- [bibus » output » lastPage](#)
- [bibus » schedule » monthlyRelativeDay](#)
- [bibus » schedule » monthlyRelativeWeek](#)
- [bibus » account » pageOrientation](#)
- [bibus » dataSourceNameBinding » qualifier](#)
- [bibus » history » scheduleType](#)
- [bibus » authoredAgentDefinition » sequencing](#)
- [bibus » jobDefinition » sequencing](#)
- [bibus » history » status](#)
- [bibus » schedule » type](#)
- [bibus » pageDefinition » unit](#)
- [bibus » baseClass » usage](#)
- [bibus » schedule » yearlyAbsoluteMonth](#)
- [bibus » schedule » yearlyRelativeDay](#)
- [bibus » schedule » yearlyRelativeMonth](#)
- [bibus » schedule » yearlyRelativeWeek](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type NMTOKEN

is encoded as type `xs:string`

nonNegativeIntegerProp

Defines the simple property class for the `nonNegativeInteger`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » batchReportService » brsMaximumProcesses](#)
- [bibus » configuration » brsMaximumProcesses](#)
- [bibus » configurationFolder » brsMaximumProcesses](#)
- [bibus » dispatcher » brsMaximumProcesses](#)
- [bibus » cacheOutput » dataSize](#)
- [bibus » dashboard » dataSize](#)
- [bibus » documentContent » dataSize](#)
- [bibus » graphic » dataSize](#)
- [bibus » historyDetailDataMovementService » dataSize](#)
- [bibus » output » dataSize](#)
- [bibus » page » dataSize](#)
- [bibus » visualization » dataSize](#)
- [bibus » uiClass » displaySequence](#)
- [bibus » baseClass » position](#)
- [bibus » configuration » rsMaximumProcesses](#)
- [bibus » configurationFolder » rsMaximumProcesses](#)
- [bibus » dispatcher » rsMaximumProcesses](#)
- [bibus » reportService » rsMaximumProcesses](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `nonNegativeInteger`
is encoded as type `xs:integer`

objectLink

Defines the class that specifies the information that is necessary to generate a link to an object in the content store.

References

Used by the following properties:

- [bibus » memoPartAgentObjectLinks » links](#)
- [bibus » memoPartObjectLinks » links](#)

Properties

This class has the following properties.

label

Specifies the label for the link in the delivered message.

This property

- is of type `string`
is encoded as type `xs:string`

objectClass

Specifies the class of the object to be linked.

This property

- is of type `bibus » classEnum`
is encoded as type `tns:classEnum`

searchPath

Specifies the location of the object to be linked in the content store.

This property

- is of type `bibus » searchPathSingleObject`
is encoded as type `tns:searchPathSingleObject`

option

Defines the common base class for all method options.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- `bibus » agentOption`
- `bibus » archiveOption`
- `bibus » asynchOption`
- `bibus » contentManagerQueryOption`
- `bibus » contentTaskOption`
- `bibus » contextOption`
- `bibus » copyAccountOption`
- `bibus » dataIntegrationTaskOption`
- `bibus » deliveryOption`
- `bibus » deploymentOption`
- `bibus » drillOption`
- `bibus » drillThroughOption`
- `bibus » genericOption`
- `bibus » indexOption`
- `bibus » indexTermOption`
- `bibus » mapOption`
- `bibus » migrationTaskOption`
- `bibus » mobileOption`
- `bibus » monitorOption`

- [bibus » pagingOption](#)
- [bibus » pdfOption](#)
- [bibus » portalOption](#)
- [bibus » powerPlay8Option](#)
- [bibus » reportServiceQueryDrillPathOption](#)
- [bibus » reportServiceQueryOption](#)
- [bibus » reportStudioOption](#)
- [bibus » rssOption](#)
- [bibus » runOption](#)
- [bibus » specificationOption](#)
- [bibus » subscriptionOption](#)
- [bibus » validateOption](#)

References

Used by the following properties:

- [bibus » session » groupAndRoleSettings](#)
- [bibus » account » options](#)
- [bibus » agentTaskDefinition » options](#)
- [bibus » archiveDescriptor » options](#)
- [bibus » asynchDetailDrillThroughRequest » options](#)
- [bibus » asynchRequest » options](#)
- [bibus » asynchSecondaryRequest » options](#)
- [bibus » baseAgentDefinition » options](#)
- [bibus » baseDataIntegrationTask » options](#)
- [bibus » baseDataMovementTask » options](#)
- [bibus » basePowerPlay8Report » options](#)
- [bibus » baseReport » options](#)
- [bibus » baseRSSTask » options](#)
- [bibus » contentTask » options](#)
- [bibus » drillPath » options](#)
- [bibus » exportDeployment » options](#)
- [bibus » group » options](#)
- [bibus » historyDetailRequestArguments » options](#)
- [bibus » humanTask » options](#)
- [bibus » importDeployment » options](#)
- [bibus » indexUpdateTask » options](#)
- [bibus » jobDefinition » options](#)
- [bibus » jobStepDefinition » options](#)
- [bibus » memo » options](#)
- [bibus » migrationTask » options](#)
- [bibus » model » options](#)
- [bibus » planningMacroTask » options](#)
- [bibus » planningTask » options](#)

- [bibus » queryServiceTask » options](#)
- [bibus » reportCache » options](#)
- [bibus » reportVersion » options](#)
- [bibus » role » options](#)
- [bibus » schedule » options](#)
- [bibus » scheduledEvent » options](#)
- [bibus » storedProcedureTask » options](#)
- [bibus » webServiceTask » options](#)
- [bibus » configuration » overrideOptions](#)
- [bibus » configuration » serviceDefaultOptions](#)
- [bibus » drillPath » targetOptions](#)
- [bibus » optionArrayProp » value](#)
- [bibus » optionProp » value](#)

Used by the following method parameters:

- [asynch » run\(objectPath, parameterValues, options\) » options](#)
- [asynch » runSpecification\(specification, parameterValues, options\) » options](#)
- [asynch » wait\(conversation, parameterValues, options\) » options](#)
- [content » copyAccount\(sourceAccountPath, targetAccountPath, options\) » options](#)
- [content » deleteAccount\(objectPath, options\) » options](#)
- [deployment » getDeploymentOptions\(archive, options\) » options](#)
- [drillThrough » convertDrillThroughContext\(inputContext, parameterValues, options\) » options](#)
- [drillThrough » findDrillThroughPaths\(objectPath, parameterValues, options\) » options](#)
- [drillThrough » queryDrillPath\(objectPath, parameterValues, options\) » options](#)
- [event » runAt\(startTime, objectPath, parameterValues, options\) » options](#)
- [indexTerm » addTermAssociation\(term, parameterValues, options\) » options](#)
- [indexTerm » deleteTermAssociation\(term, parameterValues, options\) » options](#)
- [indexUpdate » add\(objectPath, parameterValues, options\) » options](#)
- [indexUpdate » delete\(objectPath, parameterValues, options\) » options](#)
- [indexUpdate » get\(objectPath, parameterValues, options\) » options](#)
- [paging » currentPage\(conversation, parameterValues, options\) » options](#)
- [paging » firstPage\(conversation, parameterValues, options\) » options](#)
- [paging » lastPage\(conversation, parameterValues, options\) » options](#)
- [paging » nextPage\(conversation, parameterValues, options\) » options](#)
- [paging » previousPage\(conversation, parameterValues, options\) » options](#)
- [parameter » collectParameterValues\(objectPath, parameterValues, options\) » options](#)
- [parameter » collectParameterValuesSpecification\(specification, parameterValues, options\) » options](#)
- [parameter » getParameters\(objectPath, parameterValues, options\) » options](#)
- [parameter » getParametersSpecification\(specification, parameterValues, options\) » options](#)
- [promptPaging » back\(conversation, parameterValues, options\) » options](#)
- [promptPaging » forward\(conversation, parameterValues, options\) » options](#)
- [promptPaging » getPromptValues\(conversation, parameterValues, options\) » options](#)
- [report » deliver\(conversation, parameterValues, options\) » options](#)

- [report](#) » [drill\(conversation, parameterValues, options\)](#) » [options](#)
- [report](#) » [getContext\(conversation, parameterValues, options\)](#) » [options](#)
- [report](#) » [getObjectContext\(objectPath, parameterValues, options\)](#) » [options](#)
- [report](#) » [getOutput\(conversation, parameterValues, options\)](#) » [options](#)
- [report](#) » [lineage\(conversation, parameterValues, options\)](#) » [options](#)
- [report](#) » [query\(objectPath, parameterValues, options\)](#) » [options](#)
- [report](#) » [render\(conversation, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [clearCubeWorkloadLog\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [getCubeState\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [refreshCubeDataCache\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [refreshCubeMemberCache\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [refreshCubeSecurity\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [restartCubes\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [startCubes\(cubeNames, parameterValues, options\)](#) » [options](#)
- [rolapCubeAdministration](#) » [stopCubes\(cubeNames, parameterValues, options\)](#) » [options](#)
- [validate](#) » [validate\(objectPath, parameterValues, options\)](#) » [options](#)
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) » [options](#)

Used by the following method return values:

- [deployment](#) » [getDeploymentOptions\(archive, options\)](#) » [result](#)

optionArrayProp

Defines the array property class for the [bibus](#) » [option](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [session](#) » [groupAndRoleSettings](#)
- [bibus](#) » [account](#) » [options](#)
- [bibus](#) » [agentTaskDefinition](#) » [options](#)
- [bibus](#) » [baseAgentDefinition](#) » [options](#)
- [bibus](#) » [baseDataIntegrationTask](#) » [options](#)
- [bibus](#) » [baseDataMovementTask](#) » [options](#)
- [bibus](#) » [basePowerPlay8Report](#) » [options](#)
- [bibus](#) » [baseReport](#) » [options](#)
- [bibus](#) » [baseRSSTask](#) » [options](#)
- [bibus](#) » [contentTask](#) » [options](#)
- [bibus](#) » [drillPath](#) » [options](#)
- [bibus](#) » [exportDeployment](#) » [options](#)
- [bibus](#) » [group](#) » [options](#)

- [bibus » historyDetailRequestArguments » options](#)
- [bibus » humanTask » options](#)
- [bibus » importDeployment » options](#)
- [bibus » indexUpdateTask » options](#)
- [bibus » jobDefinition » options](#)
- [bibus » jobStepDefinition » options](#)
- [bibus » memo » options](#)
- [bibus » migrationTask » options](#)
- [bibus » model » options](#)
- [bibus » planningMacroTask » options](#)
- [bibus » planningTask » options](#)
- [bibus » queryServiceTask » options](#)
- [bibus » reportCache » options](#)
- [bibus » reportVersion » options](#)
- [bibus » role » options](#)
- [bibus » schedule » options](#)
- [bibus » storedProcedureTask » options](#)
- [bibus » webServiceTask » options](#)
- [bibus » configuration » overrideOptions](#)
- [bibus » configuration » serviceDefaultOptions](#)
- [bibus » drillPath » targetOptions](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`

optionProp

Defines the simple property class for the [bibus » option](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » option](#)
is encoded as type `tns:option`

output

Provides report output.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus](#) » [baseClass](#) class

References

Used by the following properties:

- [bibus](#) » [baseAgentDefinition](#) » [mostRecentEventList](#)
- [bibus](#) » [asynchDetailReportOutput](#) » [outputObjects](#)
- [bibus](#) » [annotation](#) » [outputs](#)

Container Information

Contains instances of the following classes

- [bibus](#) » [graphic](#)
- [bibus](#) » [page](#)

Contained by instances of the following classes

- [bibus](#) » [agentOutputHotList](#)
- [bibus](#) » [reportVersion](#)
- [bibus](#) » [session](#)

Properties

This class has the following properties.

burstID

Identifies the values for the burst keys that produced the output.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 32767 characters
- is searchable

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property was added.

burstKey

Specifies the label for the burst key values that produced the output.

Bursting creates a set of outputs for a report, based on the information that the recipients are interested in and are authorized to view.

For example, using the SDK or the portal, you can burst sales data by branch or geographic location.

Burst report output always has a property value for the [recipients](#) property or the [recipientsEMail](#) property. Depending on the report, values may be assigned to both properties.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 4095 characters

New in Version 8.3 – “Conditional Subscriptions” on page 1899

The documentation for this property was updated.

Related information:

IBM Cognos Administration and Security Guide

context

Specifies the context data for the output.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLMIMEProp`

New in Version 8.3 – “Improved Context Metadata for Selection” on page 1911

This property was added.

contextBlockCount

Specifies the number of context blocks stored in the [context](#) property.

This property

- is of type `int`
is encoded as type `tns:intProp`

New in Version 8.3 – “Improved Context Metadata for Selection” on page 1911

This property was added.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains a URI specifying the location of binary data for this object. The URI can specify a location in the content store or the location of an external repository.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- is read-only

New in Version 10.2.1 – “External object store for report archiving” on page 1829

This property was added.

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

format

Specifies the rendering format.

When used in the `bibus » account` class or the `bibus » contact` class, this property specifies the preferred output format of reports for the account or contact.

When used in the `bibus » documentContent` class or `bibus » output` class this property specifies the format of the data contained in the object

This property

- is of type `bibus » outputFormatEnum`
is encoded as type `tns:nmtokenProp`
- must contain no more than 10 characters
- is searchable

images

Contains the images for the report output.

lastPage

This property is not supported.

This property

- is of type `NMTOKEN`
is encoded as type `tns:nmtokenProp`
- can contain at least 255 characters

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic now accurately describes the contents of this property.

locale

Specifies the locale for this object. The format of the locale follows the Internet Engineering Task Force (IETF) open standard RFC3066.

When used by [bibus » contact](#), this property determines the language and data format of the returned content.

Use the appropriate language so that users understand object names and search paths. Use the appropriate region so that date, time, and currency values are presented in the proper localized format.

This property

- is of type language
 - is encoded as type `tns:languageProp`
- must contain no more than 64 characters
- is searchable

pages

Contains the pages of [XLS](#) output for the report.

New in Version 10.1.0 – “Documentation Updates” on page 1878

This topic was updated to indicate the correct contents of this property.

recipients

Refers to the recipients of the burst report output.

Burst report output is secured by the value of the `recipients` property and by other security features defined in IBM Cognos Analytics, such as the access permissions to objects and capabilities. To access a burst output object, a user must

- be a member of the System Administrators role
- be the owner of the object
- be referenced in the [bibus » output » recipients](#) property for the burst output object
- have `read` permission for the output object

This security filter is also applied when using the “[defaultOutput\(searchPath\)](#)” on page 1571 function and the [bibus » baseReport » defaultOutputFormat](#) property.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

recipientsEMail

Contains the set of email recipients for the burst report output.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`

package

Contains a model and all the folders, queries, reports, views, shortcuts, URLs, and job definitions associated with that model.

Use a package to organize all objects related to a particular application.

A package is created when a modeler publishes a model from IBM Cognos Framework Manager. Users can view the contents of a package in their web portal.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » baseRSSTask » channel](#)
- [bibus » authoredPowerPlay8Report » metadataModelPackage](#)
- [bibus » authoredReport » metadataModelPackage](#)
- [bibus » baseDataIntegrationTask » metadataModelPackage](#)
- [bibus » reportCache » metadataModelPackage](#)
- [bibus » reportVersion » metadataModelPackage](#)
- [bibus » storedProcedureTask » metadataModelPackage](#)
- [bibus » shortcut » target](#)

Container Information

Contains instances of the following classes

- [bibus » baseAgentDefinition](#)
- [bibus » baseDataIntegrationTask](#)
- [bibus » baseDataMovementTask](#)
- [bibus » basePowerPlay8Report](#)
- [bibus » basePowerPlayClass](#)
- [bibus » baseReport](#)
- [bibus » dashboard](#)
- [bibus » dataSource](#)
- [bibus » document](#)
- [bibus » drillPath](#)
- [bibus » folder](#)
- [bibus » jobDefinition](#)
- [bibus » launchable](#)
- [bibus » model](#)
- [bibus » package](#)
- [bibus » packageConfiguration](#)
- [bibus » pagelet](#)
- [bibus » planningApplication](#)
- [bibus » planningTask](#)

- [bibus » shortcut](#)
- [bibus » URL](#)

Contained by instances of the following classes

- [bibus » content](#)
- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)
- [bibus » session](#)

Properties

This class has the following properties.

configuration

Specifies the package configuration.

Configuration data for the package is stored as a contained object. This allows administrators to use a security policy to restrict the ability to change this data. The default security policy for packages allows many roles to update the package.

This property

- must have at most 1 item

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

dataSources

Reserved.

New in Version 8.4 — “[Package Data Sources](#)” on page 1895

This property was added.

defaultPortalAction

Specifies the default action for an object in a portal.

When users click the hyperlinked name in the portal, the default action specified, such as view, run, or open for editing, is performed.

This property

- is of type [bibus » packageActionEnum](#)
is encoded as type `tns:packageActionEnumProp`
- has a default value of `view`

drillPaths

Contains the drill paths. A drill path is used to drill through from a source to a target.

For example, a drill path can specify how to get from one report to another to allow a user to access related or more detailed information.

effectiveUserCapabilities

Contains the current user's capabilities for the object. This value is determined by intersecting the set of user capabilities granted by [bibus » package » userCapabilityPolicies](#) property with the set of user capabilities granted to the user globally, as returned by [bibus » session » userCapabilities](#) property.

In order for a capability to be effective for the object, it must be granted to the user both globally and for the object.

This property

- is an array of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “[Object Capabilities](#)” on page 1885

This property was added.

New in Version 10.1.0 – “[Object Capabilities Properties](#)” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

items

Contains the child objects for this object.

New in Version 8.3 – “[Package Hierarchies](#)” on page 1913

This property was extended to allow instances of the [bibus » package](#) class to be contained by instances of this property.

New in Version 8.4 – “[PowerPlay 8 Integration](#)” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be contained by instances of this property.

New in Version 8.4 – “[Dashboards](#)” on page 1894

This property was extended to allow instances of the [bibus » dashboard](#) class to be contained by instances of this property.

New in Version 9.0.0 – “[Support for IBM Cognos Express](#)” on page 1879

This property was extended to allow instances of the [bibus » launchable](#) class to be contained by instances of this property.

New in Version 10.1.0 – “[Variable Support for Data Movement Tasks](#)” on page 1854

This property was extended to allow instances of the [bibus » dataMovementTaskAlias](#) class to be contained by instances of this property.

model

Specifies a model contained by this package.

Administrators can set a distinct security policy for each model, differing from the security policy set for the package that contains it.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » model](#)

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “[Package Hierarchies](#)” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

planningApplication

Reserved.

This property

- must have at most 1 item

powerPlay8Configuration

Specifies the PowerPlay configuration data for this object.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXMLProp`

New in Version 8.4 – “[PowerPlay 8 Integration](#)” on page 1886

This property was added.

repositoryRules

This property

- is an array of type `bibus » repositoryRule`
is encoded as type `tns:repositoryRuleArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

New in Version 10.2.1 – “Support for archiving a namespace or namespaceFolder” on page 1829

The `bibus » account`, `bibus » namespace`, and `bibus » namespaceFolder` classes have been extended to include this property.

New in Version 10.2.0 – “New Repository Service (REST) API” on page 1843

The Repository Service REST API allows for navigation and retrieval of resources stored in an external repository.

For more information, see the Repository Service API reference chapter.

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

<i>Table 191. Rules for a new package object</i>			
Versioned Class	Maximum Age	Maximum Count	Ordered By
<code>bibus » model class</code>		0	<code>bibus » baseClass » creationTime</code> property

This property

- is an array of type `bibus » retentionRule`
is encoded as type `tns:retentionRuleArrayProp`

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property can now be acquired.

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

userCapabilities

Contains the current user's capabilities for the object, as defined by the [userCapabilityPolicies](#) property.

This property

- is an array of type `bibus » userCapabilityEnum`
 - is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Object Capabilities” on page 1885

This property was added.

New in Version 10.1.0 – “Object Capabilities Properties” on page 1877

This property can now be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required.

userCapabilityPolicies

Contains the user capability policy for the object.

This property

- is an array of type [bibus » userCapabilityPolicy](#)
is encoded as type `tns:userCapabilityPolicyArrayProp`
- can be acquired from a containing object

New in Version 8.4 — “Object Capabilities” on page 1885

This property was added.

userInterfaces

Specifies the user interfaces that can be used with this package. If this property contains no entries, any user interface may be used with this package. If this property contains one or more entries, only the specified user interfaces may be used with this package.

This property

- is an array of type [bibus » uiComponentEnum](#)
is encoded as type `tns:uiComponentEnumArrayProp`
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

packageActionEnumProp

Defines the simple property class for the [bibus » packageActionEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » package » defaultPortalAction](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » packageActionEnum](#)
is encoded as type `tns:packageActionEnum`

packageConfiguration

Defines package-specific configuration information.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus](#) » [package](#)

Properties

This class has the following properties.

crosstabItemDisplayCountDefault

Specifies the default number of data items displayed in a crosstab.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 5

crosstabItemDisplayCountLimit

Specifies the maximum number of data items displayed in a crosstab.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20

defaultAnalysis

Specifies the default analysis object to be loaded by Analysis Studio for the containing package.

If an object is not specified then Analysis Studio will start with a blank screen.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [analysis](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

memberDisplayCountDefault

Specifies the default number of members displayed in one level of the data tree control.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 20

memberDisplayCountLimit

Specifies the maximum number of members displayed in one level of the data tree control.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50

Provides a single page of HTML report output.

The name of the page is the page number.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » cacheOutput](#)
- [bibus » documentContent](#)
- [bibus » output](#)

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property `data`. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

pageDefinition

Describes the physical properties of a page.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » baseReport » executionPageDefinition](#)
- [bibus » account » page](#)

Container Information

Contained by instances of the following classes

- [bibus » configuration](#)

Properties

This class has the following properties.

height

Specifies the page height.

This property

- is of type `decimal`
 - is encoded as type `tns:decimalProp`
- must contain a value greater than 0

unit

Specifies the unit of measurement used to define page dimensions.

This property

- is of type [bibus » linearUnitEnum](#)
 - is encoded as type `tns:nmtokenProp`

width

Specifies the page width.

This property

- is of type `decimal`
 - is encoded as type `tns:decimalProp`
- must contain a value greater than 0

pagelet

Represents a user interface area made up of a set of portlets and pagelets rendered by a layout specification.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » personalization » base](#)
- [bibus » pageletInstance » pagelet](#)
- [bibus » account » portalPages](#)
- [bibus » shortcut » target](#)

Container Information

Contains instances of the following classes

- [bibus » pageletInstance](#)
- [bibus » portletInstance](#)

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » folder](#)
- [bibus » package](#)
- [bibus » pageletFolder](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

items

Specifies the list of pagelets and portlets embedded into the pagelet.

layout

Specifies the layout specification used to render the pagelet.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

metadata

Specifies the metadata for this pagelet.

This metadata includes, but is not limited to, the list of user capabilities required to use this pagelet.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

New in Version 8.3 — “Capabilities Refinements” on page 1918

This property was added.

pageletFolder

Contains the set of pagelets for a portal package.

This class is used for the hierarchical storage of pagelets. The [bibus » pageletFolder](#) name becomes part of the virtual path for accessing pagelets.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » dashboard](#)
- [bibus » pagelet](#)
- [bibus » pageletFolder](#)

Contained by instances of the following classes

- [bibus » pageletFolder](#)
- [bibus » portalPackage](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

items

Contains the set of child objects for this pagelet folder.

New in Version 8.4 – “Dashboards” on page 1894

This property was extended to allow instances of the [bibus » dashboard](#) class to be contained by instances of this property.

pageletInstance

Defines an instance of a pagelet on a rendered page.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » pagelet](#)

Properties

This class has the following properties.

context

Specifies the customizations for the object.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXMLProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)
is encoded as type `tns:deploymentReferenceArrayProp`

New in Version 8.3 – “Package Hierarchies” on page 1913

This property was added.

pagelet

Refers to the [bibus » pagelet](#) object for this instance.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » pagelet](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

pagingOption

Defines the abstract base class for all paging option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » pagingOptionInt](#)

What's new

New in Version 8.3 – “Schedule Management” on page 1907

This class was added.

Properties

This class has the following properties.

name

Identifies the paging option.

This property

- is of type [bibus » pagingOptionEnum](#)
is encoded as type `tns:pagingOptionEnum`

pagingOptionInt

Defines int values for the paging options.

This class

- inherits properties from the [bibus » pagingOption](#) class

What's new

New in Version 8.3 — “[Schedule Management](#)” on page 1907

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the paging option.

This property

- is of type `int`

is encoded as type `xs:int`

parameter

Provides the metadata that describes a parameter.

This class is used for parameters that are defined in IBM Cognos Analytics.

This class

- inherits properties from the [bibus » baseParameter](#) class

Related information:

- [bibus » parameterDataSource](#) class

Properties

This class has the following properties.

capabilities – obsolete

caption – obsolete

defaultValue – obsolete

modelDisplayItem – obsolete

modelFilterItem

Specifies the Framework Manager model item that is filtered using this parameter.

This property

- is of type `string`

is encoded as type `xs:string`

modelUseItem – obsolete

name – obsolete

type – obsolete

useValueExpression

Specifies the expression that determines the use values for the parameter.

This property

- is of type [bibus » metadataModelExpression](#)

is encoded as type `tns:metadataModelExpression`

values – obsolete

parameterAssignmentDataItem

Specifies the assignment of a data item value to a parameter.

This class

- inherits properties from the [bibus » baseParameterAssignment](#) class

Properties

This class has the following properties.

dataItemName

Specifies the name of the metadata model data item name used to determine the parameter value.

This property

- is of type [bibus » metadataModelItemName](#)
is encoded as type `tns:metadataModelItemName`

parameterDataSource

Provides the metadata that describes a data source parameter.

This class is used for parameters that are defined by a data source.

This class

- inherits properties from the [bibus » baseParameter](#) class

Related information:

- [bibus » parameter](#) class

parameterValue

Defines the type for parameter values.

Use the [bibus » parameterValue](#) class to supply values for parameters in query filters. A query filter narrows the results returned by a query, based on an expression that may include parameters.

You can also use the [bibus » parameterValue](#) class to specify credentials to connect to a data source. For example, if the data source is named `Sales`, the [bibus » parameterValue](#) instance must have the `name credential:Sales`. During report prompting, the `name credential:Sales:password` facilitates encryption by the IBM Cognos Analytics gateway, but is changed once prompting is complete.

When the user is prompted to choose a [bibus » dataSourceConnection](#) or [bibus » dataSourceSignon](#) object, the search path of the selected object is stored in the [bibus » parameterValue](#) as part of the credential. If multiple prompts are required to connect to a data source, the current [bibus » parameterValue](#) is annotated with the new information after every prompt, rather than creating additional [bibus » parameterValue](#) instances.

A distinguished value must be specified as part of a [bibus » parameterValue](#)

- to represent the SQL `missing` predicate
- to indicate that filters associated with an optional parameter should be ignored in the query

You can specify a SQL `missing` predicate either by omitting the `use` property for a value item, or by setting the attribute `xsi:nil` to `true` in the `use` property.

You can specify that optional filters should be ignored either by omitting the `value` property of the `bibus » parameterValue`, or by setting the attribute `xsi:nil` to `true` in the `value` property. This functionality may be useful when you are running reports that contain optional filters that you want ignored.

The following examples illustrate how to use the `bibus » parameterValue` class to specify these distinguished values.

Example: Selecting Two Countries

The `bibus » parameterValue` class associates values with a named parameter. In this example, values CA and DE are used to satisfy the parameter named Countries. The strings Canada and Germany are display values that correspond to the query values CA and DE.

The filter expression in the query is

```
<queryItem> = ?Countries?
```

and the condition that is generated is equivalent to

```
<queryItem> = "CA" or <queryItem> = "DE".
```

In a Simple Object Access Protocol (SOAP) request, the `bibus » parameterValue` instance for this example is encoded as follows.

```
<parameterValue    xsi:type="tns:parameterValue"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name    xsi:type="xs:string">Countries</name>
  <value  xsi:type="SOAP-ENC:Array"
    SOAP-ENC:arrayType="tns:parmValueItem[2]">
    <item  xsi:type="tns:simpleParmValueItem">
      <inclusive    xsi:type="xs:boolean">true</inclusive>
      <display     xsi:type="xs:string">Canada</display>
      <use        xsi:type="xs:string">CA</use>
    </item>
    <item  xsi:type="tns:simpleParmValueItem">
      <inclusive    xsi:type="xs:boolean">true</inclusive>
      <display     xsi:type="xs:string">Germany</display>
      <use        xsi:type="xs:string">DE</use>
    </item>
  </value>
</parameterValue>
```

Example: Selecting One Country and SQL Missing Using `xsi:nil`

In this example, the values CA and SQL missing are used to satisfy the parameter named Countries. This example uses `xsi:nil` in the `use` property to indicate SQL missing.

The filter expression in the query is

```
<queryItem> = ?Countries?
```

and the condition that is generated is equivalent to

```
<queryItem> = "CA" or <queryItem> is missing.
```

In a SOAP request, the `bibus » parameterValue` instance for this example is encoded as follows.

```
<parameterValue    xsi:type="tns:parameterValue"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
```

```

xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<name xsi:type="xs:string">Countries</name>
<value xsi:type="SOAP-ENC:Array"
  SOAP-ENC:arrayType="tns:parmValueItem[2]">
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">Canada</display>
    <use xsi:type="xs:string">CA</use>
  </item>
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">No Country</display>
    <use xsi:type="xs:string" xsi:nil="true"/>
  </item>
</value>
</parameterValue>

```

Example: Selecting One Country and SQL Missing by Omitting the Use Property

This example is equivalent to the previous one, but omits the use property rather than using `xsi:nil`.

```

<parameterValue xsi:type="tns:parameterValue"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<name xsi:type="xs:string">Countries</name>
<value xsi:type="SOAP-ENC:Array"
  SOAP-ENC:arrayType="tns:parmValueItem[2]">
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">Canada</display>
    <use xsi:type="xs:string">CA</use>
  </item>
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">No Country</display>
  </item>
</value>
</parameterValue>

```

Example: Providing an Empty Use Property Does Not Specify SQL Missing

If the use property is empty, it defines a zero-length string, not SQL missing. In this example, the empty use property results in the value CA and an empty string satisfying the parameter named Countries, instead of CA and SQL missing.

The filter expression in the query is

```
<queryItem> = ?Countries?
```

and the condition that is generated is equivalent to

```
<queryItem> = "CA" or <queryItem> = "".
```

In a SOAP request, the `bibus » parameterValue` instance for this example is encoded as follows.

```

<parameterValue xsi:type="tns:parameterValue"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

```

```

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<name xsi:type="xs:string">Countries</name>
<value xsi:type="SOAP-ENC:Array"
      SOAP-ENC:arrayType="tns:parmValueItem[2]">
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">Canada</display>
    <use xsi:type="xs:string">CA</use>
  </item>
  <item xsi:type="tns:simpleParmValueItem">
    <inclusive xsi:type="xs:boolean">true</inclusive>
    <display xsi:type="xs:string">No Country</display>
    <!-- Empty String -->
    <use xsi:type="xs:string"/>
  </item>
</value>
</parameterValue>

```

Example: Using xsi:nil to Indicate that Optional Filters Should be Ignored

In this example, `xsi:nil` is used in the `value` property to specify that optional query parameters should be ignored.

```

<parameterValue xsi:type="tns:parameterValue"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name xsi:type="xs:string">Countries</name>
  <value xsi:type="SOAP-ENC:Array"
        SOAP-ENC:arrayType="tns:parmValueItem[0]"
        xsi:nil="true"/>
</parameterValue>

```

Example: Omitting the Value Property to Indicate that Optional Filters Should be Ignored

This example is equivalent to the previous one, but omits the `value` property rather than using `xsi:nil`.

```

<parameterValue xsi:type="tns:parameterValue"
  xmlns:tns="http://developer.cognos.com/schemas/bibus/3/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <name xsi:type="xs:string">Countries</name>
</parameterValue>

```

References

Used by the following properties:

- [bibus » account » parameters](#)
- [bibus » agentTaskDefinition » parameters](#)
- [bibus » archiveDescriptor » parameters](#)
- [bibus » asynchDetailDrillThroughRequest » parameters](#)
- [bibus » asynchDetailParameterValues » parameters](#)
- [bibus » asynchRequest » parameters](#)
- [bibus » asynchSecondaryRequest » parameters](#)

- [bibus » baseAgentDefinition » parameters](#)
- [bibus » baseDataIntegrationTask » parameters](#)
- [bibus » baseDataMovementTask » parameters](#)
- [bibus » basePowerPlay8Report » parameters](#)
- [bibus » baseReport » parameters](#)
- [bibus » baseRSSTask » parameters](#)
- [bibus » contentTask » parameters](#)
- [bibus » drillPath » parameters](#)
- [bibus » exportDeployment » parameters](#)
- [bibus » historyDetailRelatedReports » parameters](#)
- [bibus » historyDetailRequestArguments » parameters](#)
- [bibus » humanTask » parameters](#)
- [bibus » importDeployment » parameters](#)
- [bibus » indexUpdateTask » parameters](#)
- [bibus » jobDefinition » parameters](#)
- [bibus » jobStepDefinition » parameters](#)
- [bibus » memo » parameters](#)
- [bibus » migrationTask » parameters](#)
- [bibus » planningMacroTask » parameters](#)
- [bibus » planningTask » parameters](#)
- [bibus » queryServiceTask » parameters](#)
- [bibus » reportCache » parameters](#)
- [bibus » reportVersion » parameters](#)
- [bibus » schedule » parameters](#)
- [bibus » scheduledEvent » parameters](#)
- [bibus » storedProcedureTask » parameters](#)
- [bibus » webServiceTask » parameters](#)
- [bibus » drillPath » targetParameters](#)
- [bibus » drillOptionParameterValues » value](#)
- [bibus » parameterValueArrayProp » value](#)
- [bibus » parameterValueProp » value](#)

Used by the following method parameters:

- [asynch » run\(objectPath, parameterValues, options\) » parameterValues](#)
- [asynch » runSpecification\(specification, parameterValues, options\) » parameterValues](#)
- [asynch » wait\(conversation, parameterValues, options\) » parameterValues](#)
- [drillThrough » convertDrillThroughContext\(inputContext, parameterValues, options\) » parameterValues](#)
- [drillThrough » findDrillThroughPaths\(objectPath, parameterValues, options\) » parameterValues](#)
- [drillThrough » queryDrillPath\(objectPath, parameterValues, options\) » parameterValues](#)
- [event » runAt\(startTime, objectPath, parameterValues, options\) » parameterValues](#)
- [indexTerm » addTermAssociation\(term, parameterValues, options\) » parameterValues](#)
- [indexTerm » deleteTermAssociation\(term, parameterValues, options\) » parameterValues](#)
- [indexUpdate » add\(objectPath, parameterValues, options\) » parameterValues](#)

- [indexUpdate » delete\(objectPath, parameterValues, options\) » parameterValues](#)
- [indexUpdate » get\(objectPath, parameterValues, options\) » parameterValues](#)
- [paging » currentPage\(conversation, parameterValues, options\) » parameterValues](#)
- [paging » firstPage\(conversation, parameterValues, options\) » parameterValues](#)
- [paging » lastPage\(conversation, parameterValues, options\) » parameterValues](#)
- [paging » nextPage\(conversation, parameterValues, options\) » parameterValues](#)
- [paging » previousPage\(conversation, parameterValues, options\) » parameterValues](#)
- [parameter » collectParameterValues\(objectPath, parameterValues, options\) » parameterValues](#)
- [parameter » collectParameterValuesSpecification\(specification, parameterValues, options\) » parameterValues](#)
- [parameter » getParameters\(objectPath, parameterValues, options\) » parameterValues](#)
- [parameter » getParametersSpecification\(specification, parameterValues, options\) » parameterValues](#)
- [promptPaging » back\(conversation, parameterValues, options\) » parameterValues](#)
- [promptPaging » forward\(conversation, parameterValues, options\) » parameterValues](#)
- [promptPaging » getPromptValues\(conversation, parameterValues, options\) » parameterValues](#)
- [report » deliver\(conversation, parameterValues, options\) » parameterValues](#)
- [report » drill\(conversation, parameterValues, options\) » parameterValues](#)
- [report » getContext\(conversation, parameterValues, options\) » parameterValues](#)
- [report » getObjectContext\(objectPath, parameterValues, options\) » parameterValues](#)
- [report » getOutput\(conversation, parameterValues, options\) » parameterValues](#)
- [report » lineage\(conversation, parameterValues, options\) » parameterValues](#)
- [report » query\(objectPath, parameterValues, options\) » parameterValues](#)
- [report » render\(conversation, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » clearCubeWorkloadLog\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » getCubeState\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » refreshCubeDataCache\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » refreshCubeMemberCache\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » refreshCubeSecurity\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » restartCubes\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » startCubes\(cubeNames, parameterValues, options\) » parameterValues](#)
- [rolapCubeAdministration » stopCubes\(cubeNames, parameterValues, options\) » parameterValues](#)
- [validate » validate\(objectPath, parameterValues, options\) » parameterValues](#)
- [validate » validateSpecification\(specification, parameterValues, options\) » parameterValues](#)

Related information:

- [Chapter 23, “Using report specifications,” on page 1457](#)

Properties

This class has the following properties.

name

Identifies the parameter.

Because the prefix `credential:` is reserved for parameters that represent data sources, do not use a colon (`:`) in a parameter name.

This property

- is of type `token`
is encoded as type `xs:string`

value

Specifies the parameter value.

This property

- is an array of type `bibus » parmValueItem`
is encoded as type `tns:parmValueItemArray`

parameterValueArrayProp

Defines the array property class for the `bibus » parameterValue` class.

This class

- inherits properties from the `bibus » baseProp` class

References

Used by the following properties:

- [bibus » account » parameters](#)
- [bibus » agentTaskDefinition » parameters](#)
- [bibus » baseAgentDefinition » parameters](#)
- [bibus » baseDataIntegrationTask » parameters](#)
- [bibus » baseDataMovementTask » parameters](#)
- [bibus » basePowerPlay8Report » parameters](#)
- [bibus » baseReport » parameters](#)
- [bibus » baseRSSTask » parameters](#)
- [bibus » contentTask » parameters](#)
- [bibus » drillPath » parameters](#)
- [bibus » exportDeployment » parameters](#)
- [bibus » historyDetailRelatedReports » parameters](#)
- [bibus » historyDetailRequestArguments » parameters](#)
- [bibus » humanTask » parameters](#)
- [bibus » importDeployment » parameters](#)
- [bibus » indexUpdateTask » parameters](#)
- [bibus » jobDefinition » parameters](#)
- [bibus » jobStepDefinition » parameters](#)
- [bibus » memo » parameters](#)
- [bibus » migrationTask » parameters](#)
- [bibus » planningMacroTask » parameters](#)
- [bibus » planningTask » parameters](#)

- [bibus » queryServiceTask » parameters](#)
- [bibus » reportCache » parameters](#)
- [bibus » reportVersion » parameters](#)
- [bibus » schedule » parameters](#)
- [bibus » storedProcedureTask » parameters](#)
- [bibus » webServiceTask » parameters](#)
- [bibus » drillPath » targetParameters](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`

parameterValueProp

Defines the simple property class for the [bibus » parameterValue](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValue`

parmValueItem

Defines the abstract base class for parameter value items used by IBM Cognos Analytics.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » boundRangeParmValueItem](#)
- [bibus » hierarchicalParmValueItem](#)
- [bibus » simpleParmValueItem](#)
- [bibus » unboundedEndRangeParmValueItem](#)
- [bibus » unboundedStartRangeParmValueItem](#)

References

Used by the following properties:

- [bibus](#) » [baseParameter](#) » [defaultValue](#)
- [bibus](#) » [parameterValue](#) » [value](#)
- [bibus](#) » [baseParameter](#) » [values](#)

Properties

This class has the following properties.

inclusive

Specifies whether the parameter values are included in the result set.

If `true`, the value should be included in the result set. If `false`, the value should be excluded.

The examples in the following table show the result set for a data set that contains the values 1, 3, 5, and 7. Inclusive range values are indicated with square brackets and exclusive range values with parentheses.

Table 192. Sample result sets based on values of the `inclusive` property

Parameter value	Value of <code>inclusive</code> property	Result set
5	true	5
5	false	1, 3, 7
[1, 5)	true	1, 3
[1, 5)	false	5, 7
(1, 5]	true	3, 5
(1, 5]	false	1, 7
[1, 5]	true	1, 3, 5
[1, 5]	false	7

This property

- is of type `boolean`
- is encoded as type `xs:boolean`

pdfCharacterEncodingEnumProp

Defines the simple property class for the [bibus](#) » [pdfCharacterEncodingEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [batchReportService](#) » [brsPDFCharacterEncoding](#)

- [bibus » configuration » brsPDFCharacterEncoding](#)
- [bibus » configurationFolder » brsPDFCharacterEncoding](#)
- [bibus » dispatcher » brsPDFCharacterEncoding](#)
- [bibus » configuration » rsPDFCharacterEncoding](#)
- [bibus » configurationFolder » rsPDFCharacterEncoding](#)
- [bibus » dispatcher » rsPDFCharacterEncoding](#)
- [bibus » reportService » rsPDFCharacterEncoding](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » pdfCharacterEncodingEnum](#)
is encoded as type `tns:pdfCharacterEncodingEnum`

pdfCompressionTypeEnumProp

Defines the simple property class for the [bibus » pdfCompressionTypeEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » batchReportService » brsPDFCompressionType](#)
- [bibus » configuration » brsPDFCompressionType](#)
- [bibus » configurationFolder » brsPDFCompressionType](#)
- [bibus » dispatcher » brsPDFCompressionType](#)
- [bibus » configuration » rsPDFCompressionType](#)
- [bibus » configurationFolder » rsPDFCompressionType](#)
- [bibus » dispatcher » rsPDFCompressionType](#)
- [bibus » reportService » rsPDFCompressionType](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » pdfCompressionTypeEnum](#)
is encoded as type `tns:pdfCompressionTypeEnum`

pdfFontEmbeddingEnumProp

Defines the simple property class for the [bibus » pdfFontEmbeddingEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » batchReportService » brsPDFEmbedFonts](#)
- [bibus » configuration » brsPDFEmbedFonts](#)
- [bibus » configurationFolder » brsPDFEmbedFonts](#)
- [bibus » dispatcher » brsPDFEmbedFonts](#)
- [bibus » configuration » rsPDFEmbedFonts](#)
- [bibus » configurationFolder » rsPDFEmbedFonts](#)
- [bibus » dispatcher » rsPDFEmbedFonts](#)
- [bibus » reportService » rsPDFEmbedFonts](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » pdfFontEmbeddingEnum](#)
is encoded as type `tns:pdfFontEmbeddingEnum`

pdfOption

Defines the abstract base class for all PDF option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » pdfOptionBoolean](#)
- [bibus » pdfOptionPrintQuality](#)
- [bibus » pdfOptionXMLEncodedXML](#)

What's new

New in Version 8.3 — [“PDF Options - Password Protection” on page 1910](#)

This class was added.

Properties

This class has the following properties.

name

Identifies the PDF option.

This property

- is of type [bibus » pdfOptionEnum](#)
is encoded as type `tns:pdfOptionEnum`

pdfOptionBoolean

Defines boolean values for the PDF options.

This class

- inherits properties from the [bibus » pdfOption](#) class

What's new

New in Version 8.3 – “PDF Options - Password Protection” on page 1910

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PDF option.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

pdfOptionPrintQuality

Defines [bibus » pdfPrintQualityEnum](#) values for the PDF options.

This class

- inherits properties from the [bibus » pdfOption](#) class

What's new

New in Version 8.3 – “PDF Options - Password Protection” on page 1910

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PDF option.

This property

- is of type [bibus » pdfPrintQualityEnum](#)

is encoded as type `tns:pdfPrintQualityEnum`

pdfOptionXMLEncodedXML

Defines [bibus » xmlEncodedXML](#) values for the PDF options.

This class

- inherits properties from the [bibus » pdfOption](#) class

What's new

New in Version 8.3 — “PDF Options - Password Protection” on page 1910

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PDF option.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

periodical

Defines a periodical as a set of documents created by a producer.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » document](#)

Contained by instances of the following classes

- [bibus » transientStateFolder](#)

What's new

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This class was added.

Properties

This class has the following properties.

documents

Contains the periodical documents.

producer

Refers to the producer of the periodical documents.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » baseReport](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

permission

Defines the access permissions for a [bibus » policy](#) object.

References

Used by the following properties:

- [bibus » policy » permissions](#)

Properties

This class has the following properties.

access

Specifies whether the access permission is granted or denied.

This property

- is of type [bibus » accessEnum](#)
is encoded as type `tns:accessEnum`

name

Specifies the name of the access permission, such as `read`, `write`, `execute`, `setPolicy`, or `traverse`.

This property

- is of type `token`
is encoded as type `xs:string`

personalization

Contains information used to personalize an IBM Cognos Workspace workspace.

If saved by a designer, a [bibus » dashboard](#) object contains the [bibus » personalization](#) object. If saved by a user, the user's dashboard [personalization](#) folder, a [bibus » personalizationFolder](#) object contains the [bibus » personalization](#) object. If both such objects exist, the user's personalization information takes precedence and is used by the UI to restore the state of the dashboard when it opens.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » personalizationFolder](#)

What's new

New in Version 8.4 — “Dashboards” on page 1894

This class was added.

Properties

This class has the following properties.

base

Refers to the object on which this personalization is based.

A personalization can exist even when the object it is based on is deleted or moved.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » dashboard](#) or [bibus » pagelet](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

context

Specifies the personalizations for the dashboard.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXMLProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)
 - is encoded as type `tns:deploymentReferenceArrayProp`

personalizationFolder

Contains the personalizations created by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » personalization](#)
- [bibus » personalizationFolder](#)

Contained by instances of the following classes

- [bibus » account](#)
- [bibus » personalizationFolder](#)

What's new

New in Version 8.4 — “Dashboards” on page 1894

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

items

Contains the child objects for this object.

pingReply

Defines the properties of the ping response returned by a dispatcher.

References

Used by the following method return values:

- [dispatcher » ping\(dispatcherPath\) » result](#)

Properties

This class has the following properties.

upTime

Specifies the time, in seconds, since the dispatcher started.

This property

- is of type `int`
 - is encoded as type `xs:int`
- is read-only

version

Identifies the software version of the dispatcher.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

planningAdministrationConsoleService

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

pacsauditLevel

Specifies the auditing level for the planning administration console service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

pacsaConnections

Specifies the maximum number of connections that a process of the planning administration console service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pacsaPeakConnections

Specifies the number of connections that a planning administration console service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

planningAdministrationConsoleServiceSpecification

Reserved.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

References

Used by the following properties:

- [bibus » planningMacroTask » specification](#)
- [bibus » planningAdministrationConsoleServiceSpecificationProp » value](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `planningAdministrationConsoleServiceSpecification` class.

Table 193. Services and methods for the `planningAdministrationConsoleServiceSpecification` class.

Action	Mode	Service	Method
Run	All	planningAdministrationConsoleService	asynch » runSpecification(specification, parameterValues, options)

planningAdministrationConsoleServiceSpecificationProp

Defines the simple property class for the [bibus » planningAdministrationConsoleServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » planningMacroTask » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » planningAdministrationConsoleServiceSpecification](#)

is encoded as type `tns:planningAdministrationConsoleServiceSpecification`

planningApplication

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » package](#)

Properties

This class has the following properties.

applicationGUID

Reserved.

This property

- is of type [bibus » guid](#)

is encoded as type `tns:guidProp`

applicationID

Reserved.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- can contain at least 255 characters

applicationState

Reserved.

This property

- is of type [bibus » planningStateEnum](#)

is encoded as type `tns:planningStateEnumProp`

applicationURL

Reserved.

This property

- is of type `anyURI`

is encoded as type `tns:anyURIProp`

- can contain at least 4095 characters

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

eList

Reserved.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus » baseClass » searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus » baseClass » ancestors](#) property instead.

planningDataService

Defines run-time configuration parameters for the `planningDataService` .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

pdsAuditLevel

Specifies the auditing level for the planning data service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

pdsConnections

Specifies the maximum number of connections that a process of the planning data service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

pdsEListAccessCacheLimit

Specifies the number of seconds that an e-list access rights cache entry can remain in the cache before it must be recalculated.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 3600
- must contain a value greater than or equal to 0

- can be acquired from a containing object

pdsMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsPeakConnections

Specifies the number of connections that a planning data service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsPeakMaximumProcesses

Specifies the maximum number of planning data service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — [“Improve Batch Processing” on page 1902](#)

This property was added.

pdsShowCellAnnotations

Specifies whether authoring studios should display cell annotations.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

planningMacroTask

Defines a planning macro task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `planningMacroTask` class.

Action	Mode	Service	Method
Run	All	planningAdministrationConsoleService	async » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » history](#)
- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
<code>bibus » history</code> class		5	<code>bibus » baseClass » creationTime</code> property

This property

- is an array of type `bibus » retentionRule`
 - is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method. The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

specification

Specifies the specification for the task.

This property

- is of type [bibus » planningAdministrationConsoleServiceSpecification](#)
is encoded as type `tns:planningAdministrationConsoleServiceSpecificationProp`

planningRuntimeService

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

prsAuditLevel

Specifies the auditing level for the planning runtime service.

This property

- is of type [bibus » auditLevelEnum](#)
is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

prsConnections

Specifies the maximum number of connections that a process of the planning runtime service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

prsPeakConnections

Specifies the number of connections that a planning runtime service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “[System Metrics](#)” on page 1908

This property was added.

planningRuntimeServiceSpecification

Reserved.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `planningRuntimeServiceSpecification` class.

Action	Mode	Service	Method
Run	All	planningRuntimeService	asynch » runSpecification(specification, parameterValues, options)

planningStateEnumProp

Defines the simple property class for the [bibus » planningStateEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » planningApplication » applicationState](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » planningStateEnum](#)
is encoded as type `tns:planningStateEnum`

planningTask

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asynchDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)

- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `planningTask` class.

<i>Table 197. Services and methods for the <code>planningTask</code> class.</i>			
Action	Mode	Service	Method
Run	All	planningTaskService	asynch » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus](#) » [history](#)
- [bibus](#) » [schedule](#)

Contained by instances of the following classes

- [bibus](#) » [dashboard](#)
- [bibus](#) » [folder](#)
- [bibus](#) » [package](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

specification

Reserved.

This property

- is of type `bibus » planningTaskServiceSpecification`
 - is encoded as type `tns:planningTaskServiceSpecificationProp`

planningTaskService

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contains instances of the following classes

- `bibus » runTimeState`
- `bibus » systemMetricThresholds`

Contained by instances of the following classes

- `bibus » dispatcher`

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

ptsAuditLevel

Specifies the auditing level for the planning task service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

ptsConnections

Specifies the maximum number of connections that a process of the planning task service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

ptsPeakConnections

Specifies the number of connections that a planning task service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “[Improve Batch Processing](#)” on page 1902

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

planningTaskServiceSpecification

Reserved.

This class

- inherits properties from the [bibus » asyncSpecification](#) class

References

Used by the following properties:

- [bibus » planningTask » specification](#)
- [bibus » planningTaskServiceSpecificationProp » value](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `planningTaskServiceSpecification` class.

<i>Table 199. Services and methods for the <code>planningTaskServiceSpecification</code> class.</i>			
Action	Mode	Service	Method
Run	All	planningTaskService	async » runSpecification(specification, parameterValues, options)

planningTaskServiceSpecificationProp

Defines the simple property class for the [bibus » planningTaskServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » planningTask » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » planningTaskServiceSpecification](#)
is encoded as type `tns:planningTaskServiceSpecification`

policy

Defines a security policy that specifies which accounts, groups, or roles can access the containing object.

References

Used by the following properties:

- [bibus » baseClass » policies](#)
- [bibus » policyArrayProp » value](#)
- [bibus » policyProp » value](#)

Properties

This class has the following properties.

permissions

Specifies the set of permissions that apply to the account, group, or role seeking access to the containing object.

This property

- is an array of type [bibus » permission](#)
is encoded as type `tns:permissionArray`

securityObject

Specifies the account, group, or role for which the security policy is defined.

This property

- is of type [bibus » baseClass](#)
is encoded as type `tns:baseClass`

policyArrayProp

Defines the array property class for the [bibus » policy](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » baseClass » policies](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » policy](#)
is encoded as type `tns:policyArray`

policyProp

Defines the simple property class for the [bibus » policy](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » policy](#)
is encoded as type `tns:policy`

portal

Contains all [bibus » portalPackage](#) class objects.

This object is created when the content store is initialized. This object cannot be deleted by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » portalPackage](#)
- [bibus » portalSkinFolder](#)

Contained by instances of the following classes

- [bibus » root](#)

Properties

This class has the following properties.

items

Contains the [bibus » portalPackage](#) class objects within the [bibus » portal](#) class object.

skinFolder

Contains the [bibus » portalSkinFolder](#) object.

This property

- must have at most 1 item

portalOption

Defines the abstract base class for all portal option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » portalOptionBoolean](#)
- [bibus » portalOptionDisplayMode](#)
- [bibus » portalOptionInt](#)
- [bibus » portalOptionListSeparator](#)
- [bibus » portalOptionSearchPathSingleObject](#)
- [bibus » portalOptionSearchPathSingleObjectArray](#)
- [bibus » portalOptionString](#)
- [bibus » portalOptionXMLEncodedXML](#)

Properties

This class has the following properties.

name

Identifies the portal option.

This property

- is of type [bibus » portalOptionEnum](#)
is encoded as type `tns:portalOptionEnum`

portalOptionBoolean

Defines boolean values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

portalOptionDisplayMode

Defines [bibus » portalDisplayModeEnum](#) values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type [bibus » portalDisplayModeEnum](#)
is encoded as type `tns:portalDisplayModeEnum`

portalOptionInt

Defines integer values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type `int`
is encoded as type `xs:int`

portalOptionListSeparator

Defines [bibus » portalListSeparatorEnum](#) values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type [bibus » portalListSeparatorEnum](#)
is encoded as type `tns:portalListSeparatorEnum`

portalOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

portalOptionSearchPathSingleObjectArray

Defines [bibus » searchPathSingleObject](#) values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the portal option.

This property

- is an array of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

portalOptionString

Defines string values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property

- is of type `string`
is encoded as type `xs:string`

portalOptionXMLEncodedXML

Defines [bibus » xmlEncodedXML](#) values for the portal options.

This class

- inherits properties from the [bibus » portalOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the portal option.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

portalPackage

Contains metadata related to a [bibus » portletProducer](#) and a hierarchy of [bibus » pagelet](#) class objects.

The [bibus » portalPackage](#) serves as the root for a set of [bibus » portlet](#) and [bibus » pagelet](#) objects that can be independently deployed.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » pageletFolder](#)
- [bibus » portletFolder](#)

Contained by instances of the following classes

- [bibus » portal](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

items

This property manages the set of child objects for this package instance.

portalSkin

Defines the properties of an IBM Cognos Analytics portal skin.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » portalSkin » base](#)

Container Information

Contained by instances of the following classes

- [bibus » portalSkinFolder](#)

Properties

This class has the following properties.

base

Refers to the [bibus » portalSkin](#) class object on which this portal skin is based.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » portalSkin](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

New in Version 10.1.0 – “Portal Skin Management” on page 1863

This property was added.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

previewImageLocation

Specifies the location of the portal skin preview image.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

published

Specifies whether the portal skin has been published.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

New in Version 10.1.0 – “Portal Skin Management” on page 1863

This property was added.

resourceLocation

Specifies the location of the portal skin resources folder.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

specification

Specifies the portal skin XML specification located in the content store.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

New in Version 10.1.0 – “Portal Skin Management” on page 1863

This property was added.

portalSkinFolder

Defines the folder class for portal skins.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » portalSkin](#)

Contained by instances of the following classes

- [bibus » portal](#)

Properties

This class has the following properties.

items

Contains the [bibus » portalSkin](#) objects.

portlet

Contains information that defines a relationship with a Web services for Remote Portlets (WSRP) producer.

A portlet instance represents a WSRP portlet that can be embedded in a [bibus » pagelet](#) class object.

A WSRP producer is a presentation Web service that serves as a container of portlets.

The portlets that are available to pagelet authors are determined by the user's access rights to the portlet object.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » portletInstance » portlet](#)

Container Information

Contained by instances of the following classes

- [bibus » portletProducer](#)

Properties

This class has the following properties.

canCustomize

Specifies whether this portlet can be customized in a [bibus » portletInstance](#).

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `true`

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

context

Specifies customization information for this portlet. When this property is nil, the portlet represents a producer offered portlet.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
 - is encoded as type `tns:xmlEncodedXMLProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)
 - is encoded as type `tns:deploymentReferenceArrayProp`

New in Version 8.3 — “[Package Hierarchies](#)” on page 1913

This property was added.

handle

Specifies the portlet handle returned by the [bibus » portletProducer](#) class object. The portlet handle is used to retrieve the portlet description from the containing portlet producer [serviceDescription](#) property.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

portletFolder

Contains the set of [bibus » portletProducer](#) objects for the containing [bibus » portalPackage](#) object.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » portletFolder](#)
- [bibus » portletProducer](#)

Contained by instances of the following classes

- [bibus » portalPackage](#)
- [bibus » portletFolder](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

items

Contains the set of child objects for this portlet folder.

portletInstance

Defines an instance of a [bibus » portlet](#) class object on a rendered page.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » pagelet](#)

Properties

This class has the following properties.

context

Specifies the customizations for the object.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

deploymentReferences

Contains the deployable object references for this object.

This property

- is an array of type [bibus » deploymentReference](#)
is encoded as type `tns:deploymentReferenceArrayProp`

New in Version 8.3 — “Package Hierarchies” on page 1913

This property was added.

portlet

Refers to the [bibus » portlet](#) class object for this instance.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » portlet](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

portletProducer

Contains information that defines a relationship with a Web services for Remote Portlets (WSRP) producer.

A WSRP producer is a presentation web service that serves as a container of [bibus » portlet](#) class objects. The producer service endpoints, registration, and service description are imported into Content Manager and stored as a [bibus » portletProducer](#) class object under the [bibus » portlet](#) class hierarchy.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » portlet](#)

Contained by instances of the following classes

- [bibus » portletFolder](#)

Properties

This class has the following properties.

binding

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

portlets

Contains the set of [bibus » portlet](#) class objects published by the WSRP producer.

registration

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXMLProp`

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point (!)
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

serviceDescription

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

positiveIntegerProp

Defines the simple property class for the `positiveInteger`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » schedule » everyNPeriods](#)
- [bibus » schedule » monthlyAbsoluteDay](#)
- [bibus » configuration » reposCacheObjTTL](#)
- [bibus » configurationFolder » reposCacheObjTTL](#)
- [bibus » dispatcher » reposCacheObjTTL](#)
- [bibus » repositoryService » reposCacheObjTTL](#)
- [bibus » configuration » reposNumObjDisk](#)
- [bibus » configurationFolder » reposNumObjDisk](#)
- [bibus » dispatcher » reposNumObjDisk](#)
- [bibus » repositoryService » reposNumObjDisk](#)
- [bibus » configuration » reposNumObjMem](#)
- [bibus » configurationFolder » reposNumObjMem](#)
- [bibus » dispatcher » reposNumObjMem](#)
- [bibus » repositoryService » reposNumObjMem](#)
- [bibus » configuration » rsQueueLimit](#)
- [bibus » configurationFolder » rsQueueLimit](#)
- [bibus » dispatcher » rsQueueLimit](#)
- [bibus » reportService » rsQueueLimit](#)
- [bibus » baseClass » version](#)
- [bibus » schedule » yearlyAbsoluteDay](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `positiveInteger`
is encoded as type `xs:integer`

powerPlay8Option

Defines the abstract base class for all PowerPlay 8 option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » powerPlay8OptionAnyURI](#)
- [bibus » powerPlay8OptionBoolean](#)
- [bibus » powerPlay8OptionData](#)
- [bibus » powerPlay8OptionLanguageArray](#)
- [bibus » powerPlay8OptionOutputFormat](#)
- [bibus » powerPlay8OptionSaveAs](#)
- [bibus » powerPlay8OptionSearchPathSingleObject](#)
- [bibus » powerPlay8OptionString](#)

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

name

Identifies the PowerPlay 8 option.

This property

- is of type [bibus » powerPlay8OptionEnum](#)
is encoded as type `tns:powerPlay8OptionEnum`

[**powerPlay8OptionAnyURI**](#)

Defines URI values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 10.1.0 – “Storing Temporary Output Objects Outside the Content Store” on page 1857

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type `anyURI`

is encoded as type `xs:string`

powerPlay8OptionBoolean

Defines boolean values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type `boolean`

is encoded as type `xs:boolean`

powerPlay8OptionData

Defines [bibus » powerPlay8DataEnum](#) values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type [bibus » powerPlay8DataEnum](#)

is encoded as type `tns:powerPlay8DataEnum`

powerPlay8OptionLanguageArray

Defines language values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the values for the PowerPlay 8 option.

This property

- is an array of type `language`
is encoded as type `tns:languageArray`

powerPlay8OptionOutputFormat

Defines [bibus » powerPlay8OutputFormatEnum](#) values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type [bibus » powerPlay8OutputFormatEnum](#)
is encoded as type `tns:powerPlay8OutputFormatEnum`

powerPlay8OptionSaveAs

Defines the class for specifying the saveAs PowerPlay8 option.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

objectClass

Specifies the class of the new object.

This property

- is of type [bibus » powerPlay8SaveAsEnum](#)
is encoded as type `tns:powerPlay8SaveAsEnum`

objectName

Specifies the name of the new object.

This property

- is of type `token`
is encoded as type `tns:multilingualTokenArray`
- can contain at least 255 characters
- is multilingual

parentSearchPath

Specifies the search path of the object that will contain the object created as a result of a `saveAs powerPlay8` option for the [bibus » powerPlay8Report](#).

Refers to instances of the [bibus » folder](#) and [bibus » package](#) classes.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

powerPlay8OptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

powerPlay8OptionString

Defines string values for the PowerPlay 8 options.

This class

- inherits properties from the [bibus » powerPlay8Option](#) class

What's new

New in Version 8.4 — “[PowerPlay 8 Integration](#)” on page 1886

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay 8 option.

This property

- is of type `string`
is encoded as type `xs:string`

[powerPlay8OutputFormatEnumArrayProp](#)

Defines the array property class for the [bibus » powerPlay8OutputFormatEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » basePowerPlay8Report » defaultOutputFormat](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » powerPlay8OutputFormatEnum](#)
is encoded as type `tns:powerPlay8OutputFormatEnumArray`

[powerPlay8OutputFormatEnumProp](#)

Defines the simple property class for the [bibus » powerPlay8OutputFormatEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » powerPlay8OutputFormatEnum](#)
is encoded as type `tns:powerPlay8OutputFormatEnum`

powerPlay8Report

Contains information used to define a PowerPlay report.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredPowerPlay8Report](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `powerPlay8Report` class.

<i>Table 200. Services and methods for the powerPlay8Report class.</i>			
Action	Mode	Service	Method
Run	All	powerPlayService	<code>asynch » run(objectPath, parameterValues, options)</code>

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This class was added.

powerPlay8ReportView

Defines the customization of an PowerPlay report.

All properties other than the underlying report specification can be changed in the report view.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » basePowerPlay8Report](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `powerPlay8ReportView` class.

<i>Table 201. Services and methods for the powerPlay8ReportView class.</i>			
Action	Mode	Service	Method
Run	All	powerPlayService	<code>asynch » run(objectPath, parameterValues, options)</code>

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

Properties

This class has the following properties.

base

Refers to the object on which this report view is based.

A report view can exist even when the object it is based on is deleted or moved.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [authoredPowerPlay8Report](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

powerPlayCube

Defines the class that contains the information required to view a PowerPlay cube.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [basePowerPlayClass](#) class

powerPlayDataBlock

Defines a block of encoded PowerPlay data.

References

Used by the following properties:

- [bibus](#) » [basePowerPlayClass](#) » [dataBlocks](#)
- [bibus](#) » [powerPlayDataBlockArrayProp](#) » [value](#)
- [bibus](#) » [powerPlayDataBlockProp](#) » [value](#)

Properties

This class has the following properties.

data

Specifies the data for the block.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

id

Identifies the type of data stored in the block.

This property

- is of type [bibus » powerPlayDataBlockEnum](#)
is encoded as type `tns:powerPlayDataBlockEnum`

powerPlayDataBlockArrayProp

Defines the array property class for the [bibus » powerPlayDataBlock](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » basePowerPlayClass » dataBlocks](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » powerPlayDataBlock](#)
is encoded as type `tns:powerPlayDataBlockArray`

powerPlayDataBlockProp

Defines the simple property class for the [bibus » powerPlayDataBlock](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » powerPlayDataBlock](#)
is encoded as type `tns:powerPlayDataBlock`

powerPlayOption

Defines the type for PowerPlay options.

This class

- is an abstract type used to define properties that are inherited by other types

Derived Classes

- [bibus » powerPlayOptionBoolean](#)

- [bibus](#) » [powerPlayOptionData](#)
- [bibus](#) » [powerPlayOptionOutputFormat](#)

References

Used by the following properties:

- [bibus](#) » [basePowerPlayClass](#) » [options](#)
- [bibus](#) » [powerPlayOptionArrayProp](#) » [value](#)
- [bibus](#) » [powerPlayOptionProp](#) » [value](#)

Properties

This class has the following properties.

name

Identifies the PowerPlay option.

This property

- is of type [bibus](#) » [powerPlayOptionEnum](#)
is encoded as type `tns:powerPlayOptionEnum`

powerPlayOptionArrayProp

Defines the array property class for the [bibus](#) » [powerPlayOption](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [basePowerPlayClass](#) » [options](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [powerPlayOption](#)
is encoded as type `tns:powerPlayOptionArray`

powerPlayOptionBoolean

Use this class to specify boolean PowerPlay options.

This class

- inherits properties from the [bibus](#) » [powerPlayOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

powerPlayOptionData

Use this class to specify PowerPlay data options.

This class

- inherits properties from the [bibus » powerPlayOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay option.

This property

- is of type [bibus » powerPlayDataEnum](#)
is encoded as type `tns:powerPlayDataEnum`

powerPlayOptionOutputFormat

Use this class to specify PowerPlay output format options.

This class

- inherits properties from the [bibus » powerPlayOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the PowerPlay option.

This property

- is of type [bibus » powerPlayOutputFormatEnum](#)
is encoded as type `tns:powerPlayOutputFormatEnum`

powerPlayOptionProp

Defines the simple property class for the [bibus » powerPlayOption](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » powerPlayOption](#)
is encoded as type `tns:powerPlayOption`

powerPlayReport

Defines the class that contains the information required to run an PowerPlay report.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » basePowerPlayClass](#) class

Properties

This class has the following properties.

defaultPortalAction

Specifies the default action for an object in a portal.

When users click the hyperlinked name in the portal, the default action specified, such as view, run, or open for editing, is performed.

This property

- is of type [bibus » powerPlayReportActionEnum](#)
is encoded as type `tns:powerPlayReportActionEnumProp`
- has a default value of `run`

hasPrompts

Specifies whether the report has prompts enabled.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`
- has a default value of `false`

powerPlayReportActionEnumProp

Defines the simple property class for the [bibus » powerPlayReportActionEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » powerPlayReport » defaultPortalAction](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » powerPlayReportActionEnum](#)
is encoded as type `tns:powerPlayReportActionEnum`

powerPlayService

Defines run-time configuration parameters for the [powerPlayService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This class was added.

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

ppsAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

High affinity requests are used to retrieve saved output, and also when a request must go to a particular instance of `powerPlayService` service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsAuditLevel

Specifies the auditing level for the PowerPlay service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsMaximumEMailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the PowerPlay service allows for an email attachment.

Requests to deliver email messages are sent to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – [“New Email Configuration Parameters” on page 1868](#)

This property was added.

ppsNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute a report, and also when a request can be sent to any instance of [powerPlayService](#) service.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This property was added.

ppsPeakAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a PowerPlay service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsPeakNonAffineConnections

Specifies the number of connections that a PowerPlay service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a PowerPlay service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

The initial request to open an IBM Cognos PowerPlay cube or report is handled as a low affinity request. Most subsequent operations on the open cube or report are handled as high affinity requests.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

ppsQueueLimit

Specifies the number of seconds that a request for the PowerPlay service can be queued before it exceeds the timeout period.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

powerPlayServiceReportSpecification

Defines the class for PowerPlay service report specifications.

This class

- inherits properties from the [bibus » powerPlayServiceSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `powerPlayServiceReportSpecification` class.

<i>Table 202. Services and methods for the powerPlayServiceReportSpecification class.</i>			
Action	Mode	Service	Method
Run	All	powerPlayService	<code>asynch » runSpecification(specification, parameterValues, options)</code>

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This class was added.

powerPlayServiceSpecification

Defines the abstract base class for PowerPlay service report specifications.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » asynchSpecification](#) class

Derived Classes

- [bibus » powerPlayServiceReportSpecification](#)

References

Used by the following properties:

- [bibus » authoredPowerPlay8Report » specification](#)

- [bibus » powerPlayServiceSpecificationProp » value](#)

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This class was added.

powerPlayServiceSpecificationProp

Defines the simple property class for the [bibus » powerPlayServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » authoredPowerPlay8Report » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » powerPlayServiceSpecification](#)
is encoded as type `tns:powerPlayServiceSpecification`

presentationService

Defines run-time configuration parameters for the presentation service .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

psAuditLevel

Specifies the auditing level for the presentation service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

printer

Defines printer properties.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contained by instances of the following classes

- [bibus » namespace](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

location

Specifies the physical location of the printer. This description does not uniquely identify a particular printer.

This property

- is of type `string`

is encoded as type `tns:stringProp`

modelName

Specifies the printer model. This description does not uniquely identify a particular printer.

This property

- is of type `string`

is encoded as type `tns:stringProp`

printerAddress

Specifies the network address of the printer.

This property

- is of type `string`

is encoded as type `tns:stringProp`

promptInfo

Defines a template used to build a page that prompts the user.

References

Used by the following properties:

- [bibus](#) » [CAMException](#) » [promptInfo](#)

Properties

This class has the following properties.

captions

Contains the label text associated with the prompt information.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

displayObjects

Contains templates for the controls for the HTML page.

This property

- is an array of type [bibus](#) » [displayObject](#)
is encoded as type `tns:displayObjectArray`

securityBlob

Provides security information, in the form of a binary large object.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

promptOption

Defines the class used to specify the possible prompt choices for this displayed object.

References

Used by the following properties:

- [bibus](#) » [displayObject](#) » [promptOptions](#)

Properties

This class has the following properties.

default

Indicates whether this should be the default choice for the containing control.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

id

Specifies the value to be returned if the option is chosen.

This property

- is of type `string`

is encoded as type `xs:string`

value

Specifies the value that appears in the prompt.

This property

- is of type `string`

is encoded as type `xs:string`

provider

Defines the class to record information about a provider.

References

Used by the following properties:

- [bibus » biBusHeader » providers](#)
- [bibus » tracking » providers](#)

Properties

This class has the following properties.

build

Specifies the version of the provider, in a provider specific format.

This property

- is of type `string`

is encoded as type `xs:string`

name

Specifies the assigned short form identifier of the provider.

This property

- is of type `string`

is encoded as type `xs:string`

patch

Specifies the patch version of the provider.

This property

- is of type `int`

is encoded as type `xs:int`

productCapability

11.1.2

Defines a product feature.

The security policy associated with the product capability determines the availability of the feature.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » productCapability](#)

Contained by instances of the following classes

- [bibus » capability](#)
- [bibus » productCapability](#)
- [bibus » securedFunction](#)

Properties

This class has the following properties.

userCapability

Specifies the user capability represented by this instance.

This property

- is of type `string`

has values that are defined in [bibus » userCapabilityEnum](#)

query

Contains information used to define a report specification.

Instances of this class are usually created in IBM Cognos Query Studio. For example, you create a query to define a quarterly report.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredReport](#) class

References

Used by the following properties:

- [bibus » drillPath » target](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the query class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus](#) » [session](#)

queryCountOptions

Defines the options associated with the [content](#) » [queryCount](#) method.

Properties

This class has the following properties.

permission

Indicates that a fault should be thrown if the caller does not have the specified permission on any object identified by the [content](#) » [queryCount](#) » [searchPath](#) parameter.

This property

- is of type `string`
is encoded as type `xs:string`

queryCountResult

Defines the return type for [content](#) » [queryCount](#) method.

Properties

This class has the following properties.

count

Specifies the number of objects that have the value contained in [value](#) property.

This property

- is of type `int`
is encoded as type `xs:int`

value

Specifies a value for the property identified by [content](#) » [queryCount](#) » [property](#) parameter that was encountered on an object identified by [content](#) » [queryCount](#) » [search](#) parameter.

This property

- is of type `string`
is encoded as type `xs:string`

queryMultipleOptions

Defines the options associated with the [content](#) » [queryMultipleCache\(requests, options\)](#) method.

References

Used by the following method parameters:

- [content](#) » [queryMultipleCache\(requests, options\)](#) » [options](#)

Properties

This class has the following properties.

cacheValidator

Specifies the cache validator value for the queries.

This property

- is of type `string`
is encoded as type `xs:string`

queryMultipleResult

Defines the result structure for the [content](#) » [queryMultipleCache\(requests, options\)](#) method.

References

Used by the following method return values:

- [content](#) » [queryMultipleCache\(requests, options\)](#) » [result](#)

Properties

This class has the following properties.

cacheValidator

Specifies the cache validator value for the queries.

This property

- is of type `string`
is encoded as type `xs:string`

replies

Specifies the results of the queries.

Each element defines the set of outputs for a single request.

This property

- is an array of type [bibus](#) » [queryReply](#)
is encoded as type `tns:queryReplyArray`

queryOptions

Defines the options associated with the [content](#) » [query\(searchPath, properties, sortBy, options\)](#) method.

References

Used by the following properties:

- [bibus](#) » [queryRequest](#) » [options](#)

Used by the following method parameters:

- [content](#) » [query\(searchPath, properties, sortBy, options\)](#) » [options](#)

Properties

This class has the following properties.

dataEncoding

Specifies the encoding used for binary properties in the result set.

This property

- is of type `bibus » encodingEnum`
is encoded as type `tns:encodingEnum`

maxDataBytes

Specifies the maximum number of bytes of data for binary properties in the result set. Use with `skipDataBytes` to partition a large binary property value.

For example, you can use this option to implement PDF byte serving. Note that for values of zero or less, all the data is returned.

This property

- is of type `integer`
is encoded as type `xs:integer`

maxObjects

Specifies the maximum number of objects in the result set. Use with `skipObjects` to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. Note that for values of zero or less, all selected objects are returned.

This property

- is of type `integer`
is encoded as type `xs:integer`

permission

Specifies that a fault will be thrown if the caller does not have the specified permission on any object selected by the `search` parameter of the request.

This property

- is of type `string`
is encoded as type `xs:string`

refProps

Specifies the names of properties to be retrieved for objects referenced by retrieved properties. For example, use to specify that the names and descriptions of the members of a group be returned.

This property

- is an array of type `bibus » refProp`
is encoded as type `tns:refPropArray`

schemaInfo

Specifies whether additional property information is returned with the property values.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

skipDataBytes

Specifies the number of data bytes to be skipped for binary properties in the result set. Use with `maxDataBytes` to partition a large binary property value.

For example, you can use this option to implement PDF byte serving. Use negative values to indicate the number of bytes to skip from the end of the data.

This property

- is of type `integer`
is encoded as type `xs:integer`

skipObjects

Specifies the number of objects to be skipped in the result set. Use with `maxObjects` to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. Use negative values to indicate the number of objects to skip from the end of the result set.

This property

- is of type `integer`
is encoded as type `xs:integer`

queryReply

Defines output parameters for content store queries that accepts multiple requests.

References

Used by the following properties:

- [bibus](#) » [queryMultipleResult](#) » [replies](#)

Used by the following method return values:

- [content](#) » [queryMultiple\(requests\)](#) » [result](#)

Properties

This class has the following properties.

queryResult

Returns the requested objects from the content store.

The `queryResponse` element contains an array of query results. Each query result contains a child element for each selected object, in the order specified by the `sortBy` parameter. The element for each object has a child element for each requested property.

This property

- is an array of type [bibus](#) » [baseClass](#)
is encoded as type `tns:baseClassArray`

queryRequest

Defines the input parameters for content store queries that can accept multiple requests.

References

Used by the following method parameters:

- [content](#) » [queryMultiple\(requests\)](#) » [requests](#)
- [content](#) » [queryMultipleCache\(requests, options\)](#) » [requests](#)

Properties

This class has the following properties.

options

Specifies the options for query methods.

This property

- is of type [bibus » queryOptions](#)

is encoded as type `tns:queryOptions`

properties

Specifies the set of properties to be returned.

If you specify a reference property, properties of the referenced object can also be returned. By default, only the [searchPath](#) property is returned for each referenced object. If the reference property is ID-based, the [storeID](#) property is also returned.

If you specify the [refProps](#) query option, only the [searchPath](#) property is returned for the referenced object. You must then specify additional properties if you want to have them returned for the referenced object.

If the [data](#) property is specified and [dataEncoding](#) is specified as `MIME` or `MIMECompressed`, the data is returned in a MIME attachment. Otherwise, data is returned inline using [base64](#) encoding.

This property

- is an array of type [bibus » propEnum](#)

is encoded as type `tns:propEnumArray`

search

Specifies the search path string so that Content Manager can locate the requested objects. The current user must have read permission for the requested objects.

For more information about specifying a search path, see [Chapter 32, “Search path syntax,” on page 1565](#).

If you use the `expandMembers` function in the search path, the query determines the accounts and contacts that are members of each group specified in the query. For each account, the query returns the properties of the account and a list of the groups that caused the account or contact to be included in the response.

At the end of the returned objects there is a list of the selected objects that contained no account or contact objects because they were not found or could not be read. A `recipientGroups` element is included in these error objects with a `status` element and no `value` element. The `status` element will contain a status of `notFound` or `noRead`.

If you use the `expandMembers` function in the search path, you cannot specify the `sortBy` parameter in the query request, nor can you specify the following query options:

- [skipObjects](#)
- [maxObjects](#)
- [skipDataBytes](#)
- [maxDataBytes](#)

For more information about the `expandMembers` function, see [“expandMembers\(object_set\)” on page 1572](#).

This property

- is of type `anyURI`

is encoded as type `xs:string`

sortBy

Specifies the sort criteria in an array. Each element of the array specifies a property by which the result set will be sorted and specifies the sort direction, either ascending or descending.

The sort order is determined by the locale of the request.

This property

- is an array of type [bibus » sort](#)
is encoded as type `tns:sortArray`

queryService

Defines run-time configuration parameters for the [queryService](#).

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This class was added.

Related information:

- [“Query service” on page 10](#)
- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

qsAdditionalJVMArguments

Specifies additional arguments that control the JVM. The arguments may vary depending on the JVM.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsAuditLevel

Specifies the auditing level for the query service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsDiagnosticsEnabled – deprecated

Specifies whether diagnostics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property is deprecated.

qsDisableQueryPlanCache – deprecated

Specifies whether query plans are cached for possible re-use. A query plan represents a set of transformations applied to query data objects to obtain desired query results.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

New in Version 10.1.1 – “Deprecation of qsDisableQueryPlanCache” on page 1849

This property is deprecated.

qsDisableVerboseGCLogging

Specifies whether garbage collection information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property was added.

qsDumpModelToFile

Specifies whether the model is dumped to a file when a query is run.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsGCPolicy

Specifies the garbage collection policy used to manage JVM heap storage.

Default: [Generational](#)

This property

- is of type `bibus » gcPolicyEnum`
 - is encoded as type `tns:anyURIProp`
- can be acquired from a containing object

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property was added.

qsGenerateCommentsInNativeSQL

Specifies whether comments in native SQL are recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This property was added.

qsIdleConnectionTimeout

Specifies the timeout period, in seconds, for an idle database connection.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 300
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

qsInitialJVMHeapSize

Specifies the initial size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsInitialJVMNurserySize

Specifies the initial nursery size, in MB, of the JVM. A value of zero indicates that the initial nursery size is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 0
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsJVMHeapSizeLimit

Specifies the maximum size, in MB, of the JVM heap.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1024
- must contain a value greater than 0
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsJVMMemorySizeLimit

Specifies the maximum size, in MB, of the JVM nursery. A value of zero indicates that the nursery size limit is system managed.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- must contain a value greater than or equal to `0`
- can be acquired from a containing object

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property was added.

qsManualCubeStart

Specifies whether ROLAP cubes are started manually.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This property was added.

qsMetricsEnabled – deprecated

Specifies whether metrics information is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is deprecated and will be removed in a future version of the product
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – [“Query Service” on page 1855](#)

This property was added.

New in Version 10.2.0 – [“New queryService configuration options” on page 1837](#)

This property is deprecated.

qsMultiDimensionalQuerySizeLimit

Specifies memory size, in MB, allotted to each locally executed multi-dimensional query. A value of `0` means the memory size is limited to available system memory.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- must contain a value greater than or equal to `0`

- can be acquired from a containing object

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

qsQueryExecutionTrace

Specifies whether information tracing the execution of queries is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsQueryPlanningTrace

Specifies whether information tracing the development of query plans is recorded in a log file.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`
- can be acquired from a containing object

New in Version 10.1.0 – “Query Service” on page 1855

This property was added.

qsResultSetCacheQueryTimeThreshold

Specifies the minimum time, in milliseconds, that must elapse during the construction of a result set before it is considered for caching.

Important: The unit of measurement for this configuration parameter is *milliseconds* (1/1000th of a second) whereas other configuration parameters that specify a duration use *seconds*.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 50
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 1000
- can be acquired from a containing object

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This property was added.

qsROLAPCubeAdministrationCommandTimeout

Specifies the time, in seconds, that the `queryService` will wait for internal resources to become available while executing a ROLAP cube administration command. If the specified timeout period has elapsed without resources becoming available, the `queryService` will time out.

Use a value of 0 when you do not want the `queryService` to time out.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `120`
- can be acquired from a containing object

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This property was added.

qsROLAPCubeConfigurations

Defines configuration data for ROLAP cubes.

This property

- is an array of type `bibus » baseROLAPCubeConfiguration`
 - is encoded as type `tns:baseROLAPCubeConfigurationArrayProp`
- can be acquired from a containing object

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This property was added.

qsROLAPMemberCacheAliasRoot

Reserved.

This property

- is of type `token`
 - is encoded as type `tns:tokenProp`
- can be acquired from a containing object

New in Version 10.2.0 – “[IBM Cognos Dynamic Cubes Administration](#)” on page 1836

This property was added.

qsVerboseGCLogLimit

Specifies the maximum number of JVM garbage collection cycles that are logged when `bibus » queryService » qsDisableVerboseGCLogging` is enabled.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1000`
- must contain a value greater than `0`
- can be acquired from a containing object

New in Version 10.2.0 – “[New queryService configuration options](#)” on page 1837

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

queryServiceTask

Defines a query service task.

This class provides a generic task definition for maintenance items related to the query service, such as cache management.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » jobStepDefinition » stepObject](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the queryServiceTask class.

Action	Mode	Service	Method
Run	All	queryService	async » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » schedule](#)

Contained by instances of the following classes

- [bibus » adminFolder](#)

What's new

New in Version 10.1.0 – [“Query Service Administration Task” on page 1869](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the [content » determineRouting\(objectPaths\)](#) method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

schedule

Contains the schedule. If an object is not scheduled, it still contains this property, however this property is empty.

This property

- must have at most 1 item

refProp

Defines the class you can use to retrieve a specified set of properties for any object that is referenced by another object's properties.

The set of properties being retrieved must include a reference to another object for any values to be returned.

References

Used by the following properties:

- [bibus](#) » [queryOptions](#) » [refProps](#)
- [bibus](#) » [contentManagerQueryOptionRefPropArray](#) » [value](#)

Properties

This class has the following properties.

properties

Contains the set of properties to be retrieved for the referenced object.

This property

- is an array of type [bibus](#) » [propEnum](#)
 - is encoded as type `tns:propEnumArray`

refPropName

Specifies the property that references the object.

This property

- is of type [bibus » propEnum](#)
is encoded as type `tns:propEnum`

relationalMetadataService

Defines run-time configuration parameters for the [relationalMetadataService](#).

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This class was added.

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

rmdsAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `1`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

`rmdsAuditLevel`

Specifies the auditing level for the relational metadata service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

`rmdsConnections`

Specifies the maximum number of connections that a process of the relational metadata service can use concurrently to execute requests.

For more information, see [“Tune Server Performance” on page 89](#).

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `4`
- must contain a value greater than or equal to `1`
- can be acquired from a containing object

New in Version 10.2.0 – “Relational metadata service” on page 1835

This property was added.

`rmdsExecutionTimeLimit`

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`

is encoded as type `tns:intProp`

- has a default value of 0
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmidsNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Writer comment

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmidsPeakAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a relational metadata service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 1
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This property was added.

rmdsPeakConnections

Specifies the number of connections that a relational metadata service process can use to execute requests during peak hours. This property limits the number of these requests that can be executed concurrently by a process.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Relational metadata service” on page 1835

This property was added.

rmdsPeakNonAffineConnections

Specifies the number of connections that a relational metadata service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a relational metadata service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 4
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

relationalMetadataServiceSpecification

Defines the class for relational metadata service specifications.

This class

- inherits properties from the [bibus](#) » [asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the relationalMetadataServiceSpecification class.

Action	Mode	Service	Method
Run	All	relationalMetadataService	asynch » runSpecification(specification, parameterValues, options)

report

Contains information used to define a report specification.

Instances of this class are usually created by IBM Cognos Analytics - Reporting and IBM Cognos Event Studio.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [authoredReport](#) class

References

Used by the following properties:

- [bibus](#) » [historyDetailRelatedReports](#) » [linkPaths](#)
- [bibus](#) » [drillPath](#) » [target](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the report class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus » agentDefinition](#)
- [bibus » session](#)

reportCache

Contains the information required to improve report processing, such as cached prompt values.

Instances of this class are created when a [bibus » baseReport](#) is run using the [savePromptCache](#) run option with a value of `true`.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » cacheOutput](#)

Contained by instances of the following classes

- [bibus » baseReport](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`

is encoded as type `tns:dateTimeProp`

- is read-only
- is searchable

New in Version 8.4 — “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » model`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » package`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type `bibus » option`
 - is encoded as type `tns:optionArrayProp`

output

Contains the cache data for the report.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

specification

Specifies the information required to execute the report. This property is the definition for the report.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

reportDataService

Defines run-time configuration parameters for the report data service .

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

- can be acquired from a containing object

rdsAuditLevel

Specifies the auditing level for the report data service.

This property

- is of type [bibus](#) » [auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

rdsGatewayMappings

Contains the mappings between the internal and external PowerPlay gateways.

This property

- is an array of type [bibus](#) » [gatewayMapping](#)
 - is encoded as type `tns:gatewayMappingArrayProp`
- can be acquired from a containing object

rdsMaximumDataSize

Specifies the maximum amount of data that can be read from a content provider in MB.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 10
- must contain a value greater than or equal to 1
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type [bibus](#) » [runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 — “System Metrics” on page 1908

This property was added.

reportDataServiceAgentDefinition

Defines an agent as a series of tasks and defines parameters for those tasks.

Events are determined using a report data service query.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredAgentDefinition](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `reportDataServiceAgentDefinition` class.

Action	Mode	Service	Method
Run	All	agentService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus » subscriptionFolder](#)

What's new

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This class was added.

Properties

This class has the following properties.

report

Specifies the report data specification used to determine the agent events.

This property

- is of type [bibus » reportDataServiceSpecification](#)

is encoded as type `tns:reportDataServiceSpecificationProp`

reportDataServiceSpecification

Defines the type for report data service specifications.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

References

Used by the following properties:

- [bibus » reportDataServiceAgentDefinition » report](#)
- [bibus » reportDataServiceSpecificationProp » value](#)

What's new

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This class was added.

reportDataServiceSpecificationProp

Defines the simple property class for the [bibus » reportDataServiceSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » reportDataServiceAgentDefinition » report](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » reportDataServiceSpecification](#)
is encoded as type `tns:reportDataServiceSpecification`

reportService

Defines run-time configuration parameters for the [reportService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

rsAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during non-peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsAuditLevel

Specifies the auditing level for the report service.

This property

- is of type `bibus » auditLevelEnum`
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

rsAuditNativeQuery

Specifies whether to log native query information for the report service.

This property

- is of type `boolean`

is encoded as type `tns:booleanProp`

- has a default value of `false`
- can be acquired from a containing object

rsChartHotspotLimit

Specifies the default maximum number of hotspots generated in a chart when the report specification does not specify the number of hotspots.

Use a value of `0` to disable the generation of chart hotspots.

Use a `nil` value to generate all hotspots in a chart.

Setting a non-`nil` value improves performance when generating charts that contain large numbers of members.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- can be acquired from a containing object

New in Version 8.3 — “Chart Hotspot Limit” on page 1909

This property was added.

rsDataSourceChange

Specifies the change time of the data source for the report service, in Coordinated Universal Time (UTC).

When the service detects that this property has been changed, it will close all connections to data sources. Subsequent requests will cause new connections to the data sources to be established. This property can be used when a new version of a PowerCube is ready to be used in the production environment.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- can be acquired from a containing object

rsExecutionTimeLimit

Specifies the maximum number of seconds that a task can run before being canceled by the dispatcher.

The dispatcher logs an error (DPR-ERR-2087) indicating that the task execution was canceled due to the execution time limit set being exceeded. A secondary request made against a conversation that has exceeded the time limit returns an error message.

Use a value of `0` when you want to allow the task to complete execution, regardless of the amount of time necessary.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of `0`
- can be acquired from a containing object

rsMaximumEmailAttachmentSize

Specifies the maximum amount of uncompressed data in MB that the report service allows for an email attachment.

Requests to deliver email messages are sent to the `deliveryService` service. If very large email attachments are causing memory errors on the server that hosts the `deliveryService` service, the value of this setting may need to be adjusted. If an email attachment that a request generates exceeds the specified amount, the attachment is removed and replaced with an error message in the body of the email.

Use a value of 0 to specify that the size of an attachment is unlimited.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 15
- must contain a value greater than or equal to 0
- can be acquired from a containing object

New in Version 10.1.0 – “New Email Configuration Parameters” on page 1868

This property was added.

rsMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during non-peak hours.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

rsNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during non-peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsPDFCharacterEncoding

Specifies the character encoding for PDF documents created by the report service.

This property

- is of type [bibus » pdfCharacterEncodingEnum](#)
 - is encoded as type `tns:pdfCharacterEncodingEnumProp`
- has a default value of [auto](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

rsPDFCompressionLevel

Specifies the compression level for PDF documents created by the report service.

A higher value indicates that more compression is applied. This increases the time required to create and to open the PDF documents. The value 0 means no compression, while the value 9 means maximum compression.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 9
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 9
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » reportService » rsPDFCompressionType](#) property

rsPDFCompressionType

Specifies the PDF compression type for PDF documents created by the report service.

This property

- is of type [bibus » pdfCompressionTypeEnum](#)
 - is encoded as type `tns:pdfCompressionTypeEnumProp`
- has a default value of [classic](#)
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

Related information:

[bibus » reportService » rsPDFCompressionLevel](#) property

rsPDFEmbedFonts

Specifies whether the report service should embed fonts in generated PDF documents.

This property

- is of type [bibus » pdfFontEmbeddingEnum](#)
 - is encoded as type `tns:pdfFontEmbeddingEnumProp`

- has a default value of `allow`
- can be acquired from a containing object

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This property was added.

rsPeakAffineConnections

Specifies the number of connections that a report service process can use to execute high affinity requests during peak hours. This property limits the number of these requests that can be executed concurrently by a report service process.

High affinity requests are requests that are closely associated with a particular process. These requests are usually executed faster than low affinity requests.

If the request is sent to a different process, that process usually requires more time to execute the request because it must perform all the activities performed by the process that received the previous request.

For example, when requesting the next page of report output, the request can be executed most efficiently by the process that executed the previous request in the conversation.

High affinity requests are used to save report output, print report output, send report output by email, move between pages of report output, and move forward and backward in prompt pages.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 — “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 2.

rsPeakMaximumProcesses

Specifies the maximum number of report service processes that can be started by a dispatcher during peak hours.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 2
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 — “Improve Batch Processing” on page 1902

This property was added.

rsPeakNonAffineConnections

Specifies the number of connections that a report service process can use to execute low affinity requests during peak hours. This property limits the number of low affinity requests that can be executed concurrently by a report service process.

Low affinity requests establish the context for requests that may follow by caching information. Low affinity requests usually take longer to execute than subsequent high affinity requests. There are no benefits to sending low affinity requests to a particular process because these requests do not use cached information.

Low affinity requests are used to execute reports, extract parameter information from a report, obtain parameter values for a report, retrieve metadata, test data source connections, and validate report specifications.

This property

- is of type `int`
 - is encoded as type `tns:intProp`
- has a default value of 8
- must contain a value greater than or equal to 1
- can be acquired from a containing object

New in Version 8.3 – “Improve Batch Processing” on page 1902

This property was added.

New in Version 10.2.0 – “Updated default settings for Report Service and Batch Report Service” on page 1842

Changing default value to 8.

rsQueueLimit

Specifies the number of seconds that a request for the report service can be queued before it exceeds the timeout period.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 240
- must contain a value greater than or equal to 5
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

reportServiceAnalysisSpecification

Defines the class for report service analysis specifications.

This class

- inherits properties from the [bibus » reportServiceSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the reportServiceAnalysisSpecification class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » runSpecification(specification, parameterValues, options)
Run	Interactive	reportService	asynch » runSpecification(specification, parameterValues, options)

reportServiceDrillThroughSpecification

Defines the class for report service drill-through specifications.

A drill-through specification specifies the information required to perform a drill-through operation between two resources, including information about the target and the action to be performed on the target. The specification may also include rules for assigning data values from the source to parameters defined by the target.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

References

Used by the following properties:

- [bibus » drillPath » specification](#)
- [bibus » reportServiceDrillThroughSpecificationProp » value](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the reportServiceDrillThroughSpecification class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » runSpecification(specification, parameterValues, options)
Run	Interactive	reportService	asynch » runSpecification(specification, parameterValues, options)

What's new

New in Version 8.4 – [“Dynamic Filtering of Report Data” on page 1889](#)

This class was added.

New in Version 8.4 – [“Supporting New Drill-through Targets” on page 1890](#)

This class was added.

reportServiceDrillThroughSpecificationProp

Defines the simple property class for the [bibus » reportServiceDrillThroughSpecification](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » drillPath » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » reportServiceDrillThroughSpecification](#)
is encoded as type `tns:reportServiceDrillThroughSpecification`

reportServiceInteractiveReportSpecification

Defines the class for report service IBM Cognos Active Report specifications.

An IBM Cognos Active Report allows users to work offline with a local copy of their data. They can explore and analyze data, make changes to their reports, and filter and sort data while disconnected from the network. After they re-connect, they can synchronize their changes with the live data source.

The [reportService](#) service and [batchReportService](#) service identify an IBM Cognos Active Report type by the object type specified in a request. If the report is persisted in the content store, the object type is [bibus » interactiveReport](#) class and the [asynch » run\(objectPath, parameterValues, options\)](#) method initiates the request. If the report is not persisted, the object type is [bibus » reportServiceInteractiveReportSpecification](#) class and the [asynch » runSpecification\(specification, parameterValues, options\)](#) method initiates the request.

An IBM Cognos Active Report is identified in the report specification by the attribute value of `application="true"` for the root `report` element. For more information about report specifications, see [Chapter 23, “Using report specifications,” on page 1457](#).

See the *IBM Cognos Analytics - Reporting User Guide* for more information.

This class

- inherits properties from the [bibus » reportServiceSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `reportServiceInteractiveReportSpecification` class.

<i>Table 210. Services and methods for the <code>reportServiceInteractiveReportSpecification</code> class.</i>			
Action	Mode	Service	Method
Run	Batch	batchReportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>
Run	Interactive	reportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>

What's new

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This class was added.

reportServiceMetadataSpecification

Defines the type for report service metadata request specifications.

This class

- inherits properties from the [bibus » asynchSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `reportServiceMetadataSpecification` class.

<i>Table 211. Services and methods for the <code>reportServiceMetadataSpecification</code> class.</i>			
Action	Mode	Service	Method
Run	Batch	batchReportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>
Run	Interactive	reportService	<code>asynch » runSpecification(specification, parameterValues, options)</code>

Related information:

- Chapter 35, “Metadata schema reference,” on page 1765

reportServiceQueryDrillPathOption

Defines the abstract base class for all report service query drill path option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » reportServiceQueryDrillPathOptionBoolean](#)

What's new

New in Version 8.4 – [“Supporting New Drill-through Targets” on page 1890](#)

This class was added.

Properties

This class has the following properties.

name

Identifies the report service query drill path option.

This property

- is of type [bibus » reportServiceQueryDrillPathOptionEnum](#)
is encoded as type `tns:reportServiceQueryDrillPathOptionEnum`

reportServiceQueryDrillPathOptionBoolean

Defines boolean values for the report service query drill path options.

This class

- inherits properties from the [bibus » reportServiceQueryDrillPathOption](#) class

What's new

New in Version 8.4 – [“Supporting New Drill-through Targets” on page 1890](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the report service query drill path option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

reportServiceQueryOption

Defines the abstract base class for all report service query option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » reportServiceQueryOptionBoolean](#)
- [bibus » reportServiceQueryOptionSpecificationFormat](#)

Properties

This class has the following properties.

name

Identifies the report service query option.

This property

- is of type [bibus » reportServiceQueryOptionEnum](#)
is encoded as type `tns:reportServiceQueryOptionEnum`

reportServiceQueryOptionBoolean

Defines boolean values for the report service query options.

This class

- inherits properties from the [bibus » reportServiceQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the report service query option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

reportServiceQueryOptionSpecificationFormat

Defines [bibus » specificationFormatEnum](#) values for the report service query options.

This class

- inherits properties from the [bibus » reportServiceQueryOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the report service query option.

This property

- is of type [bibus » specificationFormatEnum](#)
is encoded as type `tns:specificationFormatEnum`

reportServiceQuerySpecification

Defines the class for report service query specifications.

This class

- inherits properties from the [bibus » reportServiceSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `reportServiceQuerySpecification` class.

Table 212. Services and methods for the reportServiceQuerySpecification class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » runSpecification(specification, parameterValues, options)
Run	Interactive	reportService	asynch » runSpecification(specification, parameterValues, options)

reportServiceReportSpecification

Defines the class for report service report specifications.

This class

- inherits properties from the [bibus » reportServiceSpecification](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the reportServiceReportSpecification class.

Table 213. Services and methods for the reportServiceReportSpecification class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » runSpecification(specification, parameterValues, options)
Run	Interactive	reportService	asynch » runSpecification(specification, parameterValues, options)

Related information:

- Chapter 23, “Using report specifications,” on page 1457

reportServiceSpecification

Defines the abstract base class for [reportService](#) report specifications.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » asynchSpecification](#) class

Derived Classes

- [bibus » reportServiceAnalysisSpecification](#)
- [bibus » reportServiceInteractiveReportSpecification](#)
- [bibus » reportServiceQuerySpecification](#)
- [bibus » reportServiceReportSpecification](#)

reportStudioOption - deprecated

Defines the abstract base class for all Reporting option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » reportStudioOptionSearchPathSingleObject](#)

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This class is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This class was added.

Properties

This class has the following properties.

name

Identifies the Reporting option.

This property

- is of type [bibus » reportStudioOptionEnum](#)
is encoded as type `tns:reportStudioOptionEnum`

[reportStudioOptionSearchPathSingleObject - deprecated](#)

Defines [bibus » searchPathSingleObject](#) values for the Reporting options.

This class

- inherits properties from the [bibus » reportStudioOption](#) class

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This class is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the Reporting option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

[reportTemplate](#)

Contains information used to define an IBM Cognos Analytics report template.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » authoredReport](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the reportTemplate class.

<i>Table 214. Services and methods for the reportTemplate class.</i>			
Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

reportVersion

Contains the information required to reproduce report output, such as prompt values and output formats.

For example, a quarterly sales report runs on the last day of the quarter. Because sales data for the quarter continues to come in after the last day of the quarter, the report must be run again to include the new data. The reportVersion object contains the information used to produce the original output. Therefore, the user does not have to specify this information the next time the report runs.

This class also contains the rendered report output. Saving rendered output can reduce the load on application and database servers, because the report need not be run every time a user wants to view the output. The reduced load makes more resources available for other requirements, such as ad-hoc reporting jobs.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » historyDetailReportService » output](#)
- [bibus » asynchDetailReportOutput » outputObjects](#)

Container Information

Contains instances of the following classes

- [bibus » annotationFolder](#)
- [bibus » output](#)
- [bibus » SQL](#)

Contained by instances of the following classes

- [bibus » basePowerPlay8Report](#)

- [bibus](#) » [baseReport](#)

Properties

This class has the following properties.

allowAnnotations

Specifies whether the outputs in this report version can be annotated by report consumers.

The value of this property is normally determined by the value of the [allowAnnotations](#) option when the report version was created.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- has a default value of `false`

New in Version 8.4 — “[Report Output Annotations](#)” on page 1895

This property was added.

annotationFolder

Specifies the annotation folder for this report version.

This property

- must have at most 1 item

New in Version 8.4 — “[Report Output Annotations](#)” on page 1895

This property was added.

asOfTime

Specifies the end of the period used to filter data for this report version, in Coordinated Universal Time (UTC). For example, you can specify the date and time of the end of the quarter for your quarterly inventory report.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is searchable

canBurst

Specifies whether the report output can be burst.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)
 - has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable

New in Version 8.4 — “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » model`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » package`
 - is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArrayProp`

output

Contains the output for this report version.

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

serverGroup – deprecated

Specifies the server group associated with this object. This property is not used by IBM Cognos Analytics.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is deprecated and will be removed in a future version of the product
- can contain at least 255 characters
- can be acquired from a containing object

specification

Specifies the information required to execute the report. This property is the definition for the report.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

SQL

Contains the SQL for this report version.

reportView

Defines the customization of a query or report.

All properties other than the underlying report specification can be changed in the report view.

Important: The [bibus » baseReport » runAsOwner](#) property is ignored for this class. It is valid only when used with an instance of [bibus » authoredReport](#) class.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseReport](#) class

References

Used by the following properties:

- [bibus » drillPath » target](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the reportView class.

Action	Mode	Service	Method
Run	Batch	batchReportService	asynch » run(objectPath, parameterValues, options)
Run	Interactive	reportService	asynch » run(objectPath, parameterValues, options)

Container Information

Contained by instances of the following classes

- [bibus » agentDefinitionView](#)
- [bibus » agentState](#)
- [bibus » session](#)

Properties

This class has the following properties.

base

Refers to the object on which this report view is based.

A report view can exist even when the object it is based on is deleted or moved. While you cannot run a report view that does not have an associated report or query, the report output is still available.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » authoredReport](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

repositoryRule

Specifies a rule that determines whether contained objects are archived to an external repository.

Content marked for archival is copied to an external repository, if enabled, when an object is first created, or when an administrator runs a [cleanupContent](#) value task.

References

Used by the following properties:

- [bibus » account » repositoryRules](#)
- [bibus » content » repositoryRules](#)
- [bibus » folder » repositoryRules](#)
- [bibus » namespace » repositoryRules](#)
- [bibus » namespaceFolder » repositoryRules](#)
- [bibus » package » repositoryRules](#)
- [bibus » repositoryRuleArrayProp » value](#)
- [bibus » repositoryRuleProp » value](#)

What's new

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This class was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

Related information:

- *IBM Cognos Administration and Security Guide*

Properties

This class has the following properties.

dataSourceName

Specifies the name of the [bibus » dataSource](#) object that describes the external repository.

This property is also used to access content stored in the repository.

This property

- is of type token
is encoded as type `xs:string`

objectClass

Specifies the object class to which this repository rule applies.

This property

- is of type [bibus » classEnum](#)
is encoded as type `tns:classEnum`

repositoryRuleArrayProp

Defines the array property class for the [bibus » repositoryRule](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » account » repositoryRules](#)
- [bibus » content » repositoryRules](#)
- [bibus » folder » repositoryRules](#)
- [bibus » namespace » repositoryRules](#)
- [bibus » namespaceFolder » repositoryRules](#)
- [bibus » package » repositoryRules](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » repositoryRule](#)
is encoded as type `tns:repositoryRuleArray`

repositoryRuleProp

Defines the simple property class for the [bibus » repositoryRule](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » repositoryRule](#)
is encoded as type `tns:repositoryRule`

repositoryService

Reserved.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This class was added.

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

reposAuditLevel

Specifies the auditing level for the repository service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposCacheObjTTL

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 1200
- can be acquired from a containing object

New in Version 10.2.0 – [“Repository service” on page 1840](#)

This property was added.

reposNumObjDisk

Reserved.

This property

- is of type `positiveInteger`

is encoded as type `tns:positiveIntegerProp`

- has a default value of 1000
- can be acquired from a containing object

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

reposNumObjMem

Reserved.

This property

- is of type `positiveInteger`
 - is encoded as type `tns:positiveIntegerProp`
- has a default value of 100
- can be acquired from a containing object

New in Version 10.2.0 – “Repository service” on page 1840

This property was added.

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

resource

Defines a class that contains information required to reference a resource not managed by the content store.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contained by instances of the following classes

- `bibus » namespace`
- `bibus » namespaceFolder`

What's new

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)

has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

resourceType

Reserved.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is searchable

specification

Specifies the information required to reference the resource.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type [bibus » xmlEncodedXML](#)

is encoded as type `tns:xmlEncodedXMLMIMEProp`

retentionRule

Defines the number of versions or the duration for which to keep versioned objects.

Rules assigned to the parent apply to its children.

References

Used by the following properties:

- [bibus » agentState » retentions](#)
- [bibus » baseAgentDefinition » retentions](#)
- [bibus » baseDataIntegrationTask » retentions](#)
- [bibus » basePowerPlay8Report » retentions](#)
- [bibus » baseReport » retentions](#)
- [bibus » baseRSSTask » retentions](#)
- [bibus » contentTask » retentions](#)
- [bibus » document » retentions](#)
- [bibus » exportDeployment » retentions](#)
- [bibus » humanTask » retentions](#)
- [bibus » importDeployment » retentions](#)
- [bibus » indexUpdateTask » retentions](#)
- [bibus » jobDefinition » retentions](#)
- [bibus » memo » retentions](#)
- [bibus » migrationTask » retentions](#)
- [bibus » mruFolder » retentions](#)
- [bibus » package » retentions](#)
- [bibus » planningMacroTask » retentions](#)
- [bibus » planningTask » retentions](#)
- [bibus » storedProcedureTask » retentions](#)
- [bibus » webServiceTask » retentions](#)
- [bibus » contentTaskOptionRetentionRuleArray » value](#)
- [bibus » retentionRuleArrayProp » value](#)
- [bibus » retentionRuleProp » value](#)

Properties

This class has the following properties.

maxDuration

Specifies the maximum age of objects. Values must be specified using the World Wide Web Consortium (W3C) lexicon for duration (described at <http://www.w3.org/>), which is based on the ISO 8601 standard.

For example, to retain objects for 30 days, set the maxDuration value to P30D.

This property

- is of type `duration`
 - is encoded as type `xs:string`

maxObjects

Specifies the maximum number of objects to be kept. If 0 is specified, no maximum is applied to the number of objects stored.

This property

- is of type `integer`

is encoded as type `xs:integer`

objectClass

Specifies the object class to which this retention rule applies.

This property

- is of type [bibus » classEnum](#)

is encoded as type `tns:classEnum`

prop

Specifies the property used to order the versioned objects.

This property

- is of type [bibus » propEnum](#)

is encoded as type `tns:propEnum`

retentionRuleArrayProp

Defines the array property class for the [bibus » retentionRule](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » agentState » retentions](#)
- [bibus » baseAgentDefinition » retentions](#)
- [bibus » baseDataIntegrationTask » retentions](#)
- [bibus » basePowerPlay8Report » retentions](#)
- [bibus » baseReport » retentions](#)
- [bibus » baseRSSTask » retentions](#)
- [bibus » contentTask » retentions](#)
- [bibus » document » retentions](#)
- [bibus » exportDeployment » retentions](#)
- [bibus » humanTask » retentions](#)
- [bibus » importDeployment » retentions](#)
- [bibus » indexUpdateTask » retentions](#)
- [bibus » jobDefinition » retentions](#)
- [bibus » memo » retentions](#)
- [bibus » migrationTask » retentions](#)
- [bibus » mruFolder » retentions](#)
- [bibus » package » retentions](#)
- [bibus » planningMacroTask » retentions](#)
- [bibus » planningTask » retentions](#)
- [bibus » storedProcedureTask » retentions](#)
- [bibus » webServiceTask » retentions](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRuleArray`

retentionRuleProp

Defines the simple property class for the [bibus » retentionRule](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » retentionRule](#)
is encoded as type `tns:retentionRule`

rolapCubeConfiguration

Defines the configuration data for a ROLAP cube.

This class

- inherits properties from the [bibus » baseROLAPCubeConfiguration](#) class

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This class was added.

Properties

This class has the following properties.

aggregateCacheSize

Specifies the cache size, in MB, to be used for in-memory aggregates. A value of 0 means in-memory aggregate caching is disabled.

This property

- is of type `int`
is encoded as type `xs:int`
- has a default value of 0
- must contain a value greater than or equal to 0

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

aggregateLoggingEnabled

Specifies whether aggregate logging is enabled for the ROLAP cube.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

automaticAggregateOptimizationEnabled

Specifies whether in-memory aggregates are automatically optimized.

You must explicitly set the value of this property if you create it in your SDK application. Otherwise, an error will occur when you run the application.

This property

- is of type `bibus » automaticAggregateOptimizationEnum`
 - is encoded as type `tns:automaticAggregateOptimizationEnum`

New in Version 10.2.2 – New dynamic cube configuration properties

This property was added.

disableExternalAggregates

Specifies whether external aggregate support is enabled.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

enableDynamicMemberCaching

Reserved.

This property

- is of type `boolean`
 - is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Administration” on page 1836

This property was added.

maxAggregateLoadThreads

Specifies the maximum number of in-memory aggregates to load in parallel. A value of 0 sets the number of threads to be twice the number of CPU cores.

This property

- is of type `integer`
is encoded as type `xs:int`
- has a default value of 0

New in Version 10.2.2 – New dynamic cube configuration properties

This property was added.

maxHierarchyLoadThreads

Specifies the maximum number of hierarchies to load in parallel for cube start and member cache refresh. A value of 0 sets the number of threads to be twice the number of CPU cores.

This property

- is of type `integer`
is encoded as type `xs:int`
- has a default value of 0

New in Version 10.2.2 – New dynamic cube configuration properties

This property was added.

measuresThreshold

Specifies the percentage of measures to retrieve from a dynamic cube. Calculated measures, non-visible measures, and semi-aggregate measures are not included. If the percentage of measures retrieved is greater than specified here, the generated SQL query retrieves all measures.

This value must be between 0 and 100. Specify 0 if all measures for a set of levels are required for queries. Specify 100 if only a few measures are required for queries. For example, when using only predefined reports.

This property

- is of type `integer`
is encoded as type `xs:int`
- must be between 0 and 100
- has a default value of 30

New in Version 10.2.2 – New dynamic cube configuration properties

This property was added.

memberCacheLimit

Reserved.

This property

- is of type `int`
is encoded as type `xs:int`
- has a default value of 100000
- must contain a value greater than or equal to 0

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Administration” on page 1836

This property was added.

postInMemoryTriggerName

Specifies the name of the trigger event to send after in-memory aggregates have loaded for a dynamic cube. When in-memory aggregates have finished loading, the event is triggered for execution against

the server that triggered the event. The purpose of the event is to run reports to populate the cube cache with data.

This property

- is of type `string`
is encoded as type `xs:string`

New in Version 10.2.2 – New dynamic cube configuration properties

This property was added.

predicateMemberReferenceThreshold

Specifies the percentage of members in a level that will be referenced in a filter predicate. If the percentage of members (in a level referenced by a filter predicate) exceeds the value specified by this property, a reference to the containing level will be used in the filter predicate.

A value of zero means the optimization is disabled.

This property

- is of type `int`
is encoded as type `xs:int`
- has a default value of 90
- must contain a value greater than or equal to 0
- must contain a value less than or equal to 100

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

rolapCubeMessages

Contains error and warning messages for a dynamic cube.

What's new

New in Version 10.2.2 – Dynamic cube messages

This class was added.

Properties

This class has the following properties.

cubeName

Specifies the name of the cube.

This property

- is of type `string`
is encoded as type `xs:string`

rolapMessages

Specifies an array containing error and warning messages.

This property

- is an array of type `bibus » rolapMessage`
is encoded as type `tns:rolapMessageArray`

rolapCubeMetrics

Contains metrics for a cube.

What's new

New in Version 10.2.2 – Dynamic cube metrics

This class was added.

Properties

This class has the following properties.

cubeName

Specifies the cube name.

This property

- is of type `string`
is encoded as type `xs:string`

rolapMetrics

Contains an array of metric names and values.

This property

- is an array of type `rolapMetric`
is encoded as type `tns:rolapMetricArray`

rolapDataSource

Identifies a base ROLAP cube data source.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » baseROLAPDataSource` class

What's new

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This class was added.

Properties

This class has the following properties.

aggregates

Contains the aggregates specification for this ROLAP datasource.

This property

- is of type `bibus » xmlEncodedXML`
is encoded as type `tns:xmlEncodedXMLProp`

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This property was added.

rolapMessage

Contains an error or warning message and associated timestamp.

What's new

New in Version 10.2.2 – Dynamic cube messages

This class was added.

Properties

This class has the following properties.

datetime

Specifies the name of the cube.

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

message

Contains an error or warning message.

This property

- is of type `string`
is encoded as type `xs:string`

rolapMetric

Defines the abstract class for all dynamic cube metric classes.

Derived classes

- [bibus](#) » [rolapMetricDateTime](#)
- [bibus](#) » [rolapMetricNumber](#)
- [bibus](#) » [rolapMetricString](#)

What's new

New in Version 10.2.2 – Dynamic cube metrics

This class was added.

Properties

This class has the following property

name

Specifies the metric name.

This property

- is of type `string`

is encoded as type `xs:string`

rolapMetricDateTime

Defines the `DateTime` values for the dynamic cube metric values.

This class inherits properties from the [bibus](#) » [rolapMetric](#) class.

What's new

New in Version 10.2.2 – [Dynamic cube metrics](#)

This class was added.

Properties

This class has the following property

value

Specifies the metric value. The possible values and a description of each value follow.

timeLastNearRealTimeUpdateAvailable

The date and time at which updates from the latest increment were made available.

This property

- is of type `dateTime`

is encoded as type `xs:dateTime`

rolapMetricNumber

Defines the numeric values for the dynamic cube metric values.

This class inherits properties from the [bibus](#) » [rolapMetric](#) class.

What's new

New in Version 10.2.2 – [Dynamic cube metrics](#)

This class was added.

Properties

This class has the following property

value

Specifies the metric value. The possible values and a description of each value follow.

aggregateTableHitRate

The percentage of data obtained from in-database aggregate tables when attempts are made to retrieve data from them since the start of the cube.

aggregateTableHitRateInLastHour

The percentage of data obtained from the in-database aggregate tables when attempts are made to retrieve data from them in the last hour.

averageSuccessfulRequestTime

The average time in milliseconds to execute a request for data since the start of a cube.

averageSuccessfulRequestTimeInLastHour

The average time in milliseconds to execute a request for data in the last hour.

dataCacheHitRate

The percentage of data obtained from the data cache when attempts are made to retrieve data from it since the start of the cube.

dataCacheHitRateInLastHour

The percentage of data obtained from the data cache when attempts are made to retrieve data from it in the last hour.

definedInMemoryAggregates

The number of in-memory aggregates defined for a cube.

definedSizeOfDataCache

The size in MB specified for a cube's data cache.

definedSizeOfInMemoryAggregateCache

The size in MB specified for a cube's in-memory aggregate cache.

definedSizeOfResultSetCache

The size in MB specified for a cube's on-disk result set cache.

inMemoryAggregateCacheHitRate

The percentage of data obtained from the in-memory aggregate cache when attempts are made to retrieve data from it since the start of the cube.

inMemoryAggregateCacheHitRateInLastHour

The percentage of data obtained from the in-memory aggregate cache when attempts are made to retrieve data from it in the last hour.

lastMetadataLoadTime

The amount of time in milliseconds required to load all members in the a cube's member cache.

lastResponseTime

The amount of time in milliseconds required to execute the most recent request for data.

loadedInMemoryAggregates

The number of in-memory aggregates currently loaded for a cube.

numberOfMembersInMemberCache

The number of members loaded into a cube's member cache.

numberOfProcessedRequests

The number of requests for data processed by a cube.

percentageTimeSpentRetrievingData

Percentage of time spent within a cube processing queries retrieving data from the underlying relational database since the start of a cube.

percentageTimeSpentRetrievingDataInLastHour

Percentage of time spent within a cube processing queries retrieving data from the underlying relational database in the last hour.

resultSetCacheHitRate

The percentage of queries for which results are obtained from the result set cache when attempts are made to retrieve results from it since the start of the cube.

resultSetCacheHitRateInLastHour

The percentage of queries for which results are obtained from the result set cache when attempts are made to retrieve results from it in the last hour.

timeSpentLoadingInMemoryAggregates

Length of time in milliseconds required to load the current set of in-memory aggregates in the cache.

timeToApplyLastNearRealTimeUpdates

The time in milliseconds required to apply the last set of near real time updates.

upTime

Length of time in milliseconds that a cube has been in a "running" state.

usedSizeOfDataCache

The used portion of a cube's data cache in MB.

usedSizeOfInMemoryAggregateCache

The used portion of a cube's in-memory aggregate cache in MB.

usedSizeOfResultSetCache

The used portion of a cube's result set cache in MB.

This property

- is of type `integer`

is encoded as type `xs:integer`

rolapMetricString

Defines the string values for the dynamic cube metric values.

This class inherits properties from the [bibus » rolapMetric](#) class.

What's new

New in Version 10.2.2 – [Dynamic cube metrics](#)

This class was added.

Properties

This class has the following property

value

Specifies the metric value. The possible values and a description of each value follow.

cubeState

Current state of the cube. One of the following values: disabled, stopped, starting, running, stopping, paused, pausing.

valueOfLastNearRealTimeTID

The value of the most recent transactionID (TID) used to obtain near real time updates from the database.

This property

- is of type `string`

is encoded as type `xs:string`

rolapVirtualCubeConfiguration

Defines configuration data for a virtual ROLAP cube.

This class

- inherits properties from the [bibus » baseROLAPCubeConfiguration](#) class

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This class was added.

rolapVirtualDataSource

Identifies a virtual ROLAP cube data source.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseROLAPDataSource](#) class

What's new

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This class was added.

role

Contains the information used to define a role.

Since most instances are based on information in a security system, IBM Cognos Analytics may constrain your ability to manipulate these objects. For example, you may not be able to change the value of a property, or the property may not return a value when queried.

The name of an instance may match the name of a sibling. In this case, a different search strategy should be used to distinguish between objects having the same name.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » dataSourceSignon » consumers](#)
- [bibus » adminFolder » contact](#)
- [bibus » aliasRoot » contact](#)
- [bibus » annotation » contact](#)
- [bibus » annotationFolder » contact](#)
- [bibus » archiveLocation » contact](#)
- [bibus » baseAgentDefinition » contact](#)
- [bibus » baseDataIntegrationTask » contact](#)
- [bibus » baseDataMovementTask » contact](#)
- [bibus » basePowerPlay8Report » contact](#)
- [bibus » basePowerPlayClass » contact](#)
- [bibus » baseReport » contact](#)
- [bibus » baseROLAPDataSource » contact](#)
- [bibus » contentTask » contact](#)
- [bibus » dashboard » contact](#)
- [bibus » dataSource » contact](#)
- [bibus » distributionList » contact](#)
- [bibus » document » contact](#)
- [bibus » documentVersion » contact](#)
- [bibus » drillPath » contact](#)
- [bibus » favoritesFolder » contact](#)
- [bibus » folder » contact](#)
- [bibus » indexUpdateTask » contact](#)
- [bibus » jobDefinition » contact](#)
- [bibus » launchable » contact](#)

- [bibus » migrationTask » contact](#)
- [bibus » model » contact](#)
- [bibus » mruFolder » contact](#)
- [bibus » namespace » contact](#)
- [bibus » package » contact](#)
- [bibus » pagelet » contact](#)
- [bibus » pageletFolder » contact](#)
- [bibus » personalization » contact](#)
- [bibus » personalizationFolder » contact](#)
- [bibus » planningApplication » contact](#)
- [bibus » planningMacroTask » contact](#)
- [bibus » planningTask » contact](#)
- [bibus » portalPackage » contact](#)
- [bibus » portalSkin » contact](#)
- [bibus » portlet » contact](#)
- [bibus » portletFolder » contact](#)
- [bibus » portletProducer » contact](#)
- [bibus » printer » contact](#)
- [bibus » queryServiceTask » contact](#)
- [bibus » reportCache » contact](#)
- [bibus » reportVersion » contact](#)
- [bibus » resource » contact](#)
- [bibus » subscriptionFolder » contact](#)
- [bibus » tenant » contact](#)
- [bibus » uiProfile » contact](#)
- [bibus » uiProfileFolder » contact](#)
- [bibus » URL » contact](#)
- [bibus » role » distributionMembers](#)
- [bibus » session » identity](#)
- [bibus » distributionList » members](#)
- [bibus » role » members](#)
- [bibus » documentContent » recipients](#)
- [bibus » output » recipients](#)
- [bibus » role » securityMembers](#)

Container Information

Contained by instances of the following classes

- [bibus » namespace](#)
- [bibus » namespaceFolder](#)

Properties

This class has the following properties.

distributionMembers

Refers to the members of this role for the purposes of distribution.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path

New in Version 8.3 – “New Properties for Group and Role Classes” on page 1919

This property was added.

governors

Contains the set of governors for this object.

This property

- is an array of type [bibus » governor](#)
 - is encoded as type `tns:governorArrayProp`

members

Refers to the members of this role .

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path

New in Version 8.3 – “New Properties for Group and Role Classes” on page 1919

This property was extended to allow instances of the [bibus » distributionList](#) class to be referenced by instances of this property.

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

routingHints

Specifies the routing hints for this object.

This property

- is an array of type `string`
 - is encoded as type `tns:stringArrayProp`
- can be acquired from a containing object

New in Version 8.3 — “Package Hierarchies” on page 1913

This property can now be acquired.

securityMembers

Refers to the members of this role for the purposes of security.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path

New in Version 8.3 — “New Properties for Group and Role Classes” on page 1919

This property was added.

root

Defines the class of the object at the top of the hierarchy.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » adminFolder](#)
- [bibus » capability](#)
- [bibus » catalog](#)
- [bibus » configuration](#)
- [bibus » content](#)
- [bibus » directory](#)
- [bibus » portal](#)
- [bibus » transientStateFolder](#)

Properties

This class has the following properties.

admin

Contains the root admin folder.

This property

- must have exactly 1 item

capabilities

Contains the root capability folder.

This property

- must have exactly 1 item

catalog

Contains a catalog of visualization modules used in reports.

This property

- must have exactly 1 item

New in Version 10.2.1 – “Visualization support” on page 1831

This property was added.

configuration

Contains the root configuration folder.

This property

- must have exactly 1 item

content

Contains the root content folder.

This property

- must have exactly 1 item

directory

Contains the root directory folder.

This property

- must have exactly 1 item

export – obsolete

import – obsolete

portal

Contains the portal content.

This property

- must have exactly 1 item

transient

Contains the transient state folder.

This property

- must have exactly 1 item

routingHintObject

Defines the class used to specify conditions for matching routing rules using the hint values on matching object types.

References

Used by the following properties:

- [bibus](#) » [routingRuleObject](#) » [hints](#)

Properties

This class has the following properties.

objectClass

Specifies the object class.

This property

- is of type [bibus » classEnum](#)
is encoded as type `tns:classEnum`

value

Specifies the value to be matched against the hint values from the object.

This property

- is of type `string`
is encoded as type `xs:string`

routingInfo

Defines the class for routing information in the BI Bus header.

References

Used by the following properties:

- [bibus » biBusHeader » routing](#)

Properties

This class has the following properties.

routingServerGroup

Specifies the server group that should handle the request.

This property

- is of type `string`
is encoded as type `xs:string`

routingRuleObject

Defines the class for dispatcher routing rules.

This class

- inherits properties from the [bibus » baseRoutingRule](#) class

Properties

This class has the following properties.

hints

Specifies the conditions that determine whether this rule can be used. Each condition must match before the rule can be used.

This property

- is an array of type [bibus » routingHintObject](#)
is encoded as type `tns:routingHintObjectArray`

routingTableEntry

Defines an entry in the routing table.

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [routingTable](#)
- [bibus](#) » [routingTableEntryArrayProp](#) » [value](#)
- [bibus](#) » [routingTableEntryProp](#) » [value](#)

Properties

This class has the following properties.

rule

Specifies the conditions used to match this table entry.

This property

- is of type [bibus](#) » [baseRoutingRule](#)
is encoded as type `tns:baseRoutingRule`

serverGroupName

Specifies the name of the server group that will be used to handle requests if this entry is selected.

This property

- is of type `string`
is encoded as type `xs:string`

routingTableEntryArrayProp

Defines the array property class for the [bibus](#) » [routingTableEntry](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [routingTable](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [routingTableEntry](#)
is encoded as type `tns:routingTableEntryArray`

routingTableEntryProp

Defines the simple property class for the [bibus » routingTableEntry](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » routingTableEntry](#)
is encoded as type `tns:routingTableEntry`

rssOption

Defines the abstract base class for all RSS option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » rssOptionAnyURI](#)
- [bibus » rssOptionSearchPathSingleObject](#)
- [bibus » rssOptionString](#)

Properties

This class has the following properties.

name

Identifies the RSS option.

This property

- is of type [bibus » rssOptionEnum](#)
is encoded as type `tns:rssOptionEnum`

rssOptionAnyURI

Defines URI values for the RSS options.

This class

- inherits properties from the [bibus » rssOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the RSS option.

This property

- is of type `anyURI`
is encoded as type `xs:string`

rssOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the RSS options.

This class

- inherits properties from the [bibus » rssOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the RSS option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

rssOptionString

Defines string values for the RSS options.

This class

- inherits properties from the [bibus » rssOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the RSS option.

This property

- is of type `string`
is encoded as type `xs:string`

runConditionEnumProp

Defines the simple property class for the [bibus » runConditionEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » agentTaskDefinition » runCondition](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » runConditionEnum](#)
is encoded as type `tns:runConditionEnum`

runningStateEnumProp

Defines the simple property class for the [bibus » runningStateEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » adaptiveAnalyticsService » runningState](#)
- [bibus » agentService » runningState](#)
- [bibus » annotationService » runningState](#)
- [bibus » batchReportService » runningState](#)
- [bibus » contentManagerCacheService » runningState](#)
- [bibus » contentManagerService » runningState](#)
- [bibus » dataAdvisorService » runningState](#)
- [bibus » dataIntegrationService » runningState](#)
- [bibus » dataMovementService » runningState](#)
- [bibus » deliveryService » runningState](#)
- [bibus » dimensionManagementService » runningState](#)
- [bibus » dispatcher » runningState](#)
- [bibus » eventManagementService » runningState](#)
- [bibus » EVService » runningState](#)
- [bibus » graphicsService » runningState](#)
- [bibus » humanTaskService » runningState](#)
- [bibus » idVizService » runningState](#)
- [bibus » indexDataService » runningState](#)
- [bibus » indexSearchService » runningState](#)
- [bibus » indexUpdateService » runningState](#)
- [bibus » jobService » runningState](#)
- [bibus » logService » runningState](#)
- [bibus » metadataService » runningState](#)
- [bibus » metricsManagerService » runningState](#)
- [bibus » migrationService » runningState](#)
- [bibus » mobileService » runningState](#)
- [bibus » monitorService » runningState](#)

- [bibus » planningAdministrationConsoleService » runningState](#)
- [bibus » planningDataService » runningState](#)
- [bibus » planningRuntimeService » runningState](#)
- [bibus » planningTaskService » runningState](#)
- [bibus » powerPlayService » runningState](#)
- [bibus » presentationService » runningState](#)
- [bibus » queryService » runningState](#)
- [bibus » relationalMetadataService » runningState](#)
- [bibus » reportDataService » runningState](#)
- [bibus » reportService » runningState](#)
- [bibus » repositoryService » runningState](#)
- [bibus » saCAMService » runningState](#)
- [bibus » systemService » runningState](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » runningStateEnum](#)
is encoded as type `tns:runningStateEnum`

runOption

Defines the abstract base class for all run option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » runOptionAnyURI](#)
- [bibus » runOptionBoolean](#)
- [bibus » runOptionData](#)
- [bibus » runOptionDateTime](#)
- [bibus » runOptionInt](#)
- [bibus » runOptionLanguageArray](#)
- [bibus » runOptionMultilingualString](#)
- [bibus » runOptionNameValueArray](#)
- [bibus » runOptionOutputEncapsulation](#)
- [bibus » runOptionPromptCacheMode](#)
- [bibus » runOptionSaveAs](#)
- [bibus » runOptionString](#)
- [bibus » runOptionStringArray](#)

References

Used by the following properties:

- [bibus](#) » [runOptionArrayProp](#) » [value](#)
- [bibus](#) » [runOptionProp](#) » [value](#)

Properties

This class has the following properties.

name

Identifies the run option.

This property

- is of type [bibus](#) » [runOptionEnum](#)
is encoded as type `tns:runOptionEnum`

runOptionAnyURI

Defines URI values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `anyURI`
is encoded as type `xs:string`

runOptionArrayProp

Defines the array property class for the [bibus](#) » [runOption](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [runOption](#)
is encoded as type `tns:runOptionArray`

runOptionBoolean

Defines boolean values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

runOptionData

Defines [bibus](#) » [dataEnum](#) values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type [bibus](#) » [dataEnum](#)
is encoded as type `tns:dataEnum`

runOptionDateTime

Defines `dateTime` values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `dateTime`
is encoded as type `xs:dateTime`

runOptionInt

Defines integer values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `int`
is encoded as type `xs:int`

runOptionLanguageArray

Defines language values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the run option.

This property

- is an array of type `language`
is encoded as type `tns:languageArray`

runOptionMultilingualString

Defines multilingual string values for the run options.

This class

- inherits properties from the [bibus](#) » [runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `string`
is encoded as type `tns:multilingualStringArray`
- is multilingual

runOptionNameValueArray

Defines [bibus » nameValue](#) values for the run options.

This class

- inherits properties from the [bibus » runOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the run option.

This property

- is an array of type [bibus » nameValue](#)
is encoded as type `tns:nameValueArray`

runOptionOutputEncapsulation

Defines [bibus » outputEncapsulationEnum](#) values for the run options.

This class

- inherits properties from the [bibus » runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type [bibus » outputEncapsulationEnum](#)
is encoded as type `tns:outputEncapsulationEnum`

runOptionPromptCacheMode

Defines [bibus » promptCacheModeEnum](#) values for the run options.

This class

- inherits properties from the [bibus » runOption](#) class

What's new

New in Version 8.4 — “[On Demand Refresh of Prompt Cache](#)” on page 1884

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type [bibus » promptCacheModeEnum](#)
is encoded as type `tns:promptCacheModeEnum`

runOptionProp

Defines the simple property class for the [bibus » runOption](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » runOption](#)
is encoded as type `tns:runOption`

runOptionSaveAs

Defines the class for specifying the saveAs run option.

This class

- inherits properties from the [bibus » runOption](#) class

Properties

This class has the following properties.

objectClass

Specifies the class of the new object.

This property

- is of type [bibus » reportSaveAsEnum](#)
is encoded as type `tns:reportSaveAsEnum`

objectName

Specifies the name of the new object.

This property

- is of type `token`
is encoded as type `tns:multilingualTokenArray`
- can contain at least 255 characters
- is multilingual

parentSearchPath

Specifies the search path of the object that will contain the object created as a result of a saveAs run option for the report.

Refers to instances of the [bibus » folder](#) class and the [bibus » package](#) class.

This property

- is of type `string`
is encoded as type `xs:string`

runOptionString

Defines string values for the run options.

This class

- inherits properties from the [bibus » runOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the run option.

This property

- is of type `string`
is encoded as type `xs:string`

runOptionStringArray

Defines string values for the run options.

This class

- inherits properties from the [bibus » runOption](#) class

Properties

This class has the following properties.

value

Specifies the values for the run option.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

runTimeState

Provides information about the run-time state of IBM Cognos Analytics components.

Instances may also be used to store session-specific information.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » adaptiveAnalyticsService](#)
- [bibus » agentService](#)

- [bibus » annotationService](#)
- [bibus » batchReportService](#)
- [bibus » contentManagerCacheService](#)
- [bibus » contentManagerService](#)
- [bibus » dataAdvisorService](#)
- [bibus » dataIntegrationService](#)
- [bibus » dataMovementService](#)
- [bibus » deliveryService](#)
- [bibus » dimensionManagementService](#)
- [bibus » dispatcher](#)
- [bibus » eventManagementService](#)
- [bibus » EVService](#)
- [bibus » graphicsService](#)
- [bibus » humanTaskService](#)
- [bibus » idVizService](#)
- [bibus » indexDataService](#)
- [bibus » indexSearchService](#)
- [bibus » indexUpdateService](#)
- [bibus » jobService](#)
- [bibus » logService](#)
- [bibus » metadataService](#)
- [bibus » metricsManagerService](#)
- [bibus » migrationService](#)
- [bibus » mobileService](#)
- [bibus » monitorService](#)
- [bibus » planningAdministrationConsoleService](#)
- [bibus » planningDataService](#)
- [bibus » planningRuntimeService](#)
- [bibus » planningTaskService](#)
- [bibus » powerPlayService](#)
- [bibus » presentationService](#)
- [bibus » queryService](#)
- [bibus » relationalMetadataService](#)
- [bibus » reportDataService](#)
- [bibus » reportService](#)
- [bibus » repositoryService](#)
- [bibus » saCAMService](#)
- [bibus » session](#)
- [bibus » systemService](#)

Properties

This class has the following properties.

state

Specifies the run-time state.

This property contains data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`

saCAMService

Defines run-time configuration parameters for the standalone CAM service.

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

What's new

New in Version 10.2.0 – [“New standalone IBM Cognos Access Manager \(CAM\) service” on page 1838](#)

This class was added.

Related information:

- [“Standalone Cognos Access Manager” on page 11](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type `anyType`
is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type `bibus » runningStateEnum`
is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

saCAMAuditLevel

Specifies the auditing level for the saCAM service.

This property

- is of type `bibus » auditLevelEnum`
is encoded as type `tns:auditLevelEnumProp`
- has a default value of `minimal`
- can be acquired from a containing object

New in Version 10.2.0 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1838

This property was added.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

schedule

Defines a schedule for a recurring task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contained by instances of the following classes

- `bibus » baseAgentDefinition`
- `bibus » baseDataIntegrationTask`

- [bibus » baseDataMovementTask](#)
- [bibus » basePowerPlay8Report](#)
- [bibus » baseReport](#)
- [bibus » contentTask](#)
- [bibus » exportDeployment](#)
- [bibus » importDeployment](#)
- [bibus » indexUpdateTask](#)
- [bibus » jobDefinition](#)
- [bibus » migrationTask](#)
- [bibus » planningMacroTask](#)
- [bibus » planningTask](#)
- [bibus » queryServiceTask](#)

Properties

This class has the following properties.

active

Specifies whether the schedule is active. Inactive schedules do not trigger events.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is searchable

credential

Refers to the credential for the schedule. Allows for automated logon so that the job can run at the scheduled time.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » credential](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

dailyPeriod

Specifies the unit of time used by the [bibus » schedule » everyNPeriods](#) property.

This property

- is of type [bibus » scheduleDailyPeriodEnum](#)
 - is encoded as type `tns:nmtokenProp`

endDate

Specifies the date and time after which no occurrences of the task will take place, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`

endType

Specifies how the date for the last occurrence of the task is determined.

This property

- is of type [bibus » scheduleEndTypeEnum](#)
is encoded as type `tns:nmtokenProp`

everyNPeriods

Specifies the number of periods between task occurrences. Use with the [bibus » schedule » dailyPeriod](#) property to schedule repeating tasks.

For example, to run a task every five days, set the [bibus » schedule » everyNPeriods](#) property to the value 5, and the [bibus » schedule » dailyPeriod](#) property to the value day.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`

intradayRecurrenceEnd

Specifies the end time of the intraday recurrence window.

The task associated with the schedule will continue running in the recurrence window according to the value specified by the [bibus » schedule » intradayRecurrenceInterval](#) property until the time specified by this value is reached. If the last recurrence occurs before the value specified by the [bibus » schedule » intradayRecurrenceEnd](#) property is reached, the schedule will not fire again at that time.

This property is only used if the [bibus » schedule » type](#) property is set to one of the following values:

1. [bibus » scheduleTypeEnum » dailyWithIntradayRecurrence](#)
2. [bibus » scheduleTypeEnum » monthlyAbsoluteWithIntradayRecurrence](#)
3. [bibus » scheduleTypeEnum » monthlyRelativeWithIntradayRecurrence](#)
4. [bibus » scheduleTypeEnum » weeklyWithIntradayRecurrence](#)
5. [bibus » scheduleTypeEnum » yearlyAbsoluteWithIntradayRecurrence](#)
6. [bibus » scheduleTypeEnum » yearlyRelativeWithIntradayRecurrence](#)

This property

- is of type `time`
is encoded as type `tns:timeProp`

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This property was added.

Related information:

[“Using Intraday Scheduling” on page 71](#)

intradayRecurrenceInterval

Specifies the time interval between occurrences of the intraday recurrence window.

This property is only used if the [bibus » schedule » type](#) property is set to one of the following values:

1. [bibus » scheduleTypeEnum » dailyWithIntradayRecurrence](#)
2. [bibus » scheduleTypeEnum » monthlyAbsoluteWithIntradayRecurrence](#)
3. [bibus » scheduleTypeEnum » monthlyRelativeWithIntradayRecurrence](#)
4. [bibus » scheduleTypeEnum » weeklyWithIntradayRecurrence](#)
5. [bibus » scheduleTypeEnum » yearlyAbsoluteWithIntradayRecurrence](#)

6. [bibus » scheduleTypeEnum » yearlyRelativeWithIntradayRecurrence](#)

This property

- is of type `duration`
is encoded as type `tns:durationProp`

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This property was added.

Related information:

[“Using Intraday Scheduling” on page 71](#)

intradayRecurrenceStart

Specifies the start time of the intraday recurrence window.

The task associated with the schedule will run at the specified time on the appropriate day, and run as often as specified by the [bibus » schedule » intradayRecurrenceInterval](#) property until the value specified by the [bibus » schedule » intradayRecurrenceEnd](#) property is reached.

This property is only used if the [bibus » schedule » type](#) property is set to one of the following values:

1. [bibus » scheduleTypeEnum » dailyWithIntradayRecurrence](#)
2. [bibus » scheduleTypeEnum » monthlyAbsoluteWithIntradayRecurrence](#)
3. [bibus » scheduleTypeEnum » monthlyRelativeWithIntradayRecurrence](#)
4. [bibus » scheduleTypeEnum » weeklyWithIntradayRecurrence](#)
5. [bibus » scheduleTypeEnum » yearlyAbsoluteWithIntradayRecurrence](#)
6. [bibus » scheduleTypeEnum » yearlyRelativeWithIntradayRecurrence](#)

This property

- is of type `time`
is encoded as type `tns:timeProp`

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This property was added.

Related information:

[“Using Intraday Scheduling” on page 71](#)

monthlyAbsoluteDay

Specifies the day in the month that the task occurs, such as 1 for the first day of the month.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- must contain a value less than or equal to 31

monthlyRelativeDay

Specifies the weekday that the task occurs each month, such as `monday`.

Use with [monthlyRelativeWeek](#) to specify the week and the day for a monthly schedule.

This property

- is of type [bibus » daysEnum](#)
is encoded as type `tns:nmtokenProp`

monthlyRelativeWeek

Specifies the week of the month that the task occurs.

Use with [monthlyRelativeDay](#) to specify the day and the week for a monthly schedule.

This property

- is of type [bibus](#) » [weeksEnum](#)
is encoded as type `tns:nmtokenProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus](#) » [parameterValue](#)
is encoded as type `tns:parameterValueArrayProp`

periodicalProducer

Specifies the producer of the output used to process subscriptions.

This property

- is an array of type [bibus](#) » [baseClass](#)
has items that must be of class [bibus](#) » [baseReport](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

priority

Specifies the schedule priority. Higher priority tasks run before lower priority tasks. Priority values range from 1 to 5. 1 is the highest priority; 5 is the lowest.

This property

- is of type `int`
is encoded as type `tns:intProp`
- has a default value of 3
- must contain a value greater than or equal to 1
- must contain a value less than or equal to 5
- is searchable

New in Version 8.3 – [“Schedule Priority”](#) on page 1912

This property was added.

New in Version 10.1.0 – [“Documentation Updates”](#) on page 1878

This topic now accurately describes the priority rules.

startDate

Specifies the date when the first occurrence of the task will take place, in Coordinated Universal Time (UTC).

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`

taskID

Contains the ID of the task, as identified in the schedule and delivery service.

This property

- is of type `token`
 - is encoded as type `tns:tokenProp`
- is read-only
- must contain exactly 45 characters
- must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase

timeZoneID

Specifies the time zone ID used in the schedule, in accordance with the ICU standard. If no ID is provided, the scheduling service uses the time zone of the server.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

triggerName

Specifies the name of the external occurrence that initiates execution of this schedule.

If the schedule is contained by an [bibus » authoredReport](#) object, the value of this property is set to the value of the [bibus » authoredReport » triggerName](#) property from that [bibus » authoredReport](#) object when it is set.

If the schedule is contained by a [bibus » reportView](#) object, the value of this property is set to the value of the [bibus » authoredReport » triggerName](#) property from the [bibus » authoredReport](#) object referenced by that [bibus » reportView](#) object (see the [bibus » reportView » base](#) property) when it is set.

In either of these cases, if the [bibus » authoredReport » triggerName](#) property for the associated [bibus » authoredReport](#) object has not been set, this property can be set independently.

If you require multiple external occurrences or conditions to trigger the execution of an object, you must manage this in the application you use to call the [event » trigger\(triggerName\)](#) method.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- is searchable

type

Specifies the type of schedule.

This property

- is of type [bibus](#) » [scheduleTypeEnum](#)
is encoded as type `tns:nmtokenProp`

weeklyFriday

Specifies whether the task occurs each Friday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklyMonday

Specifies whether the task occurs each Monday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklySaturday

Specifies whether the task occurs each Saturday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklySunday

Specifies whether the task occurs each Sunday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklyThursday

Specifies whether the task occurs each Thursday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklyTuesday

Specifies whether the task occurs each Tuesday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

weeklyWednesday

Specifies whether the task occurs each Wednesday.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

yearlyAbsoluteDay

Specifies the day in the month that the task occurs, such as 28 for the twenty-eighth day of the month.

Use with [yearlyAbsoluteMonth](#) to specify the day and the month for a yearly schedule.

This property

- is of type `positiveInteger`
is encoded as type `tns:positiveIntegerProp`
- must contain a value less than or equal to 31

yearlyAbsoluteMonth

Specifies the month that the task occurs each year, such as `june`.

Use with [yearlyAbsoluteDay](#) to specify the month and the day for a yearly schedule.

This property

- is of type `bibus » monthsEnum`
is encoded as type `tns:nmtokenProp`

yearlyRelativeDay

Specifies the weekday that the task occurs each year, such as `friday`.

Use with [yearlyRelativeWeek](#) and [yearlyRelativeMonth](#) to specify the week, the month, and the day for a yearly schedule.

This property

- is of type `bibus » daysEnum`
is encoded as type `tns:nmtokenProp`

yearlyRelativeMonth

Specifies the month that the task occurs each year, such as `july`.

Use with [yearlyRelativeWeek](#) and [yearlyRelativeDay](#) to specify the month, the week, and the day for a yearly schedule.

This property

- is of type `bibus » monthsEnum`
is encoded as type `tns:nmtokenProp`

yearlyRelativeWeek

Specifies the week of the month that the task occurs each year.

Use with [yearlyRelativeDay](#) and [yearlyRelativeMonth](#) to specify the month, the day, and the week for a yearly schedule.

This property

- is of type `bibus » weeksEnum`
is encoded as type `tns:nmtokenProp`

scheduledEvent

Defines an IBM Cognos Analytics event.

An event occurs when a scheduled task runs.

Properties

This class has the following properties.

credentialPath

Identifies the credential object used to authenticate this event.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`
- is read-only

objectPath

Identifies the object to be run.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`
- is read-only

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » option](#)
is encoded as type `tns:optionArray`
- is read-only

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus » parameterValue](#)
is encoded as type `tns:parameterValueArray`
- is read-only

schemaInfo

Defines a set of properties that provide information about a property, the state of a property value, and how a property value is processed by Content Manager.

If a property is acquirable, it also specifies whether contained objects acquire the property value.

References

Used by the following properties:

- [bibus » baseProp » schemaInfo](#)

Properties

This class has the following properties.

acquirable

Specifies whether contained objects can acquire the property value.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

acquired

Specifies whether the property value is acquired from a containing object.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

applyUpdateToDescendants

Specifies whether the values of corresponding properties in contained objects should be changed to acquire their values from a containing object.

When `applyUpdateToDescendants` is `true`, the value of the `acquired` property in corresponding properties in contained objects is set to `true`. This means that the corresponding properties acquire their values from a containing object.

When `applyUpdateToDescendants` is `false`, the value of the `acquired` property in corresponding properties in contained objects is not changed. This means that a corresponding property in a contained object that acquired a value from a containing object continues to do so, although the value that is acquired may be different. A corresponding property in a contained object that does not acquire a value from a containing object returns the same value when queried.

This property should be set to `true` only if the `acquirable` property is `true`.

This property

- is of type `boolean`
is encoded as type `xs:boolean`
- has a default value of `false`

modifiable

Specifies whether the property value can be modified.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

searchable

Specifies whether the property can be used in a search expression.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

searchPathMultipleObject

Defines a Content Manager search path that can resolve to an arbitrary number of objects.

Use this class to specify the location of one or more objects in the content store.

This class

- inherits properties from the `string`

References

Used by the following properties:

- [bibus » deliveryOptionSearchPathMultipleObjectArray » value](#)
- [bibus » indexOptionSearchPathMultipleObjectArray » value](#)
- [bibus » mobileOptionSearchPathMultipleObjectArray » value](#)

Used by the following method parameters:

- [standaloneCAM » terminateSessions\(search\) » search](#)
- [content » query\(searchPath, properties, sortBy, options\) » searchPath](#)

searchPathSingleObject

Defines a Content Manager search path that resolves to a single object.

Use this class to specify the location of an object in the content store.

This class

- inherits properties from the `string`

References

Used by the following properties:

- [bibus » deploymentImportRule » archiveSearchPath](#)
- [bibus » scheduledEvent » credentialPath](#)
- [bibus » asyncDetailDrillThroughTarget » drillPathSearchPath](#)
- [bibus » asyncRequest » objectPath](#)
- [bibus » scheduledEvent » objectPath](#)
- [bibus » deploymentImportRule » parent](#)
- [bibus » powerPlay8OptionSaveAs » parentSearchPath](#)
- [bibus » archiveDescriptor » reportSearchPath](#)
- [bibus » archiveDescriptor » reportViewSearchPath](#)
- [bibus » deploymentObjectInformation » searchPath](#)
- [bibus » drillThroughPath » searchPath](#)
- [bibus » memoPartAgentObject » searchPath](#)
- [bibus » memoPartObject » searchPath](#)
- [bibus » objectLink » searchPath](#)
- [bibus » asyncDetailDrillThroughRequest » target](#)
- [bibus » asyncDetailDrillThroughTarget » target](#)
- [bibus » migrationCubeMapping » target](#)
- [bibus » archiveOptionSearchPathSingleObject » value](#)
- [bibus » asyncOptionSearchPathSingleObject » value](#)
- [bibus » asyncOptionSearchPathSingleObjectArray » value](#)
- [bibus » contentTaskOptionSearchPathSingleObjectArray » value](#)

- [bibus » deliveryOptionSearchPathSingleObject » value](#)
- [bibus » deploymentOptionSearchPathSingleObjectArray » value](#)
- [bibus » migrationTaskOptionSearchPathSingleObject » value](#)
- [bibus » migrationTaskOptionSearchPathSingleObjectArray » value](#)
- [bibus » portalOptionSearchPathSingleObject » value](#)
- [bibus » portalOptionSearchPathSingleObjectArray » value](#)
- [bibus » powerPlay8OptionSearchPathSingleObject » value](#)
- [bibus » reportStudioOptionSearchPathSingleObject » value](#)
- [bibus » rssOptionSearchPathSingleObject » value](#)
- [bibus » searchPathSingleObjectArrayProp » value](#)
- [bibus » searchPathSingleObjectProp » value](#)
- [bibus » subscriptionOptionSearchPathSingleObject » value](#)

Used by the following method parameters:

- [content » addAnnotations\(containerPath, objects, options\) » containerPath](#)
- [dispatcher » ping\(dispatcherPath\) » dispatcherPath](#)
- [agent » deleteHotList\(objectPath\) » objectPath](#)
- [asynch » run\(objectPath, parameterValues, options\) » objectPath](#)
- [content » deleteAccount\(objectPath, options\) » objectPath](#)
- [delivery » addNotification\(objectPath\) » objectPath](#)
- [delivery » clearNotifications\(objectPath\) » objectPath](#)
- [delivery » deleteNotification\(objectPath\) » objectPath](#)
- [delivery » queryNotification\(objectPath\) » objectPath](#)
- [drillThrough » findDrillThroughPaths\(objectPath, parameterValues, options\) » objectPath](#)
- [drillThrough » queryDrillPath\(objectPath, parameterValues, options\) » objectPath](#)
- [event » runAt\(startTime, objectPath, parameterValues, options\) » objectPath](#)
- [indexUpdate » add\(objectPath, parameterValues, options\) » objectPath](#)
- [indexUpdate » delete\(objectPath, parameterValues, options\) » objectPath](#)
- [indexUpdate » get\(objectPath, parameterValues, options\) » objectPath](#)
- [parameter » collectParameterValues\(objectPath, parameterValues, options\) » objectPath](#)
- [parameter » getParameters\(objectPath, parameterValues, options\) » objectPath](#)
- [report » getObjectContext\(objectPath, parameterValues, options\) » objectPath](#)
- [report » query\(objectPath, parameterValues, options\) » objectPath](#)
- [validate » validate\(objectPath, parameterValues, options\) » objectPath](#)
- [content » determineRouting\(objectPaths\) » objectPaths](#)
- [content » add\(parentPath, objects, options\) » parentPath](#)
- [drillThrough » addDrillPath\(parentPath, object, options\) » parentPath](#)
- [report » add\(parentPath, object, options\) » parentPath](#)
- [authentication » logon\(credentials, roles\) » roles](#)
- [authentication » selectRoles\(roles\) » roles](#)
- [content » activate\(searchPath\) » searchPath](#)
- [dispatcher » startService\(servicePath\) » servicePath](#)
- [dispatcher » stopService\(servicePath, immediately\) » servicePath](#)
- [content » copyAccount\(sourceAccountPath, targetAccountPath, options\) » sourceAccountPath](#)

- [content](#) » [copyAccount\(sourceAccountPath, targetAccountPath, options\)](#) » [targetAccountPath](#)
- [content](#) » [copy\(objects, targetPath, options\)](#) » [targetPath](#)
- [content](#) » [copyRename\(objects, targetPath, newNames, options\)](#) » [targetPath](#)
- [content](#) » [move\(objects, targetPath, options\)](#) » [targetPath](#)
- [content](#) » [moveRename\(objects, targetPath, newNames, options\)](#) » [targetPath](#)

searchPathSingleObjectArrayProp

Defines the array property class for the [bibus](#) » [searchPathSingleObject](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObjectArray`

searchPathSingleObjectProp

Defines the simple property class for the [bibus](#) » [searchPathSingleObject](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

securedFeature

Defines a product feature.

The security policy associated with the secured feature determines the availability of the feature.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus](#) » [securedFunction](#)

Properties

This class has the following properties.

userCapability

Specifies the user capability represented by this instance.

This property

- is of type [bibus](#) » [userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumProp`

securedFunction

Defines a product function.

The security policy associated with the product function determines the availability of the function.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [productCapability](#)
- [bibus](#) » [securedFeature](#)

Contained by instances of the following classes

- [bibus](#) » [capability](#)

Properties

This class has the following properties.

features

Contains the secured features for this product function. For example, support for typed-in SQL may be a feature of an IBM Cognos Analytics query or report-authoring function.

userCapability

Specifies the user capability represented by this instance.

This property

- is of type [bibus](#) » [userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumProp`

session

Contains the temporary objects for the session.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contains instances of the following classes

- [bibus » analysis](#)
- [bibus » graphic](#)
- [bibus » output](#)
- [bibus » package](#)
- [bibus » query](#)
- [bibus » report](#)
- [bibus » reportView](#)
- [bibus » runTimeState](#)

Contained by instances of the following classes

- [bibus » account](#)

Properties

This class has the following properties.

groupAndRoleSettings

Reserved.

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`
- is read-only

identity

Identifies the user for the session. Specifically, refers to the account and the set of groups, roles, and namespaces associated with the session.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » account](#), [bibus » group](#), [bibus » namespace](#), or [bibus » role](#)
 - is encoded as type `tns:baseClassArrayProp`
- is read-only
- refers to other objects in the content store using an id-based search path

items

Contains the session temporary objects.

New in Version 10.1.0 – “[Query Modes](#)” on page 1865

This property was extended to allow instances of the [bibus » package](#) class to be contained by instances of this property.

userCapabilities

Contains the set of user capabilities for this session.

This property

- is an array of type `bibus » userCapabilityEnum`
is encoded as type `tns:userCapabilityEnumArrayProp`
- is read-only

setCookieVar

Defines the class used to store information about browser cookies that are expected to be set in the response.

References

Used by the following properties:

- `bibus » hdrSession » setCookieVars`

Related information:

- *IBM Cognos Administration and Security Guide*

Properties

This class has the following properties.

domain

Specifies the domain in which the cookie will be visible.

If nil, missing, or empty, the dispatcher will set the domain to the value specified in the global configuration.

If no value is set in the global configuration, the dispatcher will derive the domain from the host name of the request.

This property

- is of type `string`
is encoded as type `xs:string`

maxAge

Specifies the maximum age of the browser cookie in seconds.

If nil, missing, empty, or 0, a session cookie will be set.

If the value is less than 0, the cookie will be deleted.

If the value is greater than 0, the cookie will expire after the specified number of seconds have elapsed.

This property

- is of type `int`
is encoded as type `xs:int`

name

Identifies the browser cookie.

This property

- is of type `string`
is encoded as type `xs:string`

path

Specifies the location of the cookie.

If `nil` or missing, the dispatcher will set the path to the first item in the URI, such as `/series8`.

If the value is empty, the dispatcher will set the path to the full path in the URI, such as `/series8/bi/v1/disp`. This is usually an inappropriate location for the cookie.

This property

- is of type `string`
is encoded as type `xs:string`

secure

Specifies whether the cookie should be secured.

If `true`, the browser will not pass the cookie to the Web server unless the Web page is being viewed over a secure connection.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

value

Specifies the value assigned when setting the browser cookie.

This property

- is of type `string`
is encoded as type `xs:string`

severityEnumProp

Defines the simple property class for the [bibus » severityEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » history » maximumDetailSeverity](#)
- [bibus » baseHistoryDetail » severity](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » severityEnum](#)
is encoded as type `tns:severityEnum`

shortcut

Points to an object in another location.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » dashboard](#)
- [bibus » favoritesFolder](#)
- [bibus » folder](#)
- [bibus » mruFolder](#)
- [bibus » package](#)

Properties

This class has the following properties.

expirationTime

Specifies the date and time at which the object should be removed from the content store due to the retention rules specified on the parent object.

This property

- is of type `dateTime`
 - is encoded as type `tns:dateTimeProp`
- is read-only
- is searchable

New in Version 8.4 – “On Demand Refresh of Prompt Cache” on page 1884

This property was added.

target

Refers to the object represented by the shortcut.

If the target object is moved, renamed, or deleted, you may need to update this property.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » baseAgentDefinition](#), [bibus » basePowerPlay8Report](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » dashboard](#), [bibus » document](#), [bibus » folder](#), [bibus » jobDefinition](#), [bibus » launchable](#), [bibus » package](#), [bibus » pagelet](#), or [bibus » URL](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using a name- and type-based search path
- must have exactly 1 item

New in Version 8.4 – “Dashboards” on page 1894

This property was extended to allow instances of the [bibus » dashboard](#) class to be referenced by instances of this property.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be referenced by instances of this property.

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This property was extended to allow instances of the [bibus » launchable](#) class to be referenced by instances of this property.

shortcutAgentRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus » shortcut](#) object.

The object referenced by the [bibus » shortcut](#) is either the event list or hot list of the [bibus » baseAgentDefinition](#) that runs the task.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseRSSTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `shortcutAgentRSSTask` class.

Action	Mode	Service	Method
Run	All	deliveryService	<code>asynch » run(objectPath, parameterValues, options)</code>

Properties

This class has the following properties.

link

Specifies the agent output type represented by the shortcut created by this task.

This property

- is of type [bibus » agentOutputEnum](#)
is encoded as type `tns:agentOutputEnumProp`

shortcutRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus » shortcut](#) object.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » baseRSSTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the shortcutRSSTask class.

<i>Table 217. Services and methods for the shortcutRSSTask class.</i>			
Action	Mode	Service	Method
Run	All	deliveryService	<code>asynch » run(objectPath, parameterValues, options)</code>

Properties

This class has the following properties.

link

Refers to the object represented by the shortcut created by this task.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » basePowerPlay8Report](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » document](#), [bibus » folder](#), or [bibus » URL](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have exactly 1 item

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This property was extended to allow instances of the [bibus » basePowerPlay8Report](#) class to be referenced by instances of this property.

simpleParmValueItem

Use this class to define simple parameter values.

These parameters have a single value.

This class

- inherits properties from the [bibus » parmValueItem](#) class

References

Used by the following properties:

- [bibus » boundRangeParmValueItem » end](#)
- [bibus » unboundedStartRangeParmValueItem » end](#)
- [bibus » boundRangeParmValueItem » start](#)
- [bibus » unboundedEndRangeParmValueItem » start](#)
- [bibus » hierarchicalParmValueItem » value](#)

Properties

This class has the following properties.

display

Specifies the display value of the parameter value, as used in the interface.

This property

- is of type `string`
is encoded as type `xs:string`

use

Specifies the value of the parameter value, as used for the query.

This property

- is of type `string`
is encoded as type `xs:string`

softwareEdition

Defines information used to build URLs to send requests to a particular edition of IBM Cognos Analytics software.

References

Used by the following properties:

- [bibus » configuration » editions](#)
- [bibus » softwareEditionArrayProp » value](#)
- [bibus » softwareEditionProp » value](#)

What's new

New in Version 8.4 — “Software Editions” on page 1893

This class was added.

Properties

This class has the following properties.

gateway

Specifies the gateway for the edition.

When using the default gateway and configuration, the value for this property is `bi/v1/disp`.

This property

- is of type `string`
is encoded as type `xs:string`
- must contain no more than 50 characters

id

Identifies the software edition.

Values for this property must be of the form `N.N.N` where `N` consists of one or more digits.

This property

- is of type `string`
is encoded as type `xs:string`
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')

virtualDirectory

Specifies the location of the IBM Cognos Analytics installation.

Values for this property *should* contain the full path the IBM Cognos Analytics software, and MAY contain the protocol and machine name. For example, /cognos8 or http://www.systems.example.com/cognos8.

This property

- is of type anyURI
 - is encoded as type xs:string
- must contain no more than 100 characters

softwareEditionArrayProp

Defines the array property class for the [bibus](#) » [softwareEdition](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [editions](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [softwareEdition](#)
 - is encoded as type tns:softwareEditionArray

softwareEditionProp

Defines the simple property class for the [bibus](#) » [softwareEdition](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [softwareEdition](#)
 - is encoded as type tns:softwareEdition

sort

Defines sort specifications for a Content Manager query request.

References

Used by the following properties:

- [bibus](#) » [queryRequest](#) » [sortBy](#)

Used by the following method parameters:

- [content](#) » [query\(searchPath, properties, sortBy, options\)](#) » [sortBy](#)

Properties

This class has the following properties.

order

Specifies the order in which the objects are sorted.

This property

- is of type [bibus](#) » [orderEnum](#)
is encoded as type `tns:orderEnum`

propName

Specifies the property by which the result set is sorted.

This property

- is of type [bibus](#) » [propEnum](#)
is encoded as type `tns:propEnum`

specification

Defines the type for service specifications.

This class

- inherits properties from the [bibus](#) » [xmlEncodedXML](#) class

References

Used by the following properties:

- [bibus](#) » [asynchSpecification](#) » [value](#)

specificationOption

Defines the abstract base class for all specification option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus](#) » [option](#) class

Derived Classes

- [bibus](#) » [specificationOptionString](#)
- [bibus](#) » [specificationOptionXMLEncodedXML](#)

What's new

New in Version 8.4 – “[Dynamic Filtering of Report Data](#)” on page 1889

This class was added.

Properties

This class has the following properties.

name

Identifies the specification option.

This property

- is of type [bibus » specificationOptionEnum](#)
is encoded as type `tns:specificationOptionEnum`

specificationOptionString

Defines string values for the specification options.

This class

- inherits properties from the [bibus » specificationOption](#) class

What's new

New in Version 10.1.0 – “[Batch Report Service/Report Service Optimizations](#)” on page 1870

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the specification option.

This property

- is of type `string`
is encoded as type `xs:string`

specificationOptionXMLEncodedXML

Defines `xmlEncodedXML` values for the specification options.

This class

- inherits properties from the [bibus » specificationOption](#) class

What's new

New in Version 8.4 – “[Dynamic Filtering of Report Data](#)” on page 1889

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the specification option.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

SQL

Contains the SQL generated during the execution of a report.

The SQL for any IBM Cognos Analytics report is stored with the output because changes in the model may modify the data returned in the report.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » reportVersion](#)

Properties

This class has the following properties.

source

Specifies the SQL statement.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 32767 characters

storedProcedureTask

Defines the class that describes a task that calls an RDBMS stored procedure.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the storedProcedureTask class.

Action	Mode	Service	Method
Run	All	agentService	<code>asynch » run(objectPath, parameterValues, options)</code>

Container Information

Contains instances of the following classes

- [bibus » history](#)

Contained by instances of the following classes

- [bibus » authoredAgentDefinition](#)

Properties

This class has the following properties.

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus » history](#)

metadataModel

Refers to the metadata model needed to execute the specification. The content store location of this Framework Manager model is identified by means of a search path.

This model specifies the data to be retrieved, how rollups are aggregated, and the dimensional hierarchies or nested levels that are present in the source cube. The model also defines query properties, such as whether columns are additive or whether governors are used to narrow the scope of the query, thereby overriding the default behavior for a specific report.

Packages can contain objects from more than one model. Therefore, you can add a model reference or change the referenced model to meet the business requirements of your report authors. For example, you can add objects to a model so that a new calculated column is available for use in your reports. After making changes to the model, you must republish the package and update any reports that use the data.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » model](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

Related information:

IBM Cognos Framework Manager *Developer Guide*

metadataModelPackage

Refers to the package containing the metadata model needed to execute the specification.

This property

- is an array of type [bibus » baseClass](#)
 - has items that must be of class [bibus » package](#)
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » option](#)
 - is encoded as type `tns:optionArrayProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type [bibus » parameterValue](#)
 - is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
bibus » history class		5	bibus » baseClass » creationTime property

This property

- is an array of type [bibus » retentionRule](#)
 - is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the [content » determineRouting\(objectPaths\)](#) method . The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Software Editions” on page 1893

This property was added.

storedProcedureName

Specifies the name of the stored procedure, as defined in the metadata model.

This property

- is of type `bibus » metadataModelItemName`
 - is encoded as type `tns:metadataModelItemNameProp`

stringArrayProp

Defines the array property class for the `string`.

This class

- inherits properties from the `bibus » baseProp` class

References

Used by the following properties:

- `bibus » agentTaskDefinition » eventKey`
- `bibus » authoredAgentDefinition » eventKey`
- `bibus » documentContent » recipientsEMail`
- `bibus » output » recipientsEMail`
- `bibus » account » routingHints`
- `bibus » dashboard » routingHints`
- `bibus » group » routingHints`
- `bibus » package » routingHints`
- `bibus » portletProducer » routingHints`

- [bibus](#) » [role](#) » [routingHints](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type `string`
is encoded as type `tns:stringArray`

stringMapEntry

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [mobileConfiguration](#)
- [bibus](#) » [stringMapEntryArrayProp](#) » [value](#)
- [bibus](#) » [stringMapEntryProp](#) » [value](#)

Properties

This class has the following properties.

key

This property

- is of type `string`
is encoded as type `xs:string`

value

This property

- is of type `string`
is encoded as type `xs:string`

stringMapEntryArrayProp

Defines the array property class for the [bibus](#) » [stringMapEntry](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [mobileConfiguration](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » stringMapEntry](#)
is encoded as type `tns:stringMapEntryArray`

stringMapEntryProp

Defines the simple property class for the [bibus » stringMapEntry](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » stringMapEntry](#)
is encoded as type `tns:stringMapEntry`

stringProp

Defines the simple property class for the `string`.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » planningApplication » applicationID](#)
- [bibus » humanTask » bindingName](#)
- [bibus » webServiceTask » bindingName](#)
- [bibus » drillPath » bookmarkText](#)
- [bibus » documentContent » burstID](#)
- [bibus » output » burstID](#)
- [bibus » documentContent » burstKey](#)
- [bibus » output » burstKey](#)
- [bibus » account » businessPhone](#)
- [bibus » contact » businessPhone](#)
- [bibus » dataSourceConnection » connectionString](#)
- [bibus » adminFolder » contactEMail](#)
- [bibus » aliasRoot » contactEMail](#)
- [bibus » annotation » contactEMail](#)
- [bibus » annotationFolder » contactEMail](#)

- [bibus » archiveLocation » contactEMail](#)
- [bibus » baseAgentDefinition » contactEMail](#)
- [bibus » baseDataIntegrationTask » contactEMail](#)
- [bibus » baseDataMovementTask » contactEMail](#)
- [bibus » basePowerPlay8Report » contactEMail](#)
- [bibus » basePowerPlayClass » contactEMail](#)
- [bibus » baseReport » contactEMail](#)
- [bibus » baseROLAPDataSource » contactEMail](#)
- [bibus » contentTask » contactEMail](#)
- [bibus » dashboard » contactEMail](#)
- [bibus » dataSource » contactEMail](#)
- [bibus » distributionList » contactEMail](#)
- [bibus » document » contactEMail](#)
- [bibus » documentVersion » contactEMail](#)
- [bibus » drillPath » contactEMail](#)
- [bibus » favoritesFolder » contactEMail](#)
- [bibus » folder » contactEMail](#)
- [bibus » indexUpdateTask » contactEMail](#)
- [bibus » jobDefinition » contactEMail](#)
- [bibus » launchable » contactEMail](#)
- [bibus » migrationTask » contactEMail](#)
- [bibus » model » contactEMail](#)
- [bibus » mruFolder » contactEMail](#)
- [bibus » namespace » contactEMail](#)
- [bibus » package » contactEMail](#)
- [bibus » pagelet » contactEMail](#)
- [bibus » pageletFolder » contactEMail](#)
- [bibus » personalization » contactEMail](#)
- [bibus » personalizationFolder » contactEMail](#)
- [bibus » planningApplication » contactEMail](#)
- [bibus » planningMacroTask » contactEMail](#)
- [bibus » planningTask » contactEMail](#)
- [bibus » portalPackage » contactEMail](#)
- [bibus » portalSkin » contactEMail](#)
- [bibus » portlet » contactEMail](#)
- [bibus » portletFolder » contactEMail](#)
- [bibus » portletProducer » contactEMail](#)
- [bibus » printer » contactEMail](#)
- [bibus » queryServiceTask » contactEMail](#)
- [bibus » reportCache » contactEMail](#)
- [bibus » reportVersion » contactEMail](#)
- [bibus » resource » contactEMail](#)
- [bibus » subscriptionFolder » contactEMail](#)

- [bibus » tenant » contactEMail](#)
- [bibus » uiProfile » contactEMail](#)
- [bibus » uiProfileFolder » contactEMail](#)
- [bibus » URL » contactEMail](#)
- [bibus » URL » contentType](#)
- [bibus » basePowerPlayClass » cubeContentName](#)
- [bibus » dataSourceCredential » dataSourceConnectionName](#)
- [bibus » dataSourceCredential » dataSourceName](#)
- [bibus » cacheOutput » dataType](#)
- [bibus » dashboard » dataType](#)
- [bibus » documentContent » dataType](#)
- [bibus » graphic » dataType](#)
- [bibus » historyDetailDataMovementService » dataType](#)
- [bibus » output » dataType](#)
- [bibus » page » dataType](#)
- [bibus » visualization » dataType](#)
- [bibus » uiClass » defaultDescription](#)
- [bibus » uiClass » defaultScreenTip](#)
- [bibus » authoredReport » defaultTriggerDescription](#)
- [bibus » baseHistoryDetail » detail](#)
- [bibus » document » documentType](#)
- [bibus » dispatcher » edition](#)
- [bibus » account » email](#)
- [bibus » contact » email](#)
- [bibus » history » eventID](#)
- [bibus » account » faxPhone](#)
- [bibus » contact » faxPhone](#)
- [bibus » account » givenName](#)
- [bibus » contact » givenName](#)
- [bibus » account » homePhone](#)
- [bibus » contact » homePhone](#)
- [bibus » humanTask » inputMessageName](#)
- [bibus » webServiceTask » inputMessageName](#)
- [bibus » launchable » launchableType](#)
- [bibus » printer » location](#)
- [bibus » account » mobileDeviceID](#)
- [bibus » account » mobilePhone](#)
- [bibus » contact » mobilePhone](#)
- [bibus » printer » modelName](#)
- [bibus » humanTask » operationName](#)
- [bibus » webServiceTask » operationName](#)
- [bibus » humanTask » outputMessageName](#)
- [bibus » webServiceTask » outputMessageName](#)

- [bibus » history » ownerEventID](#)
- [bibus » agentTaskDefinition » packageBase](#)
- [bibus » baseAgentDefinition » packageBase](#)
- [bibus » baseDataIntegrationTask » packageBase](#)
- [bibus » basePowerPlayClass » packageBase](#)
- [bibus » baseReport » packageBase](#)
- [bibus » folder » packageBase](#)
- [bibus » jobDefinition » packageBase](#)
- [bibus » jobStepDefinition » packageBase](#)
- [bibus » model » packageBase](#)
- [bibus » package » packageBase](#)
- [bibus » planningApplication » packageBase](#)
- [bibus » URL » packageBase](#)
- [bibus » account » pagerPhone](#)
- [bibus » contact » pagerPhone](#)
- [bibus » account » postalAddress](#)
- [bibus » contact » postalAddress](#)
- [bibus » portalSkin » previewImageLocation](#)
- [bibus » printer » printerAddress](#)
- [bibus » configuration » qsAdditionalJVMArguments](#)
- [bibus » configurationFolder » qsAdditionalJVMArguments](#)
- [bibus » dispatcher » qsAdditionalJVMArguments](#)
- [bibus » queryService » qsAdditionalJVMArguments](#)
- [bibus » portalSkin » resourceLocation](#)
- [bibus » resource » resourceType](#)
- [bibus » history » restartEventID](#)
- [bibus » baseAgentDefinition » routingServerGroup](#)
- [bibus » baseDataIntegrationTask » routingServerGroup](#)
- [bibus » baseDataMovementTask » routingServerGroup](#)
- [bibus » basePowerPlay8Report » routingServerGroup](#)
- [bibus » baseReport » routingServerGroup](#)
- [bibus » baseRSSTask » routingServerGroup](#)
- [bibus » contentTask » routingServerGroup](#)
- [bibus » dashboard » routingServerGroup](#)
- [bibus » exportDeployment » routingServerGroup](#)
- [bibus » humanTask » routingServerGroup](#)
- [bibus » importDeployment » routingServerGroup](#)
- [bibus » indexUpdateTask » routingServerGroup](#)
- [bibus » jobDefinition » routingServerGroup](#)
- [bibus » memo » routingServerGroup](#)
- [bibus » migrationTask » routingServerGroup](#)
- [bibus » package » routingServerGroup](#)
- [bibus » planningMacroTask » routingServerGroup](#)

- [bibus](#) » [planningTask](#) » [routingServerGroup](#)
- [bibus](#) » [portletProducer](#) » [routingServerGroup](#)
- [bibus](#) » [queryServiceTask](#) » [routingServerGroup](#)
- [bibus](#) » [storedProcedureTask](#) » [routingServerGroup](#)
- [bibus](#) » [webServiceTask](#) » [routingServerGroup](#)
- [bibus](#) » [baseAgentDefinition](#) » [routingServerGroupEdition](#)
- [bibus](#) » [baseDataIntegrationTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [baseDataMovementTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [basePowerPlay8Report](#) » [routingServerGroupEdition](#)
- [bibus](#) » [baseReport](#) » [routingServerGroupEdition](#)
- [bibus](#) » [baseRSSTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [contentTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [dashboard](#) » [routingServerGroupEdition](#)
- [bibus](#) » [exportDeployment](#) » [routingServerGroupEdition](#)
- [bibus](#) » [humanTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [importDeployment](#) » [routingServerGroupEdition](#)
- [bibus](#) » [indexUpdateTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [jobDefinition](#) » [routingServerGroupEdition](#)
- [bibus](#) » [memo](#) » [routingServerGroupEdition](#)
- [bibus](#) » [migrationTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [package](#) » [routingServerGroupEdition](#)
- [bibus](#) » [planningMacroTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [planningTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [portalSkin](#) » [routingServerGroupEdition](#)
- [bibus](#) » [portletProducer](#) » [routingServerGroupEdition](#)
- [bibus](#) » [queryServiceTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [storedProcedureTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [webServiceTask](#) » [routingServerGroupEdition](#)
- [bibus](#) » [history](#) » [scheduleTriggerName](#)
- [bibus](#) » [baseClass](#) » [searchPath](#)
- [bibus](#) » [baseReport](#) » [serverGroup](#)
- [bibus](#) » [configuration](#) » [serverGroup](#)
- [bibus](#) » [configurationFolder](#) » [serverGroup](#)
- [bibus](#) » [content](#) » [serverGroup](#)
- [bibus](#) » [dispatcher](#) » [serverGroup](#)
- [bibus](#) » [folder](#) » [serverGroup](#)
- [bibus](#) » [jobDefinition](#) » [serverGroup](#)
- [bibus](#) » [package](#) » [serverGroup](#)
- [bibus](#) » [reportVersion](#) » [serverGroup](#)
- [bibus](#) » [humanTask](#) » [serviceName](#)
- [bibus](#) » [webServiceTask](#) » [serviceName](#)
- [bibus](#) » [SQL](#) » [source](#)
- [bibus](#) » [account](#) » [surname](#)

- [bibus » contact » surname](#)
- [bibus » baseClass » tenantID](#)
- [bibus » account » timeZoneID](#)
- [bibus » contact » timeZoneID](#)
- [bibus » schedule » timeZoneID](#)
- [bibus » authoredReport » triggerName](#)
- [bibus » schedule » triggerName](#)
- [bibus » account » userName](#)
- [bibus » contact » userName](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `string`
is encoded as type `xs:string`

subscriptionFolder

Contains the subscriptions created by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » reportDataServiceAgentDefinition](#)
- [bibus » subscriptionFolder](#)

Contained by instances of the following classes

- [bibus » account](#)
- [bibus » subscriptionFolder](#)

What's new

New in Version 8.3 — [“Conditional Subscriptions” on page 1899](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)
is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
is encoded as type `tns:stringProp`

items

Contains the child objects for this object.

subscriptionOption

Defines the abstract base class for all subscription option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » subscriptionOptionSearchPathSingleObject](#)

What's new

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This class was added.

Properties

This class has the following properties.

name

Identifies the subscription option.

This property

- is of type [bibus » subscriptionOptionEnum](#)
is encoded as type `tns:subscriptionOptionEnum`

subscriptionOptionSearchPathSingleObject

Defines [bibus » searchPathSingleObject](#) values for the subscription options.

This class

- inherits properties from the [bibus » subscriptionOption](#) class

What's new

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This class was added.

Properties

This class has the following properties.

value

Specifies the value for the subscription option.

This property

- is of type [bibus » searchPathSingleObject](#)
is encoded as type `tns:searchPathSingleObject`

systemMetricThresholds

Contains the information used to define thresholds for a system metric.

By default, system metrics are not collected. To enable the collection of system metrics, you must edit the `iManage-metadata.xml` file in your installation.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » adaptiveAnalyticsService](#)
- [bibus » agentService](#)
- [bibus » annotationService](#)
- [bibus » batchReportService](#)
- [bibus » configuration](#)
- [bibus » contentManagerCacheService](#)
- [bibus » contentManagerService](#)
- [bibus » dataAdvisorService](#)
- [bibus » dataIntegrationService](#)
- [bibus » dataMovementService](#)
- [bibus » deliveryService](#)
- [bibus » dimensionManagementService](#)
- [bibus » dispatcher](#)
- [bibus » eventManagementService](#)
- [bibus » EVService](#)
- [bibus » graphicsService](#)
- [bibus » humanTaskService](#)
- [bibus » idVizService](#)
- [bibus » indexDataService](#)
- [bibus » indexSearchService](#)
- [bibus » indexUpdateService](#)

- [bibus » jobService](#)
- [bibus » logService](#)
- [bibus » metadataService](#)
- [bibus » metricsManagerService](#)
- [bibus » migrationService](#)
- [bibus » mobileService](#)
- [bibus » monitorService](#)
- [bibus » planningAdministrationConsoleService](#)
- [bibus » planningDataService](#)
- [bibus » planningRuntimeService](#)
- [bibus » planningTaskService](#)
- [bibus » powerPlayService](#)
- [bibus » presentationService](#)
- [bibus » queryService](#)
- [bibus » relationalMetadataService](#)
- [bibus » reportDataService](#)
- [bibus » reportService](#)
- [bibus » repositoryService](#)
- [bibus » saCAMService](#)
- [bibus » systemService](#)

What's new

New in Version 8.3 — [“System Metrics” on page 1908](#)

This class was added.

New in Version 10.1.0 — [“Documentation Updates” on page 1878](#)

This topic now provides additional information about enabling system metrics collection.

Related information:

- *IBM Cognos Administration and Security Guide*

Properties

This class has the following properties.

properties

Contains the threshold values for the system metric.

This property

- is an array of type [bibus » uriValue](#)
is encoded as type `tns:uriValueArrayProp`

systemMetric

Specifies the system metric.

This property

- is of type [bibus » systemMetricEnum](#)
is encoded as type `tns:anyURIProp`

- is searchable

systemService

Defines run-time configuration parameters for the [systemService](#).

Values for many of the properties of this class can be acquired from the parent object. You can use property acquisition to simplify the configuration of IBM Cognos Analytics installations.

A dispatcher creates an instance of this class within its associated [bibus » dispatcher](#) object if the object does not already exist.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus » runTimeState](#)
- [bibus » systemMetricThresholds](#)

Contained by instances of the following classes

- [bibus » dispatcher](#)

Properties

This class has the following properties.

advancedSettings

Specifies advanced configuration information in XML format.

The settings can be changed at run time, without having to restart the server. However, changing these settings using the SDK requires advanced XML programming skills.

For more information, see *Advanced settings*.

This property

- is of type anyType
 - is encoded as type `tns:anyTypeProp`
- can be acquired from a containing object

runningState

Specifies the running state of the service.

This property

- is of type [bibus » runningStateEnum](#)
 - is encoded as type `tns:runningStateEnumProp`

runTimeState

Contains the run-time state information.

This property

- must have at most 1 item

ssAuditLevel

Specifies the auditing level for the system service.

This property

- is of type [bibus » auditLevelEnum](#)
 - is encoded as type `tns:auditLevelEnumProp`
- has a default value of [minimal](#)
- can be acquired from a containing object

systemMetricThresholds

Contains the set of system metric thresholds.

New in Version 8.3 – “System Metrics” on page 1908

This property was added.

tenancy

Contains the user tenant information for the current session.

What's new

New in Version 10.2.2 – [Full tenant impersonation capability for system administrators](#)

This class was added.

Properties

This class has the following properties.

tenantID

Identifies the user's true tenantID.

This property

- is of type string
- is encoded as type `xs:string`
- is read-only

readTenantIDs

Identifies the set of tenantIDs the user belongs to, i.e. the user's tenantID bounding set.

This property

- is an array of type string
- is encoded as type `tns:stringArray`
- is read-only

writeTenantID

Identifies the user's tenantID that is used to create objects. The writeTenantID must be a subset of the readTenantIDs.

This property

- is of type string
- is encoded as type `xs:string`
- is read-only

tenant

Defines the class for tenants in a multi-tenant environment.

The `contact` property refers to a person or group of people who are responsible for the object.

This object's `tenantID` property is read-only.

There can only be one instance of this class (object in the content store) per `tenantID`.

Instances of this class in the content store can be deleted only by members of the System Administrators role using the `content » deleteTenants(tenantIDs)` method.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contained by instances of the following classes

- `bibus » tenants`

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type `bibus » baseClass`

has items that must be of class `bibus » account`, `bibus » contact`, `bibus » distributionList`, `bibus » group`, or `bibus » role`

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

tenantMembers

Contains additional tenant members. By default, all users belonging to this tenant in the authentication provider are implicitly contained in this list.

This property

- is an array of type `bibus » baseClass`

has items that must be of class `bibus » tenant`

is encoded as type `tns:baseClassArrayProp`

- refers to other tenant objects in the content manager using an id-based search path

New in Version 10.2.2 – “Support for delegated tenant administration” on page 1826

This property was added.

tenantInfo

Contains information for a specific tenant.

References

Used by the following method return values:

- [content](#) » [listTenants\(options\)](#) » [result](#)

What's new

New in Version 10.2.0 – “[Support for multi-tenancy](#)” on page 1838

This class was added.

Properties

This class has the following properties.

tenantID

Identities the tenant.

This property

- is of type `string`
 - is encoded as type `xs:string`
- is read-only

tenants

Contains tenant objects in a multi-tenancy environment.

This object is created when the content store is initialized. This object cannot be deleted by a user.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contains instances of the following classes

- [bibus](#) » [tenant](#)

Contained by instances of the following classes

- [bibus](#) » [directory](#)

Properties

This class has the following properties.

items

Contains the tenant objects.

timeProp

Defines the simple property class for the `time`.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#)
- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type `time`

is encoded as type `xs:time`

timeZone

Defines time zone information, in accordance with the ICU standard.

References

Used by the following properties:

- [bibus](#) » [configurationData](#) » [serverTimeZone](#)
- [bibus](#) » [configurationData](#) » [timeZones](#)

Properties

This class has the following properties.

id

Identifies the time zone.

This property

- is of type `string`

is encoded as type `xs:string`

- is read-only

name

Specifies the name of the time zone in the language specified by the product or content locale in the user preferences for the request.

This property

- is of type `string`

is encoded as type `xs:string`

- is read-only

tokenArrayProp

Defines the array property class for the token.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [deploymentDetail](#) » [deployedObjectAncestorDefaultNames](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type token
is encoded as type `tns:tokenArray`

tokenProp

Defines the simple property class for the token.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [baseClass](#) » [defaultName](#)
- [bibus](#) » [deploymentDetail](#) » [deployedObjectDefaultName](#)
- [bibus](#) » [configuration](#) » [qsROLAPMemberCacheAliasRoot](#)
- [bibus](#) » [configurationFolder](#) » [qsROLAPMemberCacheAliasRoot](#)
- [bibus](#) » [dispatcher](#) » [qsROLAPMemberCacheAliasRoot](#)
- [bibus](#) » [queryService](#) » [qsROLAPMemberCacheAliasRoot](#)
- [bibus](#) » [schedule](#) » [taskID](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type token
is encoded as type `xs:string`

tracking

Defines the class that contains information about a request or response for the log message facility. This information can apply to multiple operations that cross multiple components.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [tracking](#)

Properties

This class has the following properties.

conversationContext

Specifies state information relating to the asynchronous operation in progress.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus](#) » [conversationContext](#)
is encoded as type `tns:conversationContext`

hopCount

Specifies the number of Simple Object Access Protocol (SOAP) nodes that have handled the request. This property is set to 0 when the message is handled by the first IBM Cognos Analytics SOAP node. When the dispatcher load-balances the request, or forwards the request to a service, the value of the property is incremented. The value may also be incremented if the dispatcher needs to redispach the request, such as in the case of Content Manager failover.

This value is not incremented when a provider needs to perform additional requests to satisfy the original request.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `integer`
is encoded as type `xs:integer`

providers

Specifies the providers that contributed to the response.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is an array of type [bibus](#) » [provider](#)
is encoded as type `tns:providerArray`

requestContext

Identifies the request. This request ID is used to associate related operations across multiple components.

For example, the report service may consider the execution of a report to be a request.

A request occurs within a session and usually consists of multiple steps.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

sessionContext

Identifies the session in which a request or response has occurred. A session begins when a user logs on.

Multiple requests can occur within the same session.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

stepContext

Identifies the step. This step ID is used to associate an action initiated by a user or automated process, and the response to that action.

For example, the report service may consider the return of a page of report output to the client during the execution of a report to be a step.

A step occurs within a request.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type `string`
is encoded as type `xs:string`

transientStateFolder

Contains objects that store temporary data required by services to maintain information between occurrences of object execution.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the [bibus » baseClass](#) class

Container Information

Contains instances of the following classes

- [bibus » agentState](#)
- [bibus » periodical](#)

Contained by instances of the following classes

- [bibus » root](#)

Properties

This class has the following properties.

items

Contains the child objects.

New in Version 8.3 — “Conditional Subscriptions” on page 1899

This property can now contain instances of the [bibus » periodical](#) class.

uiClass

Defines the abstract class for the set of common interface-related properties.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » baseClass](#) class

Derived Classes

- [bibus » account](#)
- [bibus » adaptiveAnalyticsService](#)
- [bibus » adminFolder](#)
- [bibus » agentService](#)
- [bibus » agentTaskDefinition](#)
- [bibus » aliasLocation](#)
- [bibus » aliasRoot](#)
- [bibus » annotation](#)
- [bibus » annotationFolder](#)
- [bibus » annotationService](#)
- [bibus » archiveLocation](#)
- [bibus » baseAgentDefinition](#)
- [bibus » baseDataIntegrationTask](#)
- [bibus » baseDataMovementTask](#)
- [bibus » basePowerPlay8Report](#)
- [bibus » basePowerPlayClass](#)
- [bibus » baseReport](#)
- [bibus » baseROLAPDataSource](#)
- [bibus » baseRSSTask](#)
- [bibus » batchReportService](#)
- [bibus » capability](#)
- [bibus » catalog](#)
- [bibus » catalogFolder](#)
- [bibus » configuration](#)
- [bibus » configurationFolder](#)

- [bibus » contact](#)
- [bibus » content](#)
- [bibus » contentManagerCacheService](#)
- [bibus » contentManagerService](#)
- [bibus » contentTask](#)
- [bibus » credential](#)
- [bibus » dashboard](#)
- [bibus » dataAdvisorService](#)
- [bibus » dataIntegrationService](#)
- [bibus » dataMovementService](#)
- [bibus » dataSource](#)
- [bibus » dataSourceConnection](#)
- [bibus » dataSourceCredential](#)
- [bibus » dataSourceNameBinding](#)
- [bibus » dataSourceSignon](#)
- [bibus » deliveryService](#)
- [bibus » dimensionManagementService](#)
- [bibus » directory](#)
- [bibus » dispatcher](#)
- [bibus » distributionList](#)
- [bibus » document](#)
- [bibus » documentVersion](#)
- [bibus » drillPath](#)
- [bibus » eventManagementService](#)
- [bibus » EVService](#)
- [bibus » exportDeployment](#)
- [bibus » favoritesFolder](#)
- [bibus » folder](#)
- [bibus » graphicsService](#)
- [bibus » group](#)
- [bibus » history](#)
- [bibus » humanTask](#)
- [bibus » humanTaskService](#)
- [bibus » idVizService](#)
- [bibus » importDeployment](#)
- [bibus » indexDataService](#)
- [bibus » indexSearchService](#)
- [bibus » indexUpdateService](#)
- [bibus » indexUpdateTask](#)
- [bibus » jobDefinition](#)
- [bibus » jobService](#)
- [bibus » jobStepDefinition](#)
- [bibus » launchable](#)

- [bibus » logService](#)
- [bibus » memo](#)
- [bibus » metadataService](#)
- [bibus » metricsManagerService](#)
- [bibus » migrationService](#)
- [bibus » migrationTask](#)
- [bibus » mobileService](#)
- [bibus » model](#)
- [bibus » monitorService](#)
- [bibus » mruFolder](#)
- [bibus » namespace](#)
- [bibus » namespaceFolder](#)
- [bibus » package](#)
- [bibus » packageConfiguration](#)
- [bibus » pageDefinition](#)
- [bibus » pagelet](#)
- [bibus » pageletFolder](#)
- [bibus » periodical](#)
- [bibus » personalization](#)
- [bibus » personalizationFolder](#)
- [bibus » planningAdministrationConsoleService](#)
- [bibus » planningApplication](#)
- [bibus » planningDataService](#)
- [bibus » planningMacroTask](#)
- [bibus » planningRuntimeService](#)
- [bibus » planningTask](#)
- [bibus » planningTaskService](#)
- [bibus » portal](#)
- [bibus » portalPackage](#)
- [bibus » portalSkin](#)
- [bibus » portalSkinFolder](#)
- [bibus » portlet](#)
- [bibus » portletFolder](#)
- [bibus » portletProducer](#)
- [bibus » powerPlayService](#)
- [bibus » presentationService](#)
- [bibus » printer](#)
- [bibus » queryService](#)
- [bibus » queryServiceTask](#)
- [bibus » relationalMetadataService](#)
- [bibus » reportCache](#)
- [bibus » reportDataService](#)
- [bibus » reportService](#)

- [bibus » reportVersion](#)
- [bibus » repositoryService](#)
- [bibus » resource](#)
- [bibus » role](#)
- [bibus » root](#)
- [bibus » saCAMService](#)
- [bibus » schedule](#)
- [bibus » securedFeature](#)
- [bibus » securedFunction](#)
- [bibus » shortcut](#)
- [bibus » storedProcedureTask](#)
- [bibus » subscriptionFolder](#)
- [bibus » systemService](#)
- [bibus » tenant](#)
- [bibus » tenants](#)
- [bibus » uiProfile](#)
- [bibus » uiProfileFolder](#)
- [bibus » URL](#)
- [bibus » visualization](#)
- [bibus » webServiceTask](#)

Properties

This class has the following properties.

defaultDescription

Specifies the description of the object that best matches the description expressed in the user's preferred language. This is the content language specified by the [bibus » account » contentLocale](#) property. The user selects this language in the user preferences.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- must contain no more than 1024 characters
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

defaultScreenTip

Specifies the screen tip of the object that best matches the screen tip expressed in the user's preferred language. This is the content language specified by the [bibus » account » contentLocale](#) property. The user selects this language in the user preferences.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters
- is searchable

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

description

Specifies a description for this object.

This property

- is of type `string`
 - is encoded as type `tns:multilingualStringProp`
- must contain no more than 1024 characters
- is multilingual
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

displaySequence

Specifies a non-negative integer that determines the order in which the child objects are displayed.

The display order begins with 0. If multiple objects share the same value, an arbitrary order for these objects is determined by the implementation.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

hidden

Specifies whether an object is hidden. This property alone does not control whether an object is shown in the portal. The value of this property is one of the factors in an algorithm that determines whether an object is shown in the portal.

The result of the algorithm is returned as the value of the [bibus](#) » [uiClass](#) » [shown](#) property.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`

New in Version 8.4 – “Hiding Objects in the Portal” on page 1883

This property is used in the calculation of the [bibus](#) » [uiClass](#) » [shown](#) property.

Related information:

[“Controlling the Visibility of Objects” on page 61](#)

[bibus](#) » [uiClass](#) » [shown](#) property

iconURI

Specifies the location of the alternate icon. If empty, the default Cognos8 icon is used. The location can be an image name alone or an image name preceded by a relative path.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- can contain at least 4095 characters

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.3 – “Object Icons” on page 1914

This property was added.

screenTip

Specifies a description that can appear in the Web portal when the user's pointer passes over the object.

This property

- is of type `string`
 - is encoded as type `tns:multilingualStringProp`
- can contain at least 255 characters
- is multilingual
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

shown

Indicates whether the object should be visible in the portal. Its value is calculated by an algorithm that uses the following as inputs:

- the `hidden` property
- the `showHiddenObjects` preference
- the `canUseShowHiddenObjectsPreference` capability
- the `owner` property of the object

Use the predicate `[@shown=' true ']` in the search path to query only those objects that should be shown to the user.

The following table shows how the inputs to the algorithm are combined to produce a value for the `shown` property. The inputs are listed in the first four columns. The dashes indicate values that can be either true or false without affecting the value of `shown`.

Table 220. Example calculations of the shown property

<code>hidden</code>	<code>showHiddenObjects</code>	<code>canUseShowHiddenObjectsPreference</code>	<code>Owner?</code>	<code>shown</code>
true	true	true	—	true
true	true	false	true	true
true	true	false	false	false
true	false	—	—	false
false	—	—	—	true

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is read-only
- is searchable

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 — “Hiding Objects in the Portal” on page 1883

This property was added.

Related information:

[“Controlling the Visibility of Objects” on page 61](#)

[bibus](#) » [uiClass](#) » [hidden](#) property

viewed

Specifies whether the object was viewed by the user. Set to `true` when the object is viewed.

This property

- is of type `boolean`
 - is encoded as type `tns:booleanProp`
- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

[uiComponentEnumArrayProp](#)

Defines the array property class for the [bibus](#) » [uiComponentEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [package](#) » [userInterfaces](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [uiComponentEnum](#)
 - is encoded as type `tns:uiComponentEnumArray`

[uiComponentEnumProp](#)

Defines the simple property class for the [bibus](#) » [uiComponentEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [uiProfile](#) » [userInterface](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » uiComponentEnum](#)
is encoded as type `tns:uiComponentEnum`

uiProfile - deprecated

Defines a profile for an IBM Cognos Analytics user interface component, such as Reporting.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » uiProfileFolder](#)

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This class is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This class was added.

Properties

This class has the following properties.

configuration

Specifies the profile data.

This property contains data to be used only by IBM Cognossoftware. The structure of this data is subject to change without notice. Client applications should not attempt to interpret or modify this data.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLProp`

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus » baseClass](#)
has items that must be of class [bibus » account](#), [bibus » contact](#), [bibus » distributionList](#), [bibus » group](#), or [bibus » role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

userInterface

Specifies the user interface that can use this profile.

This property

- is of type `bibus » uiComponentEnum`

is encoded as type `tns:uiComponentEnumProp`

- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “[userInterface property of the uiProfile class](#)” on page 1897

This property is now intrinsic.

uiProfileFolder - deprecated

Contains profiles for IBM Cognos Analytics user interface components, such as Reporting.

We recommend that you represent instances of this type as non-leaf nodes in a tree structure.

This class

- inherits properties from the `bibus » uiClass` class

Container Information

Contains instances of the following classes

- `bibus » uiProfile`
- `bibus » uiProfileFolder`

Contained by instances of the following classes

- `bibus » configuration`
- `bibus » uiProfileFolder`

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This class is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This class was added.

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type `bibus » baseClass`
 - has items that must be of class `bibus » account`, `bibus » contact`, `bibus » distributionList`, `bibus » group`, or `bibus » role`
 - is encoded as type `tns:baseClassArrayProp`
- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`

items

Contains the child objects for this object.

unboundedEndRangeParmValueItem

Use this class to define a range with an unspecified end value.

Includes values greater than, or greater than or equal to, the start value.

This class

- inherits properties from the `bibus » parmValueItem` class

Properties

This class has the following properties.

start

Specifies the start, or minimum, value of the range. If inclusive, the value is within the range. If exclusive, the value is outside of the range.

This property

- is of type `bibus » simpleParmValueItem`
 - is encoded as type `tns:simpleParmValueItem`

unboundedStartRangeParmValueItem

Use this class to define a range with an unspecified start value.

Includes values either less than, or less than or equal to, the end value.

This class

- inherits properties from the `bibus » parmValueItem` class

Properties

This class has the following properties.

end

Specifies the end, or maximum, value of the range. If inclusive, the value is within the range. If exclusive, the value is outside of the range.

This property

- is of type [bibus » simpleParmValueItem](#)
is encoded as type `tns:simpleParmValueItem`

updateOptions

Defines the options you can specify for the [content » update\(objects, options\)](#) method.

References

Used by the following method parameters:

- [content » update\(objects, options\) » options](#)
- [drillThrough » updateDrillPath\(object, options\) » options](#)
- [report » update\(object, options\) » options](#)

Properties

This class has the following properties.

ignoreInvalidObjectReference

Specifies how invalid object references in ID-based reference properties are processed by Content Manager. A reference is invalid if it refers to a non-existent object.

If this property is set to `false`, Content Manager generates a fault if an object contains an invalid object reference in an ID-based reference property.

If this property is set to `true`, any invalid object reference in an ID-based reference property is automatically ignored by Content Manager when updating the content store.

This property

- is of type `boolean`
is encoded as type `xs:boolean`
- has a default value of `false`

returnProperties

Specifies the list of properties returned for each updated object.

If no properties are specified, the values for the [defaultName](#), [searchPath](#), and [storeId](#) properties are returned.

If properties are specified, only the values for the specified properties are returned. For example, if you specify [creationTime](#), only the value for this property is returned. If you also want the value for the [searchPath](#) property, you must specify it as well.

This property

- is an array of type [bibus » propEnum](#)
is encoded as type `tns:propEnumArray`

updateTenantIDRecursive

Specifies whether to update the [bibus » baseClass » tenantID](#) property for an object and all its descendant objects when the [content » update\(objects, options\)](#) method is used and the [bibus » baseClass » tenantID](#) property is present. This option is restricted to system administrators. If the

user is not a system administrators, the `bibus » baseClass » tenantID` property is ignored, regardless of whether the property is used.

During the update process, Content Manager checks the parent object's `tenantID` for consistency:

- If the parent object's `tenantID` is public, an empty string, Content Manager will allow the `tenantID` to be updated to public or any other `tenantID`, such as "tenant1".
- If the parent object's `tenantID` is not public, Content Manager will allow the `tenantID` to be updated only to the same `tenantID` of its parent.

Content Manager also checks the descendant objects' `tenantID` for consistency:

- If updating a non-public tenant, all descendants must have the same (original) `tenantID` as the object being updated.
- If updating a public tenant to a non-public `tenantID`, all descendants must be either public or have the same `tenantID` as the object being updated.

This property

- is of type `boolean`
is encoded as type `xs:boolean`
- has a default value of `false`

New in Version 10.2.0 – “Support for multi-tenancy” on page 1838

This property was added.

uriValue

Contains a URI/value pair.

References

Used by the following properties:

- `bibus » systemMetricThresholds » properties`
- `bibus » uriValueArrayProp » value`
- `bibus » uriValueProp » value`

What's new

New in Version 8.3 – “System Metrics” on page 1908

This class was added.

Properties

This class has the following properties.

uri

Specifies the name of the value as a URI.

This property

- is of type `bibus » systemMetricThresholdsPropertyEnum`
is encoded as type `xs:string`

value

Specifies the value corresponding to the `uri`.

This property

- is of type `string`
is encoded as type `xs:string`

uriValueArrayProp

Defines the array property class for the [bibus » uriValue](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » systemMetricThresholds » properties](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » uriValue](#)
is encoded as type `tns:uriValueArray`

uriValueProp

Defines the simple property class for the [bibus » uriValue](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » uriValue](#)
is encoded as type `tns:uriValue`

URL

Provides information about an external resource, such as a web page.

For example, a user can create a URL object in IBM Cognos Connection to point to a web site.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus](#) » [shortcutRSSTask](#) » [link](#)
- [bibus](#) » [shortcut](#) » [target](#)

Container Information

Contained by instances of the following classes

- [bibus](#) » [dashboard](#)
- [bibus](#) » [favoritesFolder](#)
- [bibus](#) » [folder](#)
- [bibus](#) » [mruFolder](#)
- [bibus](#) » [package](#)

Properties

This class has the following properties.

contact

Refers to a person or group of people who are responsible for the object.

This property

- is an array of type [bibus](#) » [baseClass](#)

has items that must be of class [bibus](#) » [account](#), [bibus](#) » [contact](#), [bibus](#) » [distributionList](#), [bibus](#) » [group](#), or [bibus](#) » [role](#)

is encoded as type `tns:baseClassArrayProp`

- refers to other objects in the content store using an id-based search path
- must have at most 1 item

contactEMail

Specifies the email address of the person or group of people who are responsible for the object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

contentType

Reserved.

This property

- is of type `string`

is encoded as type `tns:stringProp`

New in Version 10.1.0 – “Content Pane Improvements” on page 1862

This property was added.

packageBase

Specifies the path to the ancestor package. The format of this property is the same as the format of the [bibus](#) » [baseClass](#) » [searchPath](#) property.

If the object is not a descendant of a package, the value of this property is the path to the root object.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- is read-only
- must contain no more than 4096 characters
- can be acquired from a containing object

New in Version 8.3 – “Package Hierarchies” on page 1913

This property is deprecated. Use the [bibus](#) » [baseClass](#) » [ancestors](#) property instead.

uri

Specifies the Uniform Resource Identifier (URI) of the external resource. For example, set the URI property to `www.cognos.com` to point to the IBM Cognos corporate Web site, or set the URI property to `mailto:webmaster@cognos.com` to send an email to the IBM Cognos Web master.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- can contain at least 4095 characters

urlRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus](#) » [URL](#) object.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus](#) » [baseRSSTask](#) class

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `urlRSSTask` class.

<i>Table 221. Services and methods for the urlRSSTask class.</i>			
Action	Mode	Service	Method
Run	All	deliveryService	asynch » run(objectPath, parameterValues, options)

Properties

This class has the following properties.

link

Refers to the resource represented by the URL created by this task.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`

userCapabilityCache

Defines the structure used to cache the set of globally granted user capabilities in the BI Bus API header.

References

Used by the following properties:

- [bibus](#) » [biBusHeader](#) » [userCapabilityCache](#)

What's new

New in Version 8.4 — “Object Capabilities” on page 1885

This class was added.

Properties

This class has the following properties.

signature

Specifies the digital signature of [bibus](#) » [userCapabilityCache](#) » [userCapabilities](#) property. This property is used to determine if the user capability cache has been corrupted by a non- IBM Cognos software component.

This property

- is of type `base64Binary`
is encoded as type `xs:base64Binary`

userCapabilities

Contains the user's global capabilities.

This property contains the same value as [bibus](#) » [session](#) » [userCapabilities](#) property.

This property

- is an array of type [bibus](#) » [userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumArray`

Related information:

[“Managing Capabilities” on page 51](#)

userCapabilityEnumArrayProp

Defines the array property class for the [bibus](#) » [userCapabilityEnum](#) enumeration set.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [content](#) » [effectiveUserCapabilities](#)
- [bibus](#) » [folder](#) » [effectiveUserCapabilities](#)
- [bibus](#) » [package](#) » [effectiveUserCapabilities](#)
- [bibus](#) » [content](#) » [userCapabilities](#)
- [bibus](#) » [folder](#) » [userCapabilities](#)

- [bibus » package » userCapabilities](#)
- [bibus » session » userCapabilities](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnumArray`

userCapabilityEnumProp

Defines the simple property class for the [bibus » userCapabilityEnum](#) enumeration set.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » securedFeature » userCapability](#)
- [bibus » securedFunction » userCapability](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnum`

userCapabilityPermission

Defines whether a user capability is granted or denied in a [bibus » userCapabilityPolicy](#) object.

References

Used by the following properties:

- [bibus » userCapabilityPolicy » permissions](#)

What's new

New in Version 8.4 — “Object Capabilities” on page 1885

This class was added.

Properties

This class has the following properties.

access

Specifies whether the user capability is granted or denied.

This property

- is of type [bibus » accessEnum](#)
is encoded as type `tns:accessEnum`

userCapability

Specifies the user capability.

This property

- is of type [bibus » userCapabilityEnum](#)
is encoded as type `tns:userCapabilityEnum`

userCapabilityPolicy

Defines a user capability policy that determines the set of user capabilities granted to or denied to an [bibus » account](#), [bibus » group](#) or [bibus » role](#) for the containing object.

References

Used by the following properties:

- [bibus » content » userCapabilityPolicies](#)
- [bibus » folder » userCapabilityPolicies](#)
- [bibus » package » userCapabilityPolicies](#)
- [bibus » userCapabilityPolicyArrayProp » value](#)
- [bibus » userCapabilityPolicyProp » value](#)

What's new

New in Version 8.4 — “Object Capabilities” on page 1885

This class was added.

Properties

This class has the following properties.

permissions

Specifies the set of user capabilities that apply to the [bibus » account](#), [bibus » group](#) or [bibus » role](#) for the containing object.

This property

- is an array of type [bibus » userCapabilityPermission](#)
is encoded as type `tns:userCapabilityPermissionArray`

securityObject

Specifies the [bibus » account](#), [bibus » group](#) or [bibus » role](#) for which the user capability policy is defined.

This property

- is of type [bibus » baseClass](#)
is encoded as type `tns:baseClass`

userCapabilityPolicyArrayProp

Defines the array property class for the [bibus](#) » [userCapabilityPolicy](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

References

Used by the following properties:

- [bibus](#) » [content](#) » [userCapabilityPolicies](#)
- [bibus](#) » [folder](#) » [userCapabilityPolicies](#)
- [bibus](#) » [package](#) » [userCapabilityPolicies](#)

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus](#) » [userCapabilityPolicy](#)
is encoded as type `tns:userCapabilityPolicyArray`

userCapabilityPolicyProp

Defines the simple property class for the [bibus](#) » [userCapabilityPolicy](#) class.

This class

- inherits properties from the [bibus](#) » [baseProp](#) class

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type [bibus](#) » [userCapabilityPolicy](#)
is encoded as type `tns:userCapabilityPolicy`

userInterfaceProfile

Contains Reporting user interface profiles.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus](#) » [uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus](#) » [catalogFolder](#)

What's new

New in Version 10.2.2 – “Reporting user interface profiles” on page 1825

This class was added.

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to [MIME](#). The encoding parameter varies from method to method; see the [bibus](#) » [addOptions](#) » [dataEncoding](#) property and the [bibus](#) » [queryOptions](#) » [dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. In this case, the media type is "application/octet-stream" since the data is binary base64-encoded.

This property

- is of type `string`
is encoded as type `tns:stringProp`
- can contain at least 255 characters

userPreferenceVar

Defines the class for describing user preference values.

The value of the [bibus » account » useAccessibilityFeatures](#) property is stored using this class.

Product locale and content locale are two key user preferences. Product locale is the locale used for the user interface, messages, help text, and online documentation.

Content locale is the locale used to render customer content and report output. The rendering of the names of objects in the content store, including models, is performed using the content locale, but the rendering of property names is performed using the product locale.

References

Used by the following properties:

- [bibus » biBusHeader » userPreferenceVars](#)

Properties

This class has the following properties.

name

Identifies the user preference.

This property

- is of type `string`
is encoded as type `xs:string`

value

Specifies the value assigned to the user preference.

This property

- is of type `string`
is encoded as type `xs:string`

validateOption

Defines the abstract base class for all validate option classes.

This class

- is an abstract type used to define properties that are inherited by other types
- inherits properties from the [bibus » option](#) class

Derived Classes

- [bibus » validateOptionBoolean](#)
- [bibus » validateOptionHint](#)
- [bibus » validateOptionValidateSeverity](#)

Properties

This class has the following properties.

name

Identifies the validate option.

This property

- is of type [bibus » validateOptionEnum](#)
is encoded as type `tns:validateOptionEnum`

validateOptionBoolean

Defines boolean values for the validate options.

This class

- inherits properties from the [bibus » validateOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the validate option.

This property

- is of type `boolean`
is encoded as type `xs:boolean`

validateOptionHint

Defines [bibus » validateHintEnum](#) values for the validate options.

This class

- inherits properties from the [bibus » validateOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the validate option.

This property

- is of type [bibus » validateHintEnum](#)
is encoded as type `tns:validateHintEnum`

validateOptionValidateSeverity

Defines [bibus » validateSeverityEnum](#) values for the validate options.

This class

- inherits properties from the [bibus » validateOption](#) class

Properties

This class has the following properties.

value

Specifies the value for the validate option.

This property

- is of type [bibus » validateSeverityEnum](#)
is encoded as type `tns:validateSeverityEnum`

visualization

Contains elements required to generate a visualization in a report.

A visualization is a visual representation of data. For example, a map, a network diagram, or a bar graph. A visualization object in the content store is a module that allows authors to use visualizations in their reports.

We recommend that you represent instances of this type as a set of properties of its containing object. Users may access this object by opening a property sheet or launching an editor.

This class

- inherits properties from the [bibus » uiClass](#) class

Container Information

Contained by instances of the following classes

- [bibus » catalogFolder](#)

What's new

New in Version 10.2.1 – [“Visualization support” on page 1831](#)

This class was added.

Properties

This class has the following properties.

data

Contains the MIME data. The data format is specified in the [dataType](#) property.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to [MIME](#). The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type `base64Binary`
 - is encoded as type `tns:base64BinaryMIMEProp`

dataPathInfo

Contains the URI for the binary data for this object.

This property

- is of type `anyURI`
 - is encoded as type `tns:anyURIProp`
- is read-only

dataSize

Specifies the number of bytes of data.

This property

- is of type `nonNegativeInteger`
 - is encoded as type `tns:nonNegativeIntegerProp`
- is read-only

dataType

Specifies the media type of the value of property data. The media type is expressed as a type/subtype pair. For example, `image/jpeg` identifies images in JPEG format using JFIF encoding, and `text/plain` identifies plaintext.

This property

- is of type `string`
 - is encoded as type `tns:stringProp`
- can contain at least 255 characters

webServiceTask

Defines the class that describes a task that calls a web service operation.

We recommend that you represent instances of this type as leaf nodes in a tree structure. These objects can be manipulated independent of their containing object.

This class

- inherits properties from the [bibus » uiClass](#) class

References

Used by the following properties:

- [bibus » asyncDetailEventRecord » runnable](#)
- [bibus » eventRecord » runnable](#)
- [bibus » agentTaskDefinition » taskObject](#)

Related Services and Methods

This can be used with the following services:

The following table lists the services and methods for the `webServiceTask` class.

Action	Mode	Service	Method
Run	All	agentService	async » run(objectPath, parameterValues, options)

Container Information

Contains instances of the following classes

- [bibus » history](#)

Contained by instances of the following classes

- [bibus » authoredAgentDefinition](#)

Properties

This class has the following properties.

bindingName

Specifies the binding that defines the operation. The binding provides protocol-specific information for an operation.

This property

- is of type `string`
is encoded as type `tns:stringProp`

bulkEvents

Specifies that multiple events can be passed to the web service in a single call.

This property

- is of type `boolean`
is encoded as type `tns:booleanProp`

history

Contains the history for the object.

The default retention rule for histories is to keep 5 versions.

This property

- uses the [retentions](#) property to manage contained instances of [bibus](#) » [history](#)

inputMessageName

Specifies the name of the input message for the operation.

This property

- is of type `string`
is encoded as type `tns:stringProp`

operationName

Specifies the name of the Web service operation.

This property

- is of type `string`
is encoded as type `tns:stringProp`

options

Contains the set of options for this object.

Options can also be specified in other locations. For more information, see [“Specifying Options and Parameters” on page 66](#).

This property

- is an array of type [bibus](#) » [option](#)
is encoded as type `tns:optionArrayProp`

outputMessageName

Specifies the name of the output message for the operation.

This property

- is of type `string`

is encoded as type `tns:stringProp`

parameters

Contains the set of execution parameter values for this object.

Parameter values can also be specified in other locations. For more information, see [“Specifying Options and Parameters”](#) on page 66.

This property

- is an array of type `bibus » parameterValue`

is encoded as type `tns:parameterValueArrayProp`

retentions

Contains a set of retention rules, each associated with a specified class of child object that this parent can contain. These rules specify values for deleting child objects. For example, you can specify that Content Manager keep the latest three versions, or keep versions created in the last three months. IBM Cognos Connection limits retention configuration by either versions or duration.

The following rules are added to a new instance of this class when it is created by an IBM Cognos component:

Versioned Class	Maximum Age	Maximum Count	Ordered By
<code>bibus » history</code> class		5	<code>bibus » baseClass » creationTime</code> property

This property

- is an array of type `bibus » retentionRule`

is encoded as type `tns:retentionRuleArrayProp`

routingServerGroup

Specifies the name of the server group that should handle requests using this object.

This property is calculated using the `content » determineRouting(objectPaths)` method. The related objects are determined using information available to Content Manager about the object class. For example, a user's membership in a group or role can be derived from the user's Passport.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is read-only
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

routingServerGroupEdition

Specifies the edition of the service that should handle requests using this object.

This property

- is of type `string`

is encoded as type `tns:stringProp`

- is read-only
- must contain no more than 20 characters
- must contain exactly three decimal numbers consisting of one or more decimal digits separated by a decimal point ('.')

- is searchable
- can be retrieved without requiring read permission on the object; only traverse permission on the full path to the containing object is required

New in Version 8.4 – “Software Editions” on page 1893

This property was added.

serviceName

Specifies the name of the service.

This property

- is of type `string`
is encoded as type `tns:stringProp`

uri

Specifies the URI of the Web service to call.

This property

- is of type `anyURI`
is encoded as type `tns:anyURIProp`

xmlEncodedXML

Defines the type for XML and HTML documents.

This type is used to pass large XML and HTML documents to IBM Cognos Analytics components through the API. Characters that are reserved in XML and HTML, such as less-than (<) and greater-than (>), must be encoded or the operation may fail.

You can encode reserved characters by replacing them with the corresponding character entities, or by substituting the corresponding Unicode numeric representation.

To encode reserved characters using character entities, perform the following text substitutions on the document.

Search Text	Replacement Text
&	&
<	<
>	>
"	"
'	'

The substitution of & must be performed first.

To decode the document, perform the substitutions in reverse, replacing & with & last.

Note: This encoding and decoding applies to the raw Simple Object Access Protocol (SOAP) interface. If you are using a toolkit that performs these substitutions for you, do not explicitly encode or decode XML strings in your application.

This class

- inherits properties from the `string`

Derived Classes

- [bibus » dataSourceCommandBlock](#)
- [bibus » specification](#)

References

Used by the following properties:

- [bibus » rolapDataSource » aggregates](#)
- [bibus » portletProducer » binding](#)
- [bibus » uiProfile » configuration](#)
- [bibus » asynchDetailContext » context](#)
- [bibus » asynchDetailSelectionContext » context](#)
- [bibus » dashboard » context](#)
- [bibus » documentContent » context](#)
- [bibus » output » context](#)
- [bibus » pageletInstance » context](#)
- [bibus » personalization » context](#)
- [bibus » portlet » context](#)
- [bibus » portletInstance » context](#)
- [bibus » asynchDetailUnstructuredData » data](#)
- [bibus » asynchDetailReportValidation » defects](#)
- [bibus » portlet » handle](#)
- [bibus » pagelet » layout](#)
- [bibus » asynchDetailReportMetadata » metadata](#)
- [bibus » pagelet » metadata](#)
- [bibus » authoredPowerPlay8Report » powerPlay8Configuration](#)
- [bibus » content » powerPlay8Configuration](#)
- [bibus » folder » powerPlay8Configuration](#)
- [bibus » package » powerPlay8Configuration](#)
- [bibus » asynchDetailPromptPage » promptPage](#)
- [bibus » asynchDetailReportValidation » queryInfo](#)
- [bibus » portletProducer » registration](#)
- [bibus » annotation » selectionContext](#)
- [bibus » portletProducer » serviceDescription](#)
- [bibus » asynchDetailIndexData » sourceDocument](#)
- [bibus » launchable » specification](#)
- [bibus » portalSkin » specification](#)
- [bibus » resource » specification](#)
- [bibus » asynchRequest » stateData](#)
- [bibus » drillThroughOptionXMLEncodedXML » value](#)
- [bibus » genericOptionXMLEncodedXML » value](#)
- [bibus » pdfOptionXMLEncodedXML » value](#)

- [bibus » portalOptionXMLEncodedXML » value](#)
- [bibus » specificationOptionXMLEncodedXML » value](#)
- [bibus » xmlEncodedXMLArrayProp » value](#)
- [bibus » xmlEncodedXMLMIMEProp » value](#)
- [bibus » xmlEncodedXMLProp » value](#)

Used by the following method parameters:

- [authentication » logon\(credentials, roles\) » credentials](#)
- [dataSource » testDataSourceConnection\(connectionString, credentials\) » credentials](#)
- [drillThrough » convertDrillThroughContext\(inputContext, parameterValues, options\) » inputContext](#)
- [metadata » queryMetadata\(request\) » request](#)
- [metadata » updateMetadata\(request\) » request](#)

Used by the following method return values:

- [metadata » queryMetadata\(request\) » result](#)
- [metadata » updateMetadata\(request\) » result](#)
- [system » getFormatSamples\(name\) » result](#)

xmlEncodedXMLArrayProp

Defines the array property class for the [bibus » xmlEncodedXML](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

Properties

This class has the following properties.

value

Contains the values of the array property.

This property

- is an array of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXMLArray`

xmlEncodedXMLMIMEProp

Defines the simple MIME property class for the [bibus » xmlEncodedXML](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » documentContent » context](#)
- [bibus » output » context](#)
- [bibus » launchable » specification](#)
- [bibus » resource » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

The value may be placed in a MIME attachment.

This property may be encoded as an attachment in a method request message or response message if the encoding parameter is set to `MIME`. The encoding parameter varies from method to method; see the [bibus » addOptions » dataEncoding](#) property and the [bibus » queryOptions » dataEncoding](#) property for additional details.

This property

- is of type [bibus » xmlEncodedXML](#)
is encoded as type `tns:xmlEncodedXML`

xmlEncodedXMLProp

Defines the simple property class for the [bibus » xmlEncodedXML](#) class.

This class

- inherits properties from the [bibus » baseProp](#) class

References

Used by the following properties:

- [bibus » rolapDataSource » aggregates](#)
- [bibus » portletProducer » binding](#)
- [bibus » uiProfile » configuration](#)
- [bibus » dashboard » context](#)
- [bibus » pageletInstance » context](#)
- [bibus » personalization » context](#)
- [bibus » portlet » context](#)
- [bibus » portletInstance » context](#)
- [bibus » portlet » handle](#)
- [bibus » pagelet » layout](#)
- [bibus » pagelet » metadata](#)
- [bibus » authoredPowerPlay8Report » powerPlay8Configuration](#)
- [bibus » content » powerPlay8Configuration](#)
- [bibus » folder » powerPlay8Configuration](#)
- [bibus » package » powerPlay8Configuration](#)
- [bibus » portletProducer » registration](#)
- [bibus » annotation » selectionContext](#)
- [bibus » portletProducer » serviceDescription](#)
- [bibus » portalSkin » specification](#)

Properties

This class has the following properties.

value

Specifies the value of the property.

This property

- is of type bibus » xmlEncodedXML
is encoded as type `tns:xmlEncodedXML`

Chapter 16. Enumeration sets

If a property references an enumeration set, the value of the property must match one of the enumerated values.

These values are written in the English language. If your application uses a different language, we recommend that you write your application so that equivalent localized values appear for your users.

This section contains the possible values for properties that use enumeration sets.

Enumeration sets not included in the Cognos WSDL file

The enumeration sets in the following list are not included in the IBM Cognos WSDL file. Consult the documentation for a list of valid values for these enumeration sets and use String methods for setting and comparing these values in your applications.

- [bindingQualifierEnum](#) enumeration set
- [daysEnum](#) enumeration set
- [isolationLevelEnum](#) enumeration set
- [linearUnitEnum](#) enumeration set
- [monthsEnum](#) enumeration set
- [namespaceCapabilityEnum](#) enumeration set
- [outputFormatEnum](#) enumeration set
- [pageOrientationEnum](#) enumeration set
- [queryProcessingEnum](#) enumeration set
- [runStatusEnum](#) enumeration set
- [scheduleDailyPeriodEnum](#) enumeration set
- [scheduleEndTypeEnum](#) enumeration set
- [scheduleTypeEnum](#) enumeration set
- [sequencingEnum](#) enumeration set
- [usageEnum](#) enumeration set
- [weeksEnum](#) enumeration set

New in Version 10.2.2 – “Documentation Updates” on page 1827

This topic was added.

Enumeration Sets in Java

Enumeration sets are implemented as classes in the Java toolkit. Each of the values listed for a particular enumeration set is represented as a field of the corresponding class in the Java toolkit. Some examples are listed in the following table.

Enumeration Set	Value	Java Class	Field
bibus » encodingEnum	base64	EncodingEnum	base64
bibus » runOptionEnum	verticalElements	RunOptionEnum	verticalElements
bibus » orderEnum	ascending	OrderEnum	ascending

accessEnum

Defines the access values for security policies.

References

Used by the following properties:

- [bibus](#) » [permission](#) » [access](#)
- [bibus](#) » [userCapabilityPermission](#) » [access](#)

Members

deny

Specifies that the user, group, or role is denied the permission. If the grant and deny permissions are in conflict, access to the entry is always denied. Therefore, if a user is a member of two groups, one of which is granted access to an object, and one of which is denied access, the user is denied access.

To avoid conflicts between granting and denying access, you can remove the [bibus](#) » [permission](#) object for the user, group, or role rather than denying access.

grant

Specifies that the user, group, or role has the permission specified for the object.

agentNotificationStatusEnum

Defines the object alert status values.

References

Used by the following properties:

- [bibus](#) » [asynchDetailAgentNotificationStatus](#) » [status](#)

What's new

New in Version 8.3 — “[Report Email Alerts](#)” on page 1900

This enumeration set is now used to indicate the alert status of [bibus](#) » [baseReport](#) objects.

Members

disabled

Specifies that the object does not support alerts.

off

Specifies that the user is not on the alert list for the object.

on

Specifies that the user is on the alert list for the object.

agentOptionEnum

Provides the list of agent options.

References

Used by the following properties:

- [bibus](#) » [agentOption](#) » [name](#)

Members

allowAnnotations

Specifies whether the hot list can be annotated by agent consumers.

To specify this option, use the [bibus](#) » [agentOptionBoolean](#) class.

Default: false

New in Version 8.4 — “Report Output Annotations” on page 1895

This value was added.

availableAsEmailAttachment

Specifies that output produced by the agent task should be made available for use as attachments in subsequent email tasks.

To specify this option, use the [bibus](#) » [agentOptionBoolean](#) class.

Default: false

agentOutputEnum

Defines the agent output types.

References

Used by the following properties:

- [bibus](#) » [shortcutAgentRSSTask](#) » [link](#)
- [bibus](#) » [agentOutputEnumProp](#) » [value](#)

Members

eventList

The agent event list.

hotList

The agent hot list.

archiveConflictResolutionEnum

Provides the list of archive conflict resolution choices.

References

Used by the following properties:

- [bibus](#) » [archiveOptionConflictResolution](#) » [value](#)

What's new

New in Version 8.3 — “Saving Report Output to File System” on page 1901

This enumeration set was added.

Members

appendDateTime

Specifies that if a file name conflict occurs in the archive location, the date and time will be appended to the file name stub as specified by the [filenameStub](#) option.

appendSequenceNumber

Specifies that if a file name conflict occurs in the archive location, a sequence number will be appended to the file name stub as specified by the [filenameStub](#) option.

fail

Specifies that an attempt to archive the object will fail if a file name conflict occurs in the archive location.

replace

Specifies that if a file name conflict occurs in the archive location, the file specified by the [filenameStub](#) and [filenameExtension](#) options will be replaced.

archiveOptionEnum

Provides the list of archive options.

References

Used by the following properties:

- [bibus](#) » [archiveOption](#) » [name](#)

What's new

New in Version 8.3 — [“Saving Report Output to File System” on page 1901](#)

This enumeration set was added.

Members

archiveLocation

Specifies the [bibus](#) » [archiveLocation](#) to which the objects should be written. To specify this option, use the [bibus](#) » [archiveOptionSearchPathSingleObject](#) class.

conflictResolution

Specifies the action to take when a file name conflict occurs in the archive. To specify this option, use the [bibus](#) » [archiveOptionConflictResolution](#) class.

Default: [replace](#).

descriptor

Specifies the object descriptor. To specify this option, use the [bibus](#) » [archiveOptionDescriptor](#) class.

filenameExtension

Specifies the file name extension for the object in the archive. To specify this option, use the [bibus](#) » [archiveOptionString](#) class.

filenameStub

Specifies the stub file name for the object in the archive. The stub file name may be adorned with other texts to resolve naming conflicts. To specify this option, use the [bibus](#) » [archiveOptionString](#) class.

asynchDetailReportStatusEnum

Specifies the set of batch report service or report service-specific asynchronous conversation status values.

References

Used by the following properties:

- [bibus](#) » [asynchDetailReportStatus](#) » [status](#)

Members

prompting

Specifies that the primary or secondary request was processed successfully. The returned output is a prompt page.

responseReady

Specifies that the server is ready to provide the output. The server returns this status in response to the [asynch](#) » [wait\(conversation, parameterValues, options\)](#) method.

asynchOptionEnum

Provides the list of asynch options.

References

Used by the following properties:

- [bibus](#) » [asynchOption](#) » [name](#)

Members

alwaysIncludePrimaryRequest

Specifies that the primary request should always be included in the response message. Normally, to improve performance, the primary request, which can be quite large, is returned only when the status of the method is [complete](#).

To specify this option, use the [bibus](#) » [asynchOptionBoolean](#) class.

Default: `false`

attachmentEncoding

Specifies how attachments to the response are encoded. To specify this option, use the [bibus](#) » [asynchOptionEncoding](#) class. Possible choices are defined in the [bibus](#) » [encodingEnum](#) enumeration set.

Default: `base64`

continueOnError

Specifies whether an agent or job should continue to execute steps if a preceding step has failed. To specify this option, use the [bibus](#) » [asynchOptionBoolean](#) class.

When an agent or job runs in [sequential](#) mode, the default behaviour upon the failure of a step is to

- discontinue processing
- set the failed step object's [status](#) to [failed](#)
- set the [status](#) of all subsequent steps to [purged](#). No run history is created for these steps, so they will not show up in the portal.

Setting this value to `true` allows subsequent steps to continue, but the overall status of the run for the `bibus » baseAgentDefinition` or `bibus » jobDefinition` object is still set to `failed`.

When an agent or job runs in parallel mode, this setting has no effect.

Default: `false`

New in Version 8.3 – “Task Retry” on page 1909

This value is deprecated. Use `bibus » asynchOptionEnum » continueOnError` instead.

historyLocation

Specifies the location of the history object in the content store.

This option should not be directly populated by SDK applications. If a history is required for a task, SDK applications should send the request to the monitorService.

primaryWaitThreshold

Specifies the maximum amount of time, in seconds, the server can use to process the request before sending a response to the client. This option applies to primary requests and secondary requests other than the `asynch » wait(conversation, parameterValues, options)` method. To specify this option, use the `bibus » asynchOptionInt` class.

Use a value of 0 when you want the client to wait indefinitely.

Default: 7

restartHistoryLocation

Specifies the location of the `bibus » history` object in the content store that is used to determine the set of tasks that need to be retried and the set of tasks that will be run. This history object must be for a failed run, and the run will be continued from the point of failure. To specify this option, use the `bibus » asynchOptionSearchPathSingleObject` class.

New in Version 8.3 – “Task Retry” on page 1909

This value was added.

runContext

Identifies the events that form the context for the execution of this request. For example, when a job runs a report, the `eventID` of the job is passed to the request to run the report using this option. If the job was part of another job, the `eventID` of the job that started the job that runs the report is also present in the context. A service may determine the `eventID` of the task that started all the tasks that define the context by examining the first element in the `bibus » asynchOptionStringArray » value` property. A service may determine the `eventID` of the task that sent the request by examining the second-last element. The `eventID` assigned to the current request may be determined by examining the last element.

This option is generated by the monitorService. When the monitor service receives a request that contains this option, it appends the `eventID` of the current request to the `bibus » asynchOptionStringArray » value` property before forwarding the request to the appropriate service for execution.

This option is specified using the `bibus » asynchOptionStringArray` class. This option is generated and maintained by the monitorService service. SDK applications should not use options of this type.

secondaryWaitThreshold

Specifies the maximum amount of time, in seconds, the server can use to process the request before sending a response to the client. This option applies only to the secondary request `asynch » wait(conversation, parameterValues, options)`. To specify this option, use the `bibus » asynchOptionInt` class.

Use a value of 0 when you want the client to wait indefinitely.

Default: 30

skipTaskHistoryLocations

Specifies the locations of the [bibus » history](#) objects in the content store for the child tasks that should be skipped during a task retry. To specify this option, use the [bibus » asynchOptionSearchPathSingleObjectArray](#) class.

New in Version 8.3 – “Task Retry” on page 1909

This value was added.

asynchReplyStatusEnum

Specifies the set of asynchronous conversation status values.

References

Used by the following properties:

- [bibus » asynchReply » status](#)

Members

complete

Specifies that the primary or secondary request was processed successfully.

conversationComplete

Specifies that the primary or secondary request was processed successfully, and that the asynchronous conversation was terminated.

stillWorking

Specifies that the server is still processing the primary or secondary request. The server returns this status in response to the [asynch » wait\(conversation, parameterValues, options\)](#) method.

working

Specifies that the server is processing the primary or secondary request.

asynchStatusEnum

Specifies the set of asynchronous conversation status values.

For more information about IBM Cognos Analytics conversations, see [“Understanding the Asynchronous Conversation” on page 78](#).

Members

complete

Specifies that the primary or secondary request was processed successfully.

conversationComplete

Specifies that the primary or secondary request was processed successfully, and that the asynchronous conversation was terminated.

prompting

Specifies that the primary or secondary request was processed successfully. The returned output is a prompt page.

responseReady

Specifies that the server is ready to provide the output. The server returns this status in response to the [asynch » wait\(conversation, parameterValues, options\)](#) method.

stillWorking

Specifies that the server is still processing the primary or secondary request. The server returns this status in response to the [asynch » wait\(conversation, parameterValues, options\)](#) method.

working

Specifies that the server is processing the primary or secondary request.

auditLevelEnum

Defines the level of detail that is provided with audit logging.

References

Used by the following properties:

- [bibus » adaptiveAnalyticsService » aasAuditLevel](#)
- [bibus » configuration » aasAuditLevel](#)
- [bibus » configurationFolder » aasAuditLevel](#)
- [bibus » dispatcher » aasAuditLevel](#)
- [bibus » annotationService » ansAuditLevel](#)
- [bibus » configuration » ansAuditLevel](#)
- [bibus » configurationFolder » ansAuditLevel](#)
- [bibus » dispatcher » ansAuditLevel](#)
- [bibus » agentService » asAuditLevel](#)
- [bibus » configuration » asAuditLevel](#)
- [bibus » configurationFolder » asAuditLevel](#)
- [bibus » dispatcher » asAuditLevel](#)
- [bibus » batchReportService » brsAuditLevel](#)
- [bibus » configuration » brsAuditLevel](#)
- [bibus » configurationFolder » brsAuditLevel](#)
- [bibus » dispatcher » brsAuditLevel](#)
- [bibus » configuration » cmcsAuditLevel](#)
- [bibus » configurationFolder » cmcsAuditLevel](#)
- [bibus » contentManagerCacheService » cmcsAuditLevel](#)
- [bibus » dispatcher » cmcsAuditLevel](#)
- [bibus » configuration » cmsAuditLevel](#)
- [bibus » configurationFolder » cmsAuditLevel](#)
- [bibus » contentManagerService » cmsAuditLevel](#)
- [bibus » dispatcher » cmsAuditLevel](#)
- [bibus » configuration » dasAuditLevel](#)
- [bibus » configurationFolder » dasAuditLevel](#)
- [bibus » dataAdvisorService » dasAuditLevel](#)
- [bibus » dispatcher » dasAuditLevel](#)
- [bibus » configuration » dimsAuditLevel](#)
- [bibus » configurationFolder » dimsAuditLevel](#)
- [bibus » dimensionManagementService » dimsAuditLevel](#)
- [bibus » dispatcher » dimsAuditLevel](#)

- [bibus » configuration » disAuditLevel](#)
- [bibus » configurationFolder » disAuditLevel](#)
- [bibus » dataIntegrationService » disAuditLevel](#)
- [bibus » dispatcher » disAuditLevel](#)
- [bibus » configuration » dispatcherAuditLevel](#)
- [bibus » configurationFolder » dispatcherAuditLevel](#)
- [bibus » dispatcher » dispatcherAuditLevel](#)
- [bibus » configuration » dmsAuditLevel](#)
- [bibus » configurationFolder » dmsAuditLevel](#)
- [bibus » dataMovementService » dmsAuditLevel](#)
- [bibus » dispatcher » dmsAuditLevel](#)
- [bibus » configuration » dsAuditLevel](#)
- [bibus » configurationFolder » dsAuditLevel](#)
- [bibus » deliveryService » dsAuditLevel](#)
- [bibus » dispatcher » dsAuditLevel](#)
- [bibus » configuration » emsAuditLevel](#)
- [bibus » configurationFolder » emsAuditLevel](#)
- [bibus » dispatcher » emsAuditLevel](#)
- [bibus » eventManagementService » emsAuditLevel](#)
- [bibus » configuration » evsAuditLevel](#)
- [bibus » configurationFolder » evsAuditLevel](#)
- [bibus » dispatcher » evsAuditLevel](#)
- [bibus » EVService » evsAuditLevel](#)
- [bibus » configuration » gsAuditLevel](#)
- [bibus » configurationFolder » gsAuditLevel](#)
- [bibus » dispatcher » gsAuditLevel](#)
- [bibus » graphicsService » gsAuditLevel](#)
- [bibus » configuration » htsAuditLevel](#)
- [bibus » configurationFolder » htsAuditLevel](#)
- [bibus » dispatcher » htsAuditLevel](#)
- [bibus » humanTaskService » htsAuditLevel](#)
- [bibus » configuration » idsAuditLevel](#)
- [bibus » configurationFolder » idsAuditLevel](#)
- [bibus » dispatcher » idsAuditLevel](#)
- [bibus » indexDataService » idsAuditLevel](#)
- [bibus » configuration » idVizAuditLevel](#)
- [bibus » configurationFolder » idVizAuditLevel](#)
- [bibus » dispatcher » idVizAuditLevel](#)
- [bibus » idVizService » idVizAuditLevel](#)
- [bibus » configuration » issAuditLevel](#)
- [bibus » configurationFolder » issAuditLevel](#)
- [bibus » dispatcher » issAuditLevel](#)
- [bibus » indexSearchService » issAuditLevel](#)

- [bibus » configuration » iusAuditLevel](#)
- [bibus » configurationFolder » iusAuditLevel](#)
- [bibus » dispatcher » iusAuditLevel](#)
- [bibus » indexUpdateService » iusAuditLevel](#)
- [bibus » configuration » jsAuditLevel](#)
- [bibus » configurationFolder » jsAuditLevel](#)
- [bibus » dispatcher » jsAuditLevel](#)
- [bibus » jobService » jsAuditLevel](#)
- [bibus » configuration » mbsAuditLevel](#)
- [bibus » configurationFolder » mbsAuditLevel](#)
- [bibus » dispatcher » mbsAuditLevel](#)
- [bibus » mobileService » mbsAuditLevel](#)
- [bibus » configuration » mdsAuditLevel](#)
- [bibus » configurationFolder » mdsAuditLevel](#)
- [bibus » dispatcher » mdsAuditLevel](#)
- [bibus » metadataService » mdsAuditLevel](#)
- [bibus » configuration » misAuditLevel](#)
- [bibus » configurationFolder » misAuditLevel](#)
- [bibus » dispatcher » misAuditLevel](#)
- [bibus » migrationService » misAuditLevel](#)
- [bibus » configuration » mmsAuditLevel](#)
- [bibus » configurationFolder » mmsAuditLevel](#)
- [bibus » dispatcher » mmsAuditLevel](#)
- [bibus » metricsManagerService » mmsAuditLevel](#)
- [bibus » configuration » msAuditLevel](#)
- [bibus » configurationFolder » msAuditLevel](#)
- [bibus » dispatcher » msAuditLevel](#)
- [bibus » monitorService » msAuditLevel](#)
- [bibus » configuration » pacsAuditLevel](#)
- [bibus » configurationFolder » pacsAuditLevel](#)
- [bibus » dispatcher » pacsAuditLevel](#)
- [bibus » planningAdministrationConsoleService » pacsAuditLevel](#)
- [bibus » configuration » pdsAuditLevel](#)
- [bibus » configurationFolder » pdsAuditLevel](#)
- [bibus » dispatcher » pdsAuditLevel](#)
- [bibus » planningDataService » pdsAuditLevel](#)
- [bibus » configuration » ppsAuditLevel](#)
- [bibus » configurationFolder » ppsAuditLevel](#)
- [bibus » dispatcher » ppsAuditLevel](#)
- [bibus » powerPlayService » ppsAuditLevel](#)
- [bibus » configuration » prsAuditLevel](#)
- [bibus » configurationFolder » prsAuditLevel](#)
- [bibus » dispatcher » prsAuditLevel](#)

- [bibus » planningRuntimeService » prsAuditLevel](#)
- [bibus » configuration » psAuditLevel](#)
- [bibus » configurationFolder » psAuditLevel](#)
- [bibus » dispatcher » psAuditLevel](#)
- [bibus » presentationService » psAuditLevel](#)
- [bibus » configuration » ptsAuditLevel](#)
- [bibus » configurationFolder » ptsAuditLevel](#)
- [bibus » dispatcher » ptsAuditLevel](#)
- [bibus » planningTaskService » ptsAuditLevel](#)
- [bibus » configuration » qsAuditLevel](#)
- [bibus » configurationFolder » qsAuditLevel](#)
- [bibus » dispatcher » qsAuditLevel](#)
- [bibus » queryService » qsAuditLevel](#)
- [bibus » configuration » rdsAuditLevel](#)
- [bibus » configurationFolder » rdsAuditLevel](#)
- [bibus » dispatcher » rdsAuditLevel](#)
- [bibus » reportDataService » rdsAuditLevel](#)
- [bibus » configuration » reposAuditLevel](#)
- [bibus » configurationFolder » reposAuditLevel](#)
- [bibus » dispatcher » reposAuditLevel](#)
- [bibus » repositoryService » reposAuditLevel](#)
- [bibus » configuration » rmdsAuditLevel](#)
- [bibus » configurationFolder » rmdsAuditLevel](#)
- [bibus » dispatcher » rmdsAuditLevel](#)
- [bibus » relationalMetadataService » rmdsAuditLevel](#)
- [bibus » configuration » rsAuditLevel](#)
- [bibus » configurationFolder » rsAuditLevel](#)
- [bibus » dispatcher » rsAuditLevel](#)
- [bibus » reportService » rsAuditLevel](#)
- [bibus » configuration » saCAMAuditLevel](#)
- [bibus » configurationFolder » saCAMAuditLevel](#)
- [bibus » dispatcher » saCAMAuditLevel](#)
- [bibus » saCAMService » saCAMAuditLevel](#)
- [bibus » configuration » ssAuditLevel](#)
- [bibus » configurationFolder » ssAuditLevel](#)
- [bibus » dispatcher » ssAuditLevel](#)
- [bibus » systemService » ssAuditLevel](#)
- [bibus » auditLevelEnumProp » value](#)
- [bibus » contentTaskOptionAuditLevelEnum » value](#)
- [bibus » deploymentOptionAuditLevel » value](#)

Members

basic

The second-lowest level of logging, including user account management and data that is related to run-time usage of IBM Cognos Analytics.

full

The highest level of logging, including BI Bus log messages.

minimal

The lowest level of logging, including starting and stopping of services and run-time errors.

request

The third-lowest level of logging, including Simple Object Access Protocol (SOAP) requests and responses, and dispatcher load-balancing.

trace

The second-highest level of logging, adding IBM Cognos Analytics component debug logging.

automaticAggregateOptimizationEnum

Defines the possible values for the automaticAggregateOptimizationEnabled property of the rolapCubeConfiguration class..

References

Used by the following property

- [bibus » rolapCubeConfiguration » automaticAggregateOptimizationEnabled](#)

What's new

New in Version 10.2.2 – [New dynamic cube configuration properties](#)

This enumeration set was added.

Members

automaticAggregateOptimizationOff

Specifies that the set of in-memory aggregates that are loaded when the cube is running is not automatically optimized.

automaticAggregateOptimizationOn

Specifies that the set of in-memory aggregates that are loaded when the cube is running is automatically optimized.

baseAgentDefinitionActionEnum

Provides the list of portal actions for agents.

The user must have sufficient permissions and capabilities to perform the specified action.

References

Used by the following properties:

- [bibus » baseAgentDefinition » defaultPortalAction](#)
- [bibus » baseAgentDefinitionActionEnumProp » value](#)

What's new

New in Version 8.3 — [“Agent Default Portal Action” on page 1901](#)

This enumeration set was added.

Members

edit

Specifies that the agent should be edited with its associated studio.

run

Specifies that the agent should be run.

viewEventList

Specifies that the agent event list should be displayed, if it exists.

basePowerPlay8ReportActionEnum

Provides the list of portal actions for PowerPlay 8 reports.

This enumeration set specifies the possible actions that are available for a PowerPlay 8 report.

References

Used by the following properties:

- [bibus](#) » [basePowerPlay8Report](#) » [defaultPortalAction](#)
- [bibus](#) » [basePowerPlay8ReportActionEnumProp](#) » [value](#)

What's new

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

This enumeration set was added.

Members

edit

Specifies that the PowerPlay 8 report should be edited with the associated studio if the user has the necessary permissions and capabilities.

run

Specifies that the PowerPlay 8 report should be run if the user has the necessary permissions and capabilities.

viewOutput

Specifies that the PowerPlay 8 report output should be displayed, if it exists. Otherwise, the PowerPlay 8 report is run.

baseReportActionEnum

Provides the list of portal actions for reports.

The user must have sufficient permissions and capabilities to perform the specified action.

References

Used by the following properties:

- [bibus](#) » [drillPath](#) » [action](#)
- [bibus](#) » [baseReport](#) » [defaultPortalAction](#)
- [bibus](#) » [baseReportActionEnumProp](#) » [value](#)

Members

edit

Specifies that the report should be edited with its associated studio.

run

Specifies that the report should be run.

viewOutput

Specifies that report output should be displayed, if it exists. Otherwise, the report is run.

baseTextDirectionEnum

Defines the base text direction options with possible values defined in the enumeration set.

Default: [Auto](#)

What's new

New in Version 10.2.0 – “[Support for bi-directional languages](#)” on page 1835

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/baseTextDirectionEnum#Auto>

Automatic.

New in Version 10.2.0 – “[Support for bi-directional languages](#)” on page 1835

This value was added.

<http://developer.cognos.com/ceba/constants/baseTextDirectionEnum#LTR>

Specifies left to right.

New in Version 10.2.0 – “[Support for bi-directional languages](#)” on page 1835

This value was added.

<http://developer.cognos.com/ceba/constants/baseTextDirectionEnum#RTL>

Specifies right to left.

New in Version 10.2.0 – “[Support for bi-directional languages](#)” on page 1835

This value was added.

biDirectionalOptionEnum

Defines bi-directional language options.

What's new

New in Version 10.2.0 – [“Support for bi-directional languages” on page 1835](#)

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/biDirectionalOptionEnum#baseTextDirection>

Specifies the base text direction, defined in [bibus](#) » [baseTextDirectionEnum](#) enumeration set.

To specify this option, use an option of class [bibus](#) » [genericOptionAnyURI](#) class.

New in Version 10.2.0 – [“Support for bi-directional languages” on page 1835](#)

This value was added.

<http://developer.cognos.com/ceba/constants/biDirectionalOptionEnum#biDirectionalFeaturesEnabled>

Specifies if bi-directional features are enabled.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

New in Version 10.2.0 – [“Support for bi-directional languages” on page 1835](#)

This value was added.

bindingQualifierEnum

Defines the set of name binding qualifiers used by IBM Cognos Analytics.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [dataSourceNameBinding](#) » [qualifier](#)

Members

catalog

Use to select a catalog when you apply a name binding rule to a catalog.

schema

Use to select a schema when you apply a name binding rule to a schema.

table

Use to select a table when you apply a name binding rule to a table, view, or stored procedure.

classEnum

Defines the set of classes stored in the content store.

References

Used by the following properties:

- [bibus](#) » [deploymentDetail](#) » [deployedObjectClass](#)

- [bibus » asynchDetailIndexData » documentClass](#)
- [bibus » ancestorInfo » objectClass](#)
- [bibus » baseClass » objectClass](#)
- [bibus » deploymentObjectInformation » objectClass](#)
- [bibus » objectLink » objectClass](#)
- [bibus » repositoryRule » objectClass](#)
- [bibus » retentionRule » objectClass](#)
- [bibus » routingHintObject » objectClass](#)
- [bibus » migrationMapping » source](#)
- [bibus » migrationMapping » target](#)
- [bibus » classEnumArrayProp » value](#)
- [bibus » classEnumProp » value](#)
- [bibus » contentTaskOptionClassEnumArray » value](#)
- [bibus » deploymentOptionClassEnumArray » value](#)

Members

account

Contains the information used to define a user account.

adaptiveAnalyticsService

Defines run-time configuration parameters for the adaptive analytics service.

adminFolder

Contains the set of content objects used by administrators to automate repetitive tasks.

agentDefinition

Defines an agent as a series of tasks and defines parameters for those tasks.

agentDefinitionView

Defines a customized view of an agent definition.

agentOutputHotList

Defines output created by the comparison of two instances of report output.

agentService

Defines run-time configuration parameters for the [agentService](#).

agentState

Defines the transient state maintained during the execution of an [bibus » agentDefinition](#).

agentTaskDefinition

Specifies an object to be executed as part of an [bibus » agentDefinition](#).

agentTaskState

Defines the transient state maintained during the execution of an [bibus » agentTaskDefinition](#).

aliasRoot

Reserved.

analysis

Contains information used to define an IBM Cognos Business Intelligence Analysis specification.

annotation

Defines the class for annotations.

annotationFolder

Defines the class that contains annotations.

annotationService

Defines run-time configuration parameters for the annotation service.

archiveLocation

Defines an archive location.

batchReportService

Defines run-time configuration parameters for the [batchReportService](#).

cacheOutput

Provides report cache output.

capability

Contains objects that control access to product features.

catalog

Contains all [bibus](#) » [catalogFolder](#) class objects in the content store.

catalogFolder

Contains objects that store visualization data.

configuration

Defines the root object in the configuration hierarchy.

configurationFolder

Defines a configuration folder.

contact

Describes a contact.

content

Contains all packages and top-level folders.

contentManagerCacheService

Defines run-time configuration parameters for the content manager cache service.

contentManagerService

Defines run-time configuration parameters for the [contentManagerService](#).

contentTask

Defines a Content Manager task.

credential

Provides the information to allow authentication by a security provider.

dashboard

Contains information used to define an IBM Cognos Workspace workspace.

dataAdvisorService

Defines run-time configuration parameters for the data advisor service.

dataIntegrationService

Defines run-time configuration parameters for the [dataIntegrationService](#).

dataMovementService

Defines run-time configuration parameters for the [dataMovementService](#).

dataMovementTask

Defines a data movement task.

dataMovementTaskAlias

Defines the customization of a data movement task.

dataSet**dataSetFolder****dataSource**

Identifies an originator of data used by IBM Cognos Analytics for reports and queries.

dataSourceConnection

Defines a connection to the physical database used as the data source for IBM Cognos Analytics reports and queries.

dataSourceCredential

Stores the credentials required to sign on to the data source specified by the [dataSourceName](#) and [dataSourceConnectionName](#) properties.

dataSourceNameBinding

Defines a name binding rule for the connection to the database.

dataSourceSignon

Defines signon credentials for a database.

deliveryService

Defines run-time configuration parameters for the [deliveryService](#).

deploymentDetail

Defines information related to the deployment of an object.

dimensionManagementService

Defines run-time configuration parameters for the [dimensionManagementService](#).

directory

Defines the object that contains all directory information.

dispatcher

Defines run-time configuration parameters for the [dispatcher](#), as well as other services that are managed by the dispatcher.

distributionList

Defines a set of security objects.

document

This class defines the structure of documents created by IBM Cognos Analytics for Microsoft® Office.

documentContent

This class contains the document content.

documentVersion

This class contains the document content.

drillPath

Defines a drill-through path between a source and a target.

eventManagementService

Defines run-time configuration parameters for the [eventManagementService](#).

EVService

Defines run-time configuration parameters for the EVService .

exportDeployment

Defines the specification for a particular export deployment.

favoritesFolder

Contains objects that refer to the user's favorites.

folder

Contains the set of application content objects, such as queries, reports, job definitions, and other folders.

graphic

Contains a graphic image used in a report or query.

graphicsService

Defines run-time configuration parameters for the graphics service.

group

Contains group information, as stored in an external security provider or as defined in the Cognos namespace.

history

Provides information about the running of a task.

historyDetail

Provides a message generated during the running of a task.

historyDetailAgentService

Provides the location of the [bibus » agentOutputHotList](#).

historyDetailDataMovementService

Provides information related to the execution of a data movement task.

historyDetailDeploymentSummary

Provides summary information related to deployments.

historyDetailMigrationService

Defines information related to the execution of a migration task.

historyDetailRelatedHistory

Provides an association between the [bibus » history](#) instance of an invoked task with the [bibus » history](#) of the parent task.

historyDetailRelatedReports

Provides the information required to run a report that supplies the history for the task.

historyDetailReportService

Provides the location of the report output created in the content store during the running of a report.

historyDetailRequestArguments

Provides the options and parameter values used to run the task.

humanTask

Defines the class that describes a task that requires human interaction.

A task is defined as a notification, request, or other user-defined action, related to an issue.

humanTaskService

Defines run-time configuration parameters for the human task service.

identity**idVizService**

Reserved.

importDeployment

Defines the specification for a particular import deployment.

indexDataService

Defines run-time configuration parameters for the index data service .

indexSearchService

Defines run-time configuration parameters for the [indexSearchService](#).

indexUpdateService

Defines run-time configuration parameters for the [indexUpdateService](#).

indexUpdateTask

Reserved.

installedComponent

Contains information about an installed IBM Cognos Analytics component.

interactiveReport

Contains information that defines an IBM Cognos Active Report specification.

jobDefinition

Describes a job as a series of job steps and defines parameters for those job steps.

jobService

Defines run-time configuration parameters for the [jobService](#).

jobStepDefinition

Specifies an object to be executed as part of a job.

launchable

Defines a generic object that can be manipulated by a UI component.

logService

Defines run-time configuration parameters for the log service .

memo

Defines the class for email messages that can be sent as part of a job or by an agent.

metadataService

Defines run-time configuration parameters for the metadata service.

metricsDataSourceETLTask

Defines the metrics data source ETL task.

metricsExportTask

Defines the metrics export task.

metricsFileImportTask

Defines the metrics file import task.

metricsMaintenanceTask

Defines the metrics maintenance task.

metricsManagerService

Defines run-time configuration parameters for the metrics manager service .

migrationService

Defines run-time configuration parameters for the migrationService.

migrationTask

Defines a migration task.

mobileService

Defines run-time configuration parameters for the mobileService.

model

Contains a published IBM Cognos Framework Manager model.

modelView

Instances of this class are used to associate security policies with views stored in the IBM Cognos Framework Manager model.

monitorService

Defines run-time configuration parameters for the monitorService.

mrFolder

Contains shortcuts and URL objects that refer to the most recently used (MRU) objects.

namespace

Contains the information used to define a namespace.

namespaceFolder

Contains folders or organizational units, as stored in an external security directory.

nil

Defines an empty class, for situations in which the referenced object no longer exists in the content store, or cannot be retrieved due to security restrictions.

output

Provides report output.

package

Contains a model and all the folders, queries, reports, views, shortcuts, URLs, and job definitions associated with that model.

packageConfiguration

Defines package-specific configuration information.

page

Provides a single page of HTML report output.

pageDefinition

Describes the physical properties of a page.

pagelet

Represents a user interface area made up of a set of portlets and pagelets rendered by a layout specification.

pageletFolder

Contains the set of pagelets for a portal package.

pageletInstance

Defines an instance of a pagelet on a rendered page.

periodical

Defines a periodical as a set of documents created by a producer.

personalization

Contains information used to personalize an IBM Cognos Workspace workspace.

personalizationFolder

Contains the personalizations created by a user.

planningAdministrationConsoleService

Reserved.

planningApplication

Reserved.

planningDataService

Defines run-time configuration parameters for the `planningDataService` .

planningMacroTask

Defines a planning macro task.

planningRuntimeService

Reserved.

planningTask

Reserved.

planningTaskService

Reserved.

portal

Contains all `bibus » portalPackage` class objects.

portalPackage

Contains metadata related to a [bibus » portletProducer](#) and a hierarchy of [bibus » pagelet](#) class objects.

portalSkin

Defines the properties of an IBM Cognos Analytics portal skin.

portalSkinFolder

Defines the folder class for portal skins.

portlet

Contains information that defines a relationship with a Web services for Remote Portlets (WSRP) producer.

portletFolder

Contains the set of [bibus » portletProducer](#) objects for the containing [bibus » portalPackage](#) object.

portletInstance

Defines an instance of a [bibus » portlet](#) class object on a rendered page.

portletProducer

Contains information that defines a relationship with a Web services for Remote Portlets (WSRP) producer.

powerPlay8Report

Contains information used to define a PowerPlay report.

powerPlay8ReportView

Defines the customization of an PowerPlay report.

powerPlayCube

Defines the class that contains the information required to view a PowerPlay cube.

powerPlayReport

Defines the class that contains the information required to run an PowerPlay report.

powerPlayService

Defines run-time configuration parameters for the [powerPlayService](#).

presentationService

Defines run-time configuration parameters for the presentation service .

printer

Defines printer properties.

query

Contains information used to define a report specification.

queryService

Defines run-time configuration parameters for the [queryService](#).

queryServiceTask

Defines a query service task.

relationalMetadataService

Defines run-time configuration parameters for the [relationalMetadataService](#).

report

Contains information used to define a report specification.

reportCache

Contains the information required to improve report processing, such as cached prompt values.

reportDataService

Defines run-time configuration parameters for the report data service .

reportDataServiceAgentDefinition

Defines an agent as a series of tasks and defines parameters for those tasks.

reportService

Defines run-time configuration parameters for the [reportService](#).

reportTemplate

Contains information used to define an IBM Cognos Analytics report template.

reportVersion

Contains the information required to reproduce report output, such as prompt values and output formats.

reportView

Defines the customization of a query or report.

repositoryService

Reserved.

resource

Defines a class that contains information required to reference a resource not managed by the content store.

rolapDataSource

Identifies a base ROLAP cube data source.

rolapVirtualDataSource

Identifies a virtual ROLAP cube data source.

role

Contains the information used to define a role.

root

Defines the class of the object at the top of the hierarchy.

runTimeState

Provides information about the run-time state of IBM Cognos Analytics components.

saCAMService

Defines run-time configuration parameters for the standalone CAM service.

schedule

Defines a schedule for a recurring task.

securedFeature

Defines a product feature.

securedFunction

Defines a product function.

session

Contains the temporary objects for the session.

shortcut

Points to an object in another location.

shortcutAgentRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus](#) » [shortcut](#) object.

shortcutRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus](#) » [shortcut](#) object.

SQL

Contains the SQL generated during the execution of a report.

storedProcedureTask

Defines the class that describes a task that calls an RDBMS stored procedure.

subscriptionFolder

Contains the subscriptions created by a user.

systemMetricThresholds

Contains the information used to define thresholds for a system metric.

systemService

Defines run-time configuration parameters for the [systemService](#).

tenancy**tenant**

Defines the class for tenants in a multi-tenant environment.

tenants

Contains tenant objects in a multi-tenancy environment.

transientStateFolder

Contains objects that store temporary data required by services to maintain information between occurrences of object execution.

uiProfile

Defines a profile for an IBM Cognos Analytics user interface component, such as Reporting.

uiProfileFolder

Contains profiles for IBM Cognos Analytics user interface components, such as Reporting.

URL

Provides information about an external resource, such as a web page.

urlRSSTask

Defines the class that describes tasks that update an RSS channel using a [bibus](#) » [URL](#) object.

userInterfaceProfile

Contains Reporting user interface profiles.

visualization

Contains elements required to generate a visualization in a report.

webServiceTask

Defines the class that describes a task that calls a web service operation.

configurationDataEnum

Lists the properties of the [bibus](#) » [configurationData](#) class.

References

Used by the following method parameters:

- [system](#) » [getConfiguration\(properties\)](#) » [properties](#)

Members

bpmRestURI

Specifies the [bibus](#) » [configuration](#) » [bpmRestURI](#) property property.

New in Version 10.2.0 – “[IBM Business Process Server integration](#)” on page 1841

This value was added.

contentLocaleMap

Specifies the [contentLocaleMap](#) property.

defaultFont

Specifies the [defaultFont](#) property.

glossaryURI

New in Version 8.4 – “[Support for IBM® WebSphere® Business Glossary](#)” on page 1897

This value was added.

metadataInformationURI

New in Version 8.4 – “[Lineage Metadata](#)” on page 1891

This value was added.

productLocaleMap

Specifies the [productLocaleMap](#) property.

serverLocale

Specifies the [serverLocale](#) property.

serverTimeZone

Specifies the [serverTimeZone](#) property.

supportedContentLocales

Specifies the [supportedContentLocales](#) property.

supportedCurrencies

Specifies the [supportedCurrencies](#) property.

supportedFonts

Specifies the [supportedFonts](#) property.

supportedProductLocales

Specifies the [supportedProductLocales](#) property.

timeZones

Specifies the [timeZones](#) property.

conflictResolutionEnum

Defines the set of conflict resolution options available for use with IBM Cognos Analytics.

References

Used by the following properties:

- [bibus](#) » [deploymentOptionResolution](#) » [value](#)
- [bibus](#) » [migrationTaskOptionResolution](#) » [value](#)

Members

keep

In the event of a conflict, keep the original.

replace

In the event of a conflict, replace the original.

contentManagerQueryOptionEnum

Provides the list of content manager query options.

References

Used by the following properties:

- [bibus](#) » [contentManagerQueryOption](#) » [name](#)

Members

maximumDataBytes

Specifies the maximum number of bytes of data for binary properties in the result set. Use with [skipDataBytes](#) to partition a large binary property value.

For example, you can use this option to implement PDF byte-serving. Note that, for values of zero or less, all the data is returned.

To specify this option, use the [bibus](#) » [contentManagerQueryOptionInt](#) class.

Default: 0

maximumObjects

Specifies the maximum number of objects in the result set. Use with [skipObjects](#) to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. Note that, for values of zero or less, all selected objects are returned.

To specify this option, use the [bibus](#) » [contentManagerQueryOptionInt](#) class.

Default: 0

permission

Specifies that a fault will be thrown if the caller does not have the specified permission on any object selected by the `content » query(searchPath, properties, sortBy, options) » searchPath` parameter.

To specify this option, use the `bibus » contentManagerQueryString` class.

properties

Specifies the set of properties to be returned.

If you specify a reference property, properties of the referenced object can also be returned. By default, only the `searchPath` property is returned for each referenced object. If the reference property is ID-based, the `storeID` property is also returned.

If you specify the `referenceProperties` option, only the `searchPath` property is returned for the referenced object. You must then specify additional properties if you want to have them returned for the referenced object.

If the `data` property is specified and `attachmentEncoding` is specified as `MIME` or `MIMECompressed`, the data is returned in a MIME attachment. Otherwise, data is returned inline using `base64` encoding.

To specify this option, use the `bibus » contentManagerQueryOptionPropEnumArray` class.

Default: { `searchPath` }

referenceProperties

Specifies the properties to be retrieved for objects referenced by retrieved properties. For example, use this option to specify that the names and descriptions of the members of a group be returned.

To specify this option, use the `bibus » contentManagerQueryOptionRefPropArray` class.

schemaInfo

Specifies whether additional property information is returned with the property values.

To specify this option, use the `bibus » contentManagerQueryOptionBoolean` class.

Default: `false`

skipDataBytes

Specifies the number of data bytes to be skipped for binary properties in the result set. Use with `maximumDataBytes` to partition a large binary property value.

For example, you can use this option to implement PDF byte-serving. Note that a negative value indicates the number of bytes from the end of the data.

To specify this option, use the `bibus » contentManagerQueryOptionInt` class.

Default: 0

skipObjects

Specifies the number of objects to be skipped in the result set. Use with `maximumObjects` to partition a large result set.

For example, you can use this option to select the set of rendered items when scrolling in your Web portal. Note that a negative value indicates the number of objects from the end of the result set.

To specify this option, use the `bibus » contentManagerQueryOptionInt` class.

Default: 0

contentTaskOptionEnum

Provides the list of content task options.

References

Used by the following properties:

- [bibus](#) » [contentTaskOption](#) » [name](#)

Members

checkInternalInconsistencies

Reserved.

checkSecurityReferences

Specifies whether a content task should check references to security objects defined in external security providers. References are checked only if the session in which the task is run is authenticated with the external security provider used by the reference.

To specify this option, use the [bibus](#) » [contentTaskOptionBoolean](#) class.

Default: `false`

cleanupContent

Specifies whether a content task deletes objects that do not satisfy the retention rules that the [cleanupContentRetentionRules](#) option specifies.

To specify this option, use an option of class [bibus](#) » [contentTaskOptionBoolean](#).

Default: `false`.

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This value was added.

cleanupContentAuditLevel

Specifies the amount of content cleanup detail to record in the content task history.

To specify this option, use an option of class [bibus](#) » [contentTaskOptionAuditLevelEnum](#).

Default: `minimal`

Important: For this option to take effect, [cleanupContent](#) value must be set to `true`.

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This value was added.

cleanupContentContext

Specifies the set of objects that the [cleanupContent](#) option acts upon. All specified objects, and the objects they contain, are considered.

To specify this option, use the [bibus](#) » [contentTaskOptionSearchPathSingleObjectArray](#) class.

Default: All objects.

Important: For this option to take effect, [cleanupContent](#) value must be set to `true`.

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This value was added.

cleanupContentDoNotWriteToRepository

Specifies whether content should be written to an external repository. Use this option to override all repository rules specified by the containing [bibus](#) » [folder](#) class or [bibus](#) » [package](#) class.

To specify this option, use an option of class [bibus](#) » [contentTaskOptionBoolean](#).

Default: `false`.

Important: For this option to take effect, [cleanupContent](#) value must be set to `true`.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

cleanupContentRetentionRules

Specifies the retention rules that determine whether to delete an object. Objects that do not satisfy the conditions that are defined by a retention rule associated with the object's class are deleted.

To specify this option, use the `bibus » contentTaskOptionRetentionRuleArray` class.

Important: For this option to take effect, `cleanupContent` value must be set to `true`.

New in Version 10.1.0 – “Content Cleanup Content Manager Task” on page 1861

This value was added.

cleanupContentUpdateContextObjects

Specifies whether the retention rules of the objects identified by the `cleanupContentContext` option are replaced with the retention rules specified by the `cleanupContentRetentionRules` option.

To specify this option, use an option of class `bibus » contentTaskOptionBoolean`.

Default: `false`.

Important: For this option to take effect, `cleanupContent` value must be set to `true`.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

createContentStoreUtilizationInfo

Specifies whether a content task should create content store utilization information.

To specify `createContentStoreUtilizationInfo` use the `bibus » contentStoreUtilizationConfiguration` class.

repairInternalInconsistencies

Reserved.

repairSecurityReferences

Specifies whether a content task should check references to security objects defined in external security providers and repair them if they are invalid. References are checked only if the session in which the task is run is authenticated with the external security provider used by the reference.

A reference is repaired by removing it from the content store.

To specify this option, use the `bibus » contentTaskOptionBoolean` class.

Default: `false`

securityReferencesContext

Specifies the set of namespaces to be acted upon by the `checkSecurityReferences` and `repairSecurityReferences` options.

To specify this option, use the `bibus » contentTaskOptionSearchPathSingleObjectArray` class.

Default: All namespaces except the Cognos namespace are checked or repaired.

upgradeClasses

Specifies the classes of objects to be upgraded.

To specify this option, use the [bibus » contentTaskOptionClassEnumArray](#) class.

Default: An empty collection.

upgradeClassesContext

Specifies the set of objects to be acted upon by the [upgradeClasses](#) option. All specified objects, and the objects they contain, are considered for upgrade.

To specify this option, use the [bibus » contentTaskOptionSearchPathSingleObjectArray](#) class.

Default: All objects are upgraded.

upgradeClassesReferenceContext

Specifies the set of objects acted upon by the [upgradeClasses](#) option. All objects that reference the specified objects are considered for upgrade.

To specify this option, use the [bibus » contentTaskOptionSearchPathSingleObjectArray](#) class.

contextFormatEnum

Provides the list of context formats.

References

Used by the following properties:

- [bibus » contextOptionFormat » value](#)

What's new

New in Version 8.3 — “[Improved Context Metadata for Selection](#)” on page 1911

This enumeration set was added.

Members

initializer

Specifies that the context data should be formatted as a Javascript initializer.

XML

Specifies that the context data should be formatted as an XML fragment.

contextOptionEnum

Provides the list of context options.

References

Used by the following properties:

- [bibus » contextOption » name](#)

What's new

New in Version 8.3 — “[Improved Context Metadata for Selection](#)” on page 1911

This enumeration set was added.

Members

format

Specifies the context format. To specify this option, use the [bibus » contextOptionFormat](#) class.

selection

Identifies the context data to be returned. To specify this option, use the [bibus » contextOptionStringArray](#) class.

type

Specifies the context type. To specify this option, use the [bibus » contextOptionType](#) class.

contextTypeEnum

Provides the list of context types.

References

Used by the following properties:

- [bibus » contextOptionType » value](#)

What's new

New in Version 8.3 — “Improved Context Metadata for Selection” on page 1911

This enumeration set was added.

Members

reportService

Specifies context data as produced by the [reportService](#).

copyAccountOptionEnum

Defines the options for the [content » copyAccount\(sourceAccountPath, targetAccountPath, options\)](#) method .

References

Used by the following properties:

- [bibus » copyAccountOption » name](#)

Members

folders

Specifies whether to copy objects from the My Folders object associated with the source account.

To specify this option, use the [bibus » copyAccountOptionBoolean](#) class.

Default: false

pages

Specifies whether to copy objects from the My Pages object associated with the source account.

To specify this option, use the [bibus » copyAccountOptionBoolean](#) class.

Default: false

preferences

Specifies whether to copy account preferences.

To specify this option, use the [bibus](#) » [copyAccountOptionBoolean](#) class.

Default: false

dashboardOptionEnum

Provides the list of IBM Cognos Workspace options.

What's new

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/dashboardOptionEnum#showStartPage>

Specifies whether the IBM Cognos Workspace Start Page is displayed. To specify this option, use the [bibus](#) » [genericOptionBoolean](#) class.

Default: true

dataEnum

Provides the list of data option values.

References

Used by the following properties:

- [bibus](#) » [runOptionData](#) » [value](#)

Members

previewWithLimitedData

Specifies that IBM Cognos Analytics will retrieve limited data when running the report. The data returned is based on design mode filters defined in Framework Manager.

previewWithNoData

Specifies that IBM Cognos Analytics will not retrieve data when running the report. Artificial data is used instead of actual data from the data source.

runWithAllData

Specifies that IBM Cognos Analytics will retrieve data when running the report.

dataIntegrationTaskOptionEnum

Provides the list of data integration task options.

References

Used by the following properties:

- [bibus](#) » [dataIntegrationTaskOption](#) » [name](#)

Members

clearAuditLogs

Specifies whether a data integration task should clear the audit logs. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

New in Version 10.1.0 – “New Data Integration Service Task Option” on page 1853

This value was added.

clearCalendar

Specifies whether a data integration task should clear the fiscal calendar. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

clearHistory

Specifies whether a data integration task should clear numerical values. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

clearRejects

Specifies whether a data integration task should clear the staged data rejects tables. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

loadFromStagingTables

Specifies whether a data integration task should move data from the staging area into IBM Cognos Metrics Manager. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

processWatchListQueue

Specifies whether a data integration task should prepare and send pending emails from the IBM Cognos Metrics Manager watch list queue. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

rebuildIndexes

Specifies whether a data integration task should rebuild the IBM Cognos Metrics Manager search engine index files. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

recalculateMetrics

Specifies whether a data integration task should perform calculations on existing IBM Cognos Metrics Manager data. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

synchronizeUserInformation

Specifies whether a data integration task should locate the corresponding [bibus » account](#) object for every IBM Cognos Metrics Manager user. If a corresponding object is found, the two are synchronized. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

updateIndexes

Specifies whether a data integration task should update the IBM Cognos Metrics Manager search engine index files. To specify this option, use the [bibus » dataIntegrationTaskOptionBoolean](#) class.

Default: false

dataSourceCapabilityEnum

Lists the set of capabilities supported by data sources.

References

Used by the following properties:

- [bibus » baseROLAPDataSource » capabilities](#)
- [bibus » dataSource » capabilities](#)

What's new

New in Version 10.1.0 – “[IBM Cognos TM1 and IBM Cognos Workspace Integration](#)” on page 1867

This enumeration set was added.

New in Version 10.1.0 – “[Personal Packages](#)” on page 1863

This enumeration set was added.

New in Version 10.1.0 – “[IBM Cognos Content Manager/Enterprise Content Management Integration](#)” on page 1866

This enumeration set was added.

New in Version 10.1.1 – “[Support for IBM Cognos Content Archival!](#)” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

Members

<http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#atomContentProvider>

Specifies that the data source can provide content as an ATOM feed.

New in Version 10.1.0 – “[IBM Cognos TM1 and IBM Cognos Workspace Integration](#)” on page 1867

This value was added.

<http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#lobData>

Specifies that the data source can be used as a Line of Business (LOB) data repository

New in Version 10.2.2 – [My data sets](#)

This value was added.

<http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#personalPackages>

Specifies that the data source can be used to create a personal package. Data sources specified with this capability are made available when creating a package in the New Package wizard.

New in Version 10.1.0 – “[Personal Packages](#)” on page 1863

This value was added.

<http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#reporting>

Specifies that the data source can be used as a reporting data source.

<http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#repository>

Specifies that the data source can be used as a content management repository.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

daysEnum

Specifies the days of the week.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [schedule](#) » [monthlyRelativeDay](#)
- [bibus](#) » [schedule](#) » [yearlyRelativeDay](#)

Members

friday

Specifies the day Friday.

monday

Specifies the day Monday.

saturday

Specifies the day Saturday.

sunday

Specifies the day Sunday.

thursday

Specifies the day Thursday.

tuesday

Specifies the day Tuesday.

wednesday

Specifies the day Wednesday.

deliveryChannelEnum

Provides the list of delivery channels.

References

Used by the following properties:

- [bibus](#) » [deliveryOptionChannel](#) » [value](#)

Members

archive

Specifies that content is archived.

New in Version 8.3 – [“Saving Report Output to File System” on page 1901](#)

This value was added.

email

Specifies that content is delivered by email.

mobile

This enumeration value is not used.

New in Version 8.4 – [“Report Server deliver\(\) method” on page 1883](#)

This value is deprecated. Use the [bibus » runOptionEnum » mobile](#) value instead.

deliveryOptionEnum

Provides the list of delivery options.

References

Used by the following properties:

- [bibus » deliveryOption » name](#)

Members

bcc

Specifies the recipients on the bcc list. Recipients must be instances of the [bibus » account](#) class, the [bibus » contact](#) class, the [bibus » distributionList](#) class, the [bibus » group](#) class, or the [bibus » role](#) class.

To specify this option, use the [bibus » deliveryOptionSearchPathMultipleObjectArray](#) class.

bccAddress

Specifies the address of each recipient on the bcc list. To specify this option, use the [bibus » deliveryOptionAddressSMTPArray](#) class.

cc

Specifies the recipients on the cc list. Recipients must be instances of the [bibus » account](#) class, the [bibus » contact](#) class, the [bibus » distributionList](#) class, the [bibus » group](#) class, or the [bibus » role](#) class.

To specify this option, use the [bibus » deliveryOptionSearchPathMultipleObjectArray](#) class.

ccAddress

Specifies the address of each recipient on the cc list. To specify this option, use the [bibus » deliveryOptionAddressSMTPArray](#) class.

channel

Specifies the delivery channel for the content.

To specify this option, use the [bibus » deliveryOptionChannel](#) class.

Default: [email](#)

characterSetEncoding

Specifies the character set encoding to use for the memo. The default is UTF-8. Use this option if the device receiving the memo, such as a cell phone, requires a specific character set encoding and does not support UTF-8. To specify this option, use the [bibus » deliveryOptionString](#) class.

memoPart

Specifies the content of the memo. To specify this option, use the [bibus » deliveryOptionMemoPart](#) class.

notificationList

Specifies the [bibus » baseAgentDefinition](#) or [bibus » baseReport](#) object that contains the alert list. Users on the alert list are treated as bcc recipients of the memo.

To specify this option, use the [bibus » deliveryOptionSearchPathSingleObject](#) class.

New in Version 8.3 — “Report Email Alerts” on page 1900

This enumeration value can now be used with [bibus » baseReport](#) objects.

subject

Specifies the memo subject line. To specify this option, use the [bibus » deliveryOptionString](#) class.

to

Specifies the recipients on the "to" list. Recipients must be instances of the [bibus » account](#) class, the [bibus » contact](#) class, the [bibus » distributionList](#) class, the [bibus » group](#) class, or the [bibus » role](#) class.

To specify this option, use the [bibus » deliveryOptionSearchPathMultipleObjectArray](#) class.

toAddress

Specifies the address of each recipient on the "to" list. To specify this option, use the [bibus » deliveryOptionAddressSMTPArray](#) class.

deploymentOptionEnum

Defines the deployment options.

References

Used by the following properties:

- [bibus » deploymentOption » name](#)

Members

archive

Specifies the archive used to perform the deployment.

To specify this option, use the [bibus » deploymentOptionString](#) class.

archiveEncryptPassword

Specifies the password used to encrypt and decrypt the deployment archive. A password is required if [dataSourceSignonSelect](#) value or [entireContentStoreSelect](#) value are `true`. If a password is not specified during the export, the archive is not encrypted. An encrypted archive can be read only when the correct password is provided as input to the [deployment » getDeploymentOptions\(archive, options\)](#) method or the [asynch » run\(objectPath, parameterValues, options\)](#) method.

This password must be in credential XML format. For information about this format, see the [bibus » credential](#) class.

To specify this option, use the [bibus » deploymentOptionAnyType](#) class.

archiveOverwrite

Specifies whether the deployment overwrites the archive.

To specify this option, use the [bibus](#) » [deploymentOptionBoolean](#) class.

Default: `false`

dataSourceConflictResolution

Specifies whether the original data source is kept or is overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus](#) » [deploymentOptionResolution](#) class.

Default: `replace`

dataSourceSelect

Specifies whether the data sources are processed during the deployment.

To specify this option, use the [bibus](#) » [deploymentOptionBoolean](#) class.

Default: `false`

dataSourceSignonSelect

Specifies whether the data source signons are processed during the deployment.

To specify this option, use the [bibus](#) » [deploymentOptionBoolean](#) class.

Default: `true`

deploymentDescription

Specifies the description of the deployment.

To specify this option, use the [bibus](#) » [deploymentOptionMultilingualString](#) class.

deploymentName

Specifies the name of the deployment.

To specify this option, use the [bibus](#) » [deploymentOptionMultilingualString](#) class.

deploymentScreenTip

Specifies the screen tip of the deployment.

To specify this option, use the [bibus](#) » [deploymentOptionMultilingualString](#) class.

entireContentStoreReplace

Specifies whether the entire content store is replaced.

To specify this option, use the [bibus](#) » [deploymentOptionBoolean](#) class.

Default: `true`

New in Version 10.2.0 – “Support for multi-tenancy” on page 1838

The default value changed from `false` to `true`. SDK applications designed to support multi-tenancy must specify a value. When `tenants` is specified, this value must be set to `false`; otherwise, an error is returned.

entireContentStoreSelect

Specifies whether the entire content store, except for personal data, is deployed. If the value is `true`, the rest of the deployment options, except for `archive` value, `archiveEncryptPassword` value, `archiveOverwrite` value, and `personalDataSelect` value, are ignored.

To specify this option, use the [bibus](#) » [deploymentOptionBoolean](#) class.

Default: `false`

export

Specifies the objects to be exported from the deployment source.

To specify this option, use the [bibus » deploymentOptionSearchPathSingleObjectArray](#) class.

There is no default.

New in Version 8.3 – “Package Hierarchies” on page 1913

This value was added.

import

Specifies how to import objects from the deployment archive.

To specify this option, use the [bibus » deploymentOptionImportRuleArray](#) class.

There is no default.

New in Version 8.3 – “Package Hierarchies” on page 1913

This value was added. It replaces the [bibus » deploymentOptionEnum](#) » package value.

information

Returns information about the root objects exported from the deployment source.

Never specify this option. They are returned by the [deployment » getDeploymentOptions\(archive, options\)](#) method. This option is used with the [bibus » deploymentOptionObjectInformationArray](#) class.

New in Version 8.3 – “Package Hierarchies” on page 1913

This value was added. It replaces the [bibus » deploymentOptionEnum](#) » package value.

namespaceConflictResolution

Specifies whether the Cognos groups and roles are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

namespaceSelect

Specifies whether the Cognos groups and roles are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

namespaceThirdParty

Specifies whether references to security objects are processed during deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `true`

objectPolicies

Specifies whether the identified object and user capability policies are imported during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

New in Version 10.1.0 – “Package Capabilities Overwritten on Default Import/Export” on page 1867

This value was updated to also import user capability policies during deployment.

objectPoliciesConflictResolution

Specifies whether the original object policies are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

ownershipConflictResolution

Specifies whether the original object owners are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

packageDatasetOutputs

Specifies whether data set outputs for deployed packages are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: `false`

packageDatasetOutputsConflictResolution

Specifies whether the original data set outputs and versions are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

packageDataSourceConflictResolution

Reserved.

New in Version 8.4 – “Package Data Sources” on page 1895

This value was added.

packageDataSourceSelect

Reserved.

New in Version 8.4 – “Package Data Sources” on page 1895

This value was added.

packageDataSourceSignonSelect

Reserved.

New in Version 8.4 – “Package Data Sources” on page 1895

This value was added.

packageHistories

Specifies whether the run histories for all deployed packages are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

packageHistoriesConflictResolution

Specifies whether the run histories are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

packageOutputs

Specifies whether report outputs for deployed packages are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

packageOutputsConflictResolution

Specifies whether the original report outputs and versions are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

packageSchedules

Specifies whether the schedules for all deployed packages are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

packageSchedulesConflictResolution

Specifies whether the original schedules are kept or are overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: [replace](#)

packageSelect

Specifies whether all deployed packages are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `true`

personalDataSelect

Specifies whether user account information is deployed. Use this option only when the entire content store is deployed.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

preserveStoreIDs

Specifies whether the [bibus » baseClass » storeID](#) values for objects should be preserved during a deployment import.

Two objects cannot have the same [bibus » baseClass » storeID](#) value in the same content store.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

recipientsConflictResolution

Specifies whether the recipients are kept or overwritten by the deployment when a conflict occurs. This option can be set during the export and import but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: `replace`

recipientsSelect

Specifies whether distribution lists and contacts are processed during the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

recordingLevel

Specifies the amount of detail recorded in the history of the deployment.

To specify this option, use the [bibus » deploymentOptionAuditLevel](#) class. Valid options are [minimal](#), [basic](#), and [trace](#).

Default: `basic`

New in Version 8.3 – “Deployment History” on page 1915

This value was added.

takeOwnership

Specifies whether the `owner` property for the object should be changed to reference the [bibus » account](#) used to [import](#) the deployment.

If the Content Manager advanced setting `CM.DeploymentUpdateScheduleCredential` is set to `true`, the `credential` property of all imported [bibus » schedule](#) objects will be changed to reference the [bibus » credential](#) contained in the [bibus » account](#) used to [import](#) the deployment.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: `false`

New in Version 8.4 – “Updating the Credential of Imported Schedules” on page 1895

The Content Manager advanced setting `CM.DeploymentUpdateScheduleCredential` was added so that when a deployment archive is imported into the content store, the `credential` property of all imported [bibus » schedule](#) objects can be changed to reference the [bibus » credential](#) contained in the [bibus » account](#) used to [import](#) the deployment.

tenants

Specifies the list of tenants whose entire content will be exported from the source content store and imported into the target content store.

This value can only be set by system administrators who can access all tenants and their content. For non-system administrators, if this value is set, Content Manager will generate a fault.

If the `tenants` value is set, [entireContentStoreSelect](#) must be set to `true`, and [entireContentStoreReplace](#) must be set to `false`.

To specify this option, use the [bibus » deploymentOptionStringArray](#) class.

Default: An empty collection.

New in Version 10.2.0 – “Support for multi-tenancy” on page 1838

This value was added.

tenantsContentConflictResolution

Specifies whether the tenant content in the target repository is kept or replaced by the tenant deployment when a conflict occurs.

If the value is set to keep (default), the target object's properties are not updated with the values from the archive.

If the value is set to replace, the target object's properties are updated with the values from the archive.

This value can be set during export and import, but is only applied during the import.

To specify this option, use the [bibus » deploymentOptionResolution](#) class.

Default: keep

upgradeArchiveOptions

Specifies whether deployment options stored in an archive should be upgraded before being returned by the [deployment » getDeploymentOptions\(archive, options\)](#) method.

To specify this option, use the [bibus » deploymentOptionBoolean](#) class.

Default: false

New in Version 8.3 — “Package Hierarchies” on page 1913

This value was added.

upgradeClasses

Specifies the classes of objects in the deployment that will be upgraded when they are imported.

To specify this option, use the [bibus » deploymentOptionClassEnumArray](#) class.

Default: An empty collection.

deploymentStatusEnum

Defines the deployment status of an object.

References

Used by the following properties:

- [bibus » deploymentDetail » deployedObjectStatus](#)
- [bibus » deploymentStatusEnumProp » value](#)

Members

added

The object was added to the content store during the deployment.

existing

The object was not deployed. More information about the deployment of this object is provided by an instance of [bibus » deploymentDetail](#).

exported

The object was exported to the archive during the deployment.

failed

The deployment on this object failed.

kept

The object was kept. Information in the deployment archive related to the deployed object was not used.

replaced

The object was replaced during the deployment.

updated

The object was updated during the deployment.

drillOptionEnum

Defines the drill options.

References**Used by the following properties:**

- [bibus](#) » [drillOption](#) » [name](#)

Members**down**

Specifies the context for a drill-down operation.

To specify this option, use the [bibus](#) » [drillOptionParameterValues](#) class.

up

Specifies the context for a drill-up operation.

To specify this option, use the [bibus](#) » [drillOptionParameterValues](#) class.

drillThroughActionEnum

Provides the set of actions that can be used in a drill-through request.

References**Used by the following properties:**

- [bibus](#) » [asynchDetailDrillThroughRequest](#) » [action](#)
- [bibus](#) » [asynchDetailDrillThroughTarget](#) » [action](#)

What's new**New in Version 8.4 — “[Dynamic Filtering of Report Data](#)” on page 1889**

This enumeration set was added.

Members**<http://developer.cognos.com/ceba/constants/drillThroughActionEnum#edit>**

Specifies that the target object should be edited using the drill-through request recipient.

<http://developer.cognos.com/ceba/constants/drillThroughActionEnum#get>

Specifies that the target object should be viewed using an HTTP GET.

<http://developer.cognos.com/ceba/constants/drillThroughActionEnum#post>

Specifies that the target object should be viewed using an HTTP POST.

<http://developer.cognos.com/ceba/constants/drillThroughActionEnum#run>

Specifies that the target object should be run using the drill-through request recipient.

<http://developer.cognos.com/ceba/constants/drillThroughActionEnum#viewOutput>

Specifies that the target object output should be viewed using the drill-through request recipient.

If suitable output is not available, the `run` action is used to generate appropriate output for the user. The “[defaultOutput\(searchPath\)](#)” on page 1571 contains additional information about how IBM Cognos determines the most suitable output for a user.

drillThroughOptionEnum

Defines the drill-through options.

References

Used by the following properties:

- [bibus](#) » [drillThroughOption](#) » [name](#)

Members

contextFormat

Specifies the requested selection context format for the [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#) method.

To specify this option, use the [bibus](#) » [drillThroughOptionAnyURI](#) class.

The supported values are defined by the [bibus](#) » [selectionContextFormatEnum](#) enumeration set.

Default: `identity`

New in Version 8.4 – “Drill-Through Improvements” on page 1889

This value was added.

diagnostics

Reserved.

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This value was added.

includeDataSourceActions

Specifies that actions (in the form of a URL) from the underlying data source should be returned.

To specify this option, use the [bibus](#) » [drillThroughOptionBoolean](#) class.

Default: `true`

includeDrillThroughPaths

Specifies that relevant drill-through paths from the content store should be returned.

To specify this option, use the [bibus](#) » [drillThroughOptionBoolean](#) class.

Default: `true`

selectionContext

Specifies the selection context from the drill-through source.

The selection context must be specified using the `cognos8` format. You can use the `drillThrough » convertDrillThroughContext(inputContext, parameterValues, options)` method to convert a selection context from another format to `cognos8` format.

To specify this option, use the `bibus » drillThroughOptionXMLEncodedXML` class.

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This value was added.

useEditSpecification

Specifies whether an edit specification should be constructed as part of the drill-through request. The edit specification can provide additional filter criteria that cannot be passed to the target using its defined parameters. For example, the target may not have sufficient parameters defined, or the selection may be too complex to specify using parameters.

The target may or may not apply the criteria in the edit specification.

To specify this option, use the `bibus » drillThroughOptionBoolean` class.

Default: `false`

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This value was added.

drillThroughRecipientEnum

Provides the set of recipients that can be used in a drill-through request.

References

Used by the following properties:

- `bibus » asyncDetailDrillThroughRequest » recipient`
- `bibus » asyncDetailDrillThroughTarget » recipients`

What's new

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/drillThroughRecipientEnum#analysisStudio>

Specifies Analysis Studio.

<http://developer.cognos.com/ceba/constants/drillThroughRecipientEnum#cognosViewer>

Specifies Cognos Viewer.

<http://developer.cognos.com/ceba/constants/drillThroughRecipientEnum#powerPlayStudio>

Specifies PowerPlay Studio.

<http://developer.cognos.com/ceba/constants/drillThroughRecipientEnum#queryStudio>

Specifies Query Studio.

<http://developer.cognos.com/ceba/constants/drillThroughRecipientEnum#reportStudio>

Specifies Reporting.

encodingEnum

Specifies how certain properties are placed in the message.

Some properties have large values. When these values are passed as part of the Simple Object Access Protocol (SOAP) message, the demand for memory on your system may be too high. If this happens, the request will fail. By placing these values in an attachment, the amount of memory required for the request may be decreased.

References

Used by the following properties:

- [bibus](#) » [addOptions](#) » [dataEncoding](#)
- [bibus](#) » [queryOptions](#) » [dataEncoding](#)
- [bibus](#) » [asynchOptionEncoding](#) » [value](#)

Members

base64

Specifies that binary properties will be base64 encoded. The value will be located in the SOAP body.

MIME

Specifies that data may be placed in a MIME attachment, depending on the request. This method of transmitting the data must be supported by the server.

MIMECompressed

Specifies that data may be placed in a MIME attachment, depending on the request. The data is compressed using the gzip compression algorithm. This method of transmitting the data must be supported by the server.

eventTypeEnum

Defines the types of events created by an agent condition query.

References

Used by the following properties:

- [bibus](#) » [agentTaskDefinition](#) » [eventTypes](#)
- [bibus](#) » [eventTypeEnumArrayProp](#) » [value](#)
- [bibus](#) » [eventTypeEnumProp](#) » [value](#)

Members

changed

Specifies that this is a changed event. A changed event existed the last time the agent was executed, but its value is different.

deleted

Specifies that this is a deleted event. A deleted event existed the last time the agent was executed, but no longer exists.

new

Specifies that this is a new event. A new event did not exist the last time the agent was executed.

nil

Specifies that there are no events.

unchanged

Specifies that this is an unchanged event. An unchanged event existed the last time the agent was executed, and has not changed since the last execution of the agent.

formatEnum

Specifies whether the information is encrypted.

References

Used by the following properties:

- [bibus](#) » [formFieldVar](#) » [format](#)

Members

encrypted

Specifies that the information is encrypted.

not_encrypted

Specifies that the information is not encrypted.

gcPolicyEnum

Defines the set of JVM garbage collection policies.

References

Used by the following properties:

- [bibus](#) » [configuration](#) » [qsGCPolicy](#)
- [bibus](#) » [configurationFolder](#) » [qsGCPolicy](#)
- [bibus](#) » [dispatcher](#) » [qsGCPolicy](#)
- [bibus](#) » [queryService](#) » [qsGCPolicy](#)

What's new

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/gcPolicyEnum#Balanced>

Uses the `-Xgcpolicy:balanced` garbage collection option supported in a 64-bit IBM JVM.

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This value was added.

<http://developer.cognos.com/ceba/constants/gcPolicyEnum#Custom>

Customized garbage collection tuning is specified using [bibus](#) » [configuration](#) » [qsAdditionalJVMArguments](#), [bibus](#) » [configurationFolder](#) » [qsAdditionalJVMArguments](#), [bibus](#) » [dispatcher](#) » [qsAdditionalJVMArguments](#), and [bibus](#) » [queryService](#) » [qsAdditionalJVMArguments](#).

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This value was added.

<http://developer.cognos.com/ceba/constants/gcPolicyEnum#Generational>

Uses the `-Xgcpolicy:gencon` garbage collection option supported in IBM JVM.

When an alternate JVM is used, the following garbage collection options are implemented: `-XX:+UseConcMarkSweepGC` `-XX:+UseParNewGC` `-XX:SurvivorRatio=8` `-XX:TargetSurvivorRatio=90` `-XX:MaxTenuringThreshold=31`.

New in Version 10.2.0 – “New queryService configuration options” on page 1837

This value was added.

governorEnum

Defines the set of governors used by IBM Cognos Analytics.

References

Used by the following properties:

- [bibus](#) » [governor](#) » [name](#)

Members

crossProductAllowed

Specifies whether the query can contain cross-product joins.

To specify this option, use a governor of class [bibus](#) » [governorBoolean](#).

Default: `false`

directEntrySQL

Specifies whether the query can contain direct-entry SQL.

To specify this option, use a governor of class [bibus](#) » [governorBoolean](#).

Default: `false`

maxQueryExecutionTime

Specifies the maximum amount of time a query can run.

To specify this option, use a governor of class [bibus](#) » [governorInt](#).

maxRowsRetrieved

Specifies the maximum number of rows a query can retrieve.

To specify this option, use a governor of class [bibus](#) » [governorInt](#).

maxTablesPerQuery

Specifies the maximum number of tables a query can access.

To specify this option, use a governor of class [bibus](#) » [governorInt](#).

maxTextBlobCharacters

Specifies the maximum number of characters a query can retrieve from text blobs.

To specify this option, use a governor of class [bibus](#) » [governorInt](#).

outerJoinAllowed

Specifies whether the query can use outer joins.

To specify this option, use a governor of class [bibus » governorBoolean](#).

Default: `true`

sortOnNonIndexedColumn

Specifies whether a sort based on a non-indexed column can occur in the query.

To specify this option, use a governor of class [bibus » governorBoolean](#).

Default: `false`

useSQLWithClause

Specifies whether the SQL WITH clause can be used when generating database queries.

When this value is `true`, IBM Cognos Analytics may construct an SQL WITH clause to satisfy the user's request. When this value is `false`, IBM Cognos Analytics constructs SQL queries using derived tables, if required. For simple queries, neither derived tables nor the WITH clause are used.

Use of the WITH clause is dependent on the following:

- The WITH clause is supported by the underlying database.
- The user query is not recursive.
- This governor is set to `true`.

This value is `false` by default. To enable the use of the WITH clause, use a governor of class [bibus » governorBoolean](#).

indexOptionEnum

Provides the list of index options.

References

Used by the following properties:

- [bibus » indexOption » name](#)

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Members

excludedObjects

Specifies the list of objects that the index update task excludes. The descendants of the objects are also excluded. An object is included if it, or any of its ancestors, are specified by the [includedObjects](#) option.

If an object was previously indexed and it is then excluded using this option, the object is removed from the index when the index update task is run.

To specify this option, use the [bibus » indexOptionSearchPathMultipleObjectArray](#) class.

Default: No objects are excluded from the index when the index update task is run.

New in Version 8.4 — “[Index Options](#)” on page 1893

This value was added.

New in Version 10.1.0 – “Previously Reserved Index Options Now Available for Client Use” on page 1877

This item, previously marked as "Reserved", is now available for client use.

force

Specifies that the operation must proceed. This ignores any caching or optimizations, for example, based on the modification date of the document to be indexed or other factors. To specify this option, use the [bibus » indexOptionBoolean](#) class.

Default: false

New in Version 10.1.0 – “Previously Reserved Index Options Now Available for Client Use” on page 1877

This item, previously marked as "Reserved", is now available for client use.

includedObjects

Specifies the list of objects included by the index update task. The descendants of the specified objects are indexed unless they are specified by the [excludedObjects](#) option. To specify this option, use the [bibus » indexOptionSearchPathMultipleObjectArray](#) class.

Default: All objects are included in the index by the index update task.

New in Version 8.4 – “Index Options” on page 1893

This value was added.

New in Version 10.1.0 – “Previously Reserved Index Options Now Available for Client Use” on page 1877

This item, previously marked as "Reserved", is now available for client use.

indexModeledData

Specifies that all data in the models in scope of the indexing task are indexed. To specify this option, use the [bibus » indexOptionBoolean](#) class.

Default: false

New in Version 10.1.0 – “Data Collection Options for Indexing Tasks” on page 1870

This value was added.

indexObjectTypes

Specifies that properties, metadata, and related objects (for example, output) for objects that are in scope of the indexing task and have been selected as indexable types are indexed. To specify this option, use the [bibus » indexOptionBoolean](#) class.

Default: false

New in Version 10.1.0 – “Data Collection Options for Indexing Tasks” on page 1870

This value was added.

indexReferencedData

Specifies that only data referenced by expressions in objects in scope of the indexing task are indexed. Model objects in scope are ignored. To specify this option, use the [bibus » indexOptionBoolean](#) class.

Default: false

New in Version 10.1.0 – “Data Collection Options for Indexing Tasks” on page 1870

This value was added.

maximumDocuments

Reserved.

returnLocales

Reserved.

returnSourceDocument

Reserved.

returnTerms

Reserved.

indexTermOptionEnum

Reserved.

References

Used by the following properties:

- [bibus](#) » [indexTermOption](#) » [name](#)

What's new

New in Version 8.3 — “[Search — For Internal Use Only](#)” on page 1920

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

Members

childTerms

Reserved.

exampleTerms

Reserved.

parentTerms

Reserved.

returnChildTerms

Reserved.

returnExampleTerms

Reserved.

returnParentTerms

Reserved.

returnSiblingTerms

Reserved.

siblingTerms

Reserved.

installedComponentEnum

Provides the list of IBM Cognos Analytics install components.

References

Used by the following properties:

- [bibus » installedComponent » componentID](#)
- [bibus » installedComponentEnumProp » value](#)

Members

adaptiveAnalytics

Specifies Analyst Add-in for IBM Cognos Analytic Applications.

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

This value was added.

adaptiveAnalyticsService

Specifies the service that supports Analyst Add-in for IBM Cognos Analytic Applications.

New in Version 10.1.0 – [“Adaptive Analytics Service” on page 1856](#)

This value was added.

agentService

Specifies the agent service.

analysisStudio

Specifies IBM Cognos Analysis Studio. .

annotationService

New in Version 10.1.0 – [“Annotation Service” on page 1864](#)

This value was added.

batchReportService

Specifies the batch report service.

contentManagerCacheService

Specifies the Content Manager cache service.

New in Version 10.1.0 – [“Content Manager Cache Service” on page 1862](#)

This value was added.

contentManagerService

Specifies the Content Manager service.

controllerStudio

Specifies Controller Studio.

dashboardViewer

Specifies IBM Cognos Workspace.

New in Version 8.4 – [“Dashboards” on page 1894](#)

This value was added.

dataAdvisorService

Specifies the service that supports IBM Cognos Express Data Advisor.

New in Version 10.1.0 – [“Updated Support for IBM Cognos Express” on page 1871](#)

This value was added.

dataIntegrationService

Specifies the data integration service.

dataMovementService**deliveryService**

Specifies the delivery service.

dimensionManagementService

Specifies the dimension management service.

New in Version 8.4 – “Dimension Management Service” on page 1896

This value was added.

eventManagementService

Specifies the event management service.

eventStudio

Specifies Event Studio.

EVService

Specifies the service that supports IBM Cognos Express Advisor.

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This value was added.

EVStudio

Specifies IBM Cognos Express Advisor.

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This value was added.

graphicsService

Specifies the graphics service.

New in Version 10.1.0 – “Graphics Service” on page 1860

This value was added.

humanTaskService

Specifies the human task service.

New in Version 10.1.0 – “Human Task Service” on page 1854

This value was added.

idVizService

Reserved.

New in Version 10.2.0 – “Support for interactive discovery and visualization” on page 1839

This value was added.

indexDataService

Specifies the index data service.

indexSearchService

Specifies the index search service.

indexUpdateService

Specifies the index update service.

jobService

Specifies the job service.

logService

Specifies the log service.

metadataService**metricsManagerService**

Specifies the metrics manager service.

metricStudio

Specifies IBM Cognos Metric Studio.

migrationService

Specifies the migration service.

New in Version 8.4 – “Migration Service” on page 1883

This value was added.

mobileService

Specifies IBM Cognos Analytics Mobile Reports.

monitorService

Specifies the monitor service.

myDataSets**New in Version 10.2.2 – My data sets**

This property was added.

planningAdministrationConsoleService

Specifies the planning administration console service.

planningContributor

Specifies IBM Cognos Planning - Contributor.

planningDataService

Specifies IBM Cognos Planning.

planningRuntimeService

Specifies the planning runtime service.

planningTaskService

Specifies the planning task service.

powerPlayService

Specifies the PowerPlay service.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

powerPlayStudio

Specifies IBM Cognos PowerPlay Studio.

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

presentationService

Specifies the presentation service.

queryService

Specifies the query service.

New in Version 10.1.0 – “Query Service” on page 1855

This value was added.

queryStudio

Specifies IBM Cognos Query Studio.

relationalMetadataService

Specifies the service responsible for querying and updating unpublished IBM Cognos Framework Manager models.

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This value was added.

reportDataService

Specifies IBM Cognos Mashup Service.

reportService

Specifies the report service.

reportStudio

Specifies IBM Cognos Analytics - Reporting.

repositoryIntegration

Specifies IBM Cognos Content Archival.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

repositoryService

Reserved.

New in Version 10.2.0 – “Repository service” on page 1840

This value was added.

saCAMService**New in Version 10.2.0 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1838**

This value was added.

New in Version 10.2.1 – “New standalone IBM Cognos Access Manager (CAM) service” on page 1832

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

systemService

Specifies the system service.

tm1Studio

Specifies IBM Cognos TM1 Studio

New in Version 10.1.0 – “IBM Cognos TM1 Packages” on page 1858

This value was added.

isolationLevelEnum

Specifies the database isolation levels.

In increasing order of isolation, the isolation levels are: readUncommitted, readCommitted, cursorStability, reproducibleRead, phantomProtection, and serializable.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [dataSourceConnection](#) » [isolationLevel](#)

Members

cursorStability

Specifies that other transactions cannot update the row on which the cursor is positioned.

phantomProtection

Specifies that access by a transaction is prevented for rows inserted or deleted after the start of the transaction.

readCommitted

Specifies that transaction access is given only to those rows that are committed by other transactions.

readUncommitted

Specifies that changes made by other transactions are immediately available to a transaction.

reproducibleRead

Specifies that rows selected or updated by a transaction will not be changed by another transaction until the transaction is complete.

serializable

Specifies that a set of transactions executed concurrently produce the same result as if they were performed sequentially.

jobOptionEnum – obsolete

Provides the list of job options.

Members

continueOnError – obsolete

Specifies whether a job should continue to execute steps if a preceding step has failed. To specify this option, use the [bibus](#) » [jobOptionBoolean](#) class.

When running jobs in [sequential](#) fashion, the default behavior upon the failure of a job step is to:

- discontinue processing

- set the failed step object's [status](#) to [failed](#)
- set the [status](#) of all subsequent steps to [purged](#)

Setting this value to `true` allows subsequent job steps to continue, but the overall [status](#) of the [bibus » jobDefinition](#) is still set to [failed](#).

When running jobs in [parallel](#), this setting has no effect.

Default: `false`

This value is deprecated. Use the [bibus » asynchOptionEnum » continueOnError](#) value instead.

linearUnitEnum

Defines the linear units of measurement supported by IBM Cognos Analytics.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus » pageDefinition » unit](#)

Members

cm

Specifies page dimensions in centimetres (cm).

in

Specifies page dimensions in inches (in).

loadBalancingModeEnum

Defines the load balancing mode used by dispatchers.

Load-balancing settings must be used in conjunction with the [bibus » configuration » serverGroup](#) property, [bibus » configurationFolder » serverGroup](#) property, and [bibus » dispatcher » serverGroup](#) property to properly define dispatcher groups.

References

Used by the following properties:

- [bibus » configuration » loadBalancingMode](#)
- [bibus » configurationFolder » loadBalancingMode](#)
- [bibus » dispatcher » loadBalancingMode](#)
- [bibus » loadBalancingModeEnumProp » value](#)

Members

clusterCompatible

Specifies that low affinity requests are processed locally whenever possible. The local dispatcher may forward the request to a remote dispatcher if the service necessary to run the request is not available locally. Requests that can not be processed locally are load balanced to other dispatchers in the same application cluster using a weighted round-robin algorithm.

High affinity requests are forwarded to the associated dispatcher whenever possible. This type of load balancing takes place regardless of the load balancing mode specified for the local dispatcher.

weightedRoundRobin

Specifies that the dispatcher load balances requests using a weighted round-robin algorithm.

mapLayerTypeEnum

Defines the map layer types.

Members

other

point

region

mapOptionEnum

Provides the list of map options.

References

Used by the following properties:

- [bibus](#) » [mapOption](#) » [name](#)

Members

featureID

Identifies the map feature to be returned.

The value of [bibus](#) » [mapFeatureInfo](#) » [id](#) in the returned [bibus](#) » [mapFeatureInfo](#) object will match the value of the [bibus](#) » [mapOptionString](#) » [value](#) property.

To specify this option, use the [bibus](#) » [mapOptionString](#) class.

layerID

Identifies the layer of the map to be searched.

To specify this option, use the [bibus](#) » [mapOptionString](#) class.

mapID

Identifies the map to be searched.

To specify this option, use the [bibus](#) » [mapOptionString](#) class.

maximumFeatureCount

Specifies the maximum number of features in the result set.

To specify this option, use the [bibus](#) » [mapOptionInt](#) class.

Default: 0 (will return all values)

parentFeatureID

Identifies the map feature to be searched.

The value of [bibus](#) » [mapFeatureInfo](#) » [parentID](#) in the returned [bibus](#) » [mapFeatureInfo](#) object will match the value of the [bibus](#) » [mapOptionString](#) » [value](#) property.

To specify this option, use the [bibus](#) » [mapOptionString](#) class.

searchLocale

Specifies the locale to be used to perform the search.

To specify this option, use the [bibus](#) » [mapOptionLanguage](#) class.

Default: the user's preferred content locale, as defined by the [bibus](#) » [account](#) » [contentLocale](#) property.

searchType

Specifies the type of search to be performed.

To specify this option, use the [bibus](#) » [mapOptionMapSearchType](#) class.

searchValue

Specifies the value to be used to perform the search.

To specify this option, use the [bibus](#) » [mapOptionString](#) class.

migrationTaskOptionEnum

Provides the list of migration task options.

This enumeration set is not used.

References

Used by the following properties:

- [bibus](#) » [migrationTaskOption](#) » [name](#)

What's new

New in Version 8.4 — “Migration Service” on page 1883

This enumeration set was added.

Members

conflictResolution

Specifies whether the original content is kept or is overwritten by the [bibus](#) » [migrationTask](#) when a conflict occurs.

To specify this option, use the [bibus](#) » [migrationTaskOptionResolution](#) class.

Default: [replace](#)

cubeMappings

Specifies the location of the object in the content store that contains the cube mappings. To specify this option, use the [bibus](#) » [migrationTaskOptionSearchPathSingleObject](#) class.

mappings

Specifies the target class for each source object class. To specify this option, use the [bibus](#) » [migrationTaskOptionMappingArray](#) class.

The following mappings are supported:

Table 226. Target class list for source object classes

From one of	To one of
bibus » powerPlayReport	bibus » analysis bibus » powerPlay8Report bibus » report
bibus » powerPlay8Report	bibus » analysis bibus » report

sourceObjects

Specifies the source objects to be migrated. To specify this option, use the [bibus » migrationTaskOptionSearchPathSingleObjectArray](#) class.

targetPath

Specifies the location to be used when creating migrated content. To specify this option, use the [bibus » migrationTaskOptionSearchPathSingleObject](#) class.

mobileOptionEnum

Provides the list of mobile options.

References

Used by the following properties:

- [bibus » mobileOption » name](#)

What's new

New in Version 8.4 — “Mobile Service” on page 1891

This enumeration set was added.

Members

to

Specifies the recipients on the "to" list. Recipients must be instances of the [bibus » account](#) class, the [bibus » distributionList](#) class, the [bibus » group](#) class, or the [bibus » role](#) class.

To specify this option, use the [bibus » mobileOptionSearchPathMultipleObjectArray](#) class.

monitorOptionEnum

Provides the list of options for the monitoring service.

References

Used by the following properties:

- [bibus » monitorOption » name](#)

Members

background

Specifies whether the task should be run in the background. To specify this option, use the [bibus » monitorOptionBoolean](#) class.

Default: `false`

writeCompleteHistory

Specifies whether the complete task history should be written to the content store.

When this option is `true` or is omitted, the complete task history is written to the content store.

When this option is set to `false`, and the task and child tasks complete successfully, the task history for the root task is written to the content store and task history for child tasks is *not* written to the content store. Throughput of background tasks is thus improved by reducing the number of history objects that need to be written to the content store when the task completes successfully. Features of the Past Activities portal pages that rely on the presence of task history logs are not available for those tasks affected by this option.

If a failure occurs during the execution of the task (including child tasks) the complete task history is written to the content store, regardless of the value specified for this option.

If the task causes a child task to run, then the option value for the task is included in the run request for the child task. Therefore, even if the child task specifies this option, the value specified in the child task is overridden by the parent task option.

To specify this option, use the [bibus » monitorOptionBoolean](#) class.

Default: `true`

New in Version 8.4 — “Writing Task History Subsets” on page 1890

This value was added.

monthsEnum

Specifies the months of the year.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus » schedule » yearlyAbsoluteMonth](#)
- [bibus » schedule » yearlyRelativeMonth](#)

Members

april

Specifies the month of April.

august

Specifies the month of August.

december

Specifies the month of December.

february

Specifies the month of February.

january

Specifies the month of January.

july

Specifies the month of July.

june

Specifies the month of June.

march

Specifies the month of March.

may

Specifies the month of May.

november

Specifies the month of November.

october

Specifies the month of October.

september

Specifies the month of September.

namespaceCapabilityEnum

Lists the set of capabilities that are supported by this namespace.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [namespace](#) » [capabilities](#)

Members

caseSensitive

Specifies that search operations are case-sensitive.

contains

Specifies that the function `contains` can be used in a search path.

For more information about the `contains` function, see [“contains\(propertyname,string\)” on page 1574.](#)

equals

Specifies that the equal sign (=) can be used in a search path. Use for searches in which an exact match is required.

sort

Specifies that sorting can be performed in a search operation.

startsWith

Specifies that the function `starts-with` can be used in a search path.

For more information about the `starts-with` function, see [“starts-with\(propertyname,string\)” on page 1575.](#)

orderEnum

Specifies the sort options.

References

Used by the following properties:

- [bibus](#) » [sort](#) » [order](#)

Members

ascending

Specifies ascending order.

descending

Specifies descending order.

outputEncapsulationEnum

Defines how report outputs can be encapsulated in response to the [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method and the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method.

References

Used by the following properties:

- [bibus](#) » [runOptionOutputEncapsulation](#) » [value](#)

Members

HTML

Specifies that an HTML document should be returned.

If the requested output format is not HTML, the output will be cached on the server and an HTML document referencing the cached output will be returned.

If the requested output format is HTML, the output will be returned without modification.

none

Specifies that the output is not encapsulated.

SDK developers usually use this value when running reports.

URL

Specifies that a URL should be returned.

When this value is specified, the output will be cached on the server and a URL referencing the cached output will be returned.

URLQueryString

Specifies that the query portion of the URL of the document should be returned. This is the portion of the URL that begins with a question mark (?).

When this value is specified, the output will be cached on the server and a URL query string that links to the cached output will be returned.

outputFormatEnum

Defines the set of output format types.

For information about report output formats, including limitations and security impacts, see the *Administration and Security Guide*.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [baseReport](#) » [defaultOutputFormat](#)
- [bibus](#) » [account](#) » [format](#)
- [bibus](#) » [contact](#) » [format](#)
- [bibus](#) » [documentContent](#) » [format](#)
- [bibus](#) » [memoPartAgentObject](#) » [format](#)
- [bibus](#) » [output](#) » [format](#)

Members

CSV

Specifies the Comma Separated Values (CSV) text format. When this format is returned inline, it is base64 encoded.

HTML

Specifies the Hypertext Markup Language (HTML) format.

HTMLFragment

Specifies an HTML fragment that does not contain HTML, head, or body tags. Use to include an IBM Cognos Analytics report within another HTML document.

layoutDataXML

Reserved.

Note: The layoutDataXML format specified here differs from the layoutDataXML format used in the IBM Cognos Mashup Service and is subject to change without notice. Customers who wish to retrieve report outputs in layoutDataXML format should use the methods documented in the *Mashup Service Developer Guide*.

New in Version 8.3 — “Advanced Features for Report Output” on page 1912

This value was added.

MHT

Specifies the HTML Web Archive (MHT) format.

pagedLayoutDataXML

Reserved.

Note: The paged layoutDataXML format specified here differs from the paged layoutDataXML format used in the IBM Cognos Mashup Service and is subject to change without notice. Customers who wish to retrieve paged report outputs in layoutDataXML format should use the methods documented in the *Mashup Service Developer Guide*.

New in Version 8.4 – “pagedLayoutDataXML Report Output Format” on page 1889

This value was added.

PDF

Specifies the Acrobat Portable Document Format (PDF). When this format is returned inline, it is base64 encoded.

singleXLS – deprecated

Specifies the single sheet or [®] Excel Spreadsheet (XLS) format. Note that at most 65,536 rows of data will be placed in the output when using this format.

This value

- is deprecated and will be removed in a future version of the product

New in Version 10.1.0 – “Production of Excel 2000 Format Report Outputs is Obsolete” on page 1866

This value is deprecated. Use the following values instead:

- [bibus](#) » [outputFormatEnum](#) » [spreadsheetML](#) value
- [bibus](#) » [outputFormatEnum](#) » [XLWA](#) value

spreadsheetML

Specifies the SpreadsheetML (Microsoft [®] Excel Office Open XML) format. When this format is returned inline, it is base64 encoded.

This value is represented as **Excel 2007** in the portal.

New in Version 8.3 – “spreadsheetML Output Format” on page 1910

This value was added.

XHTML

Specifies the Extensible Hypertext Markup Language (XHTML) format.

XLS – deprecated

Specifies the Microsoft Excel Spreadsheet (XLS) format.

This value

- is deprecated and will be removed in a future version of the product

New in Version 10.1.0 – “Production of Excel 2000 Format Report Outputs is Obsolete” on page 1866

This value is deprecated. Use the following values instead:

- [bibus](#) » [outputFormatEnum](#) » [spreadsheetML](#) value
- [bibus](#) » [outputFormatEnum](#) » [XLWA](#) value

xlsxData

Specifies the Microsoft Excel 2007 Data output format. When this format is returned inline, it is base64 encoded.

This value is represented as **Excel 2007 Data** in the portal.

Note: When a report contains multiple data queries, the first list query data is saved as output. Subsequent data queries are ignored.

Crosstab and chart data are not supported. When the first query of a report refers to crosstab or chart data, a runtime error occurs.

New in Version 10.2.0 – “Excel 2007 Data output format” on page 1841

This value was added.

XLWA

Specifies the Microsoft Excel Web Archive (XLWA) format.

This value is represented as **Excel 2002** in the portal.

XML

Specifies the Extensible Markup Language (XML) format.

packageActionEnum

Provides the list of portal actions for packages.

This enumeration set provides the capability to specify a default action to perform on a package in the portal.

References

Used by the following properties:

- [bibus » package » defaultPortalAction](#)
- [bibus » packageActionEnumProp » value](#)

Members

run

Specifies that the package should be run using the studio specified by the [bibus » package » userInterfaces](#) property.

view

Specifies that package should be viewed using IBM Cognos Connection.

pageOrientationEnum

Specifies the possible values for page orientation.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus » account » pageOrientation](#)

Members

landscape

Specifies landscape orientation.

portrait

Specifies portrait orientation.

pagingOptionEnum

Provides the list of paging options.

These options are supported by the [eventManagementService](#) service and [monitorService](#) service only.

References

Used by the following properties:

- [bibus](#) » [pagingOption](#) » [name](#)

What's new

New in Version 8.3 — “[Schedule Management](#)” on page 1907

This enumeration set was added.

Members

maximumObjects

Specifies the maximum number of objects returned in a response. To specify this option, use the [bibus](#) » [pagingOptionInt](#) class.

Use a value of 0 if you want all objects returned in a single response.

parameterCapabilityEnum

Specifies the parameter capabilities.

References

Used by the following properties:

- [bibus](#) » [baseParameter](#) » [capabilities](#)

Members

boundRange

Specifies that this parameter can contain a range with a defined start and end value.

defaultValueNotAcceptable

Specifies that the provided default value cannot be used.

discreteValue

Specifies that this parameter can contain discrete values, such as 29.

excludeValues

Specifies that values can be excluded by setting the `inclusive` property to false.

multivalued

Specifies that this parameter can contain multiple parameter values.

optional

Specifies that a value is not required for this parameter. If no value is provided, the query that contains this parameter can still run successfully.

unboundedRange

Specifies that this parameter can contain a range with an unspecified value for either the start or the end of the range.

parameterDataTypeEnum

Specifies the data type used for the parameter.

For information about XML data types, see the W3C Web site.

References

Used by the following properties:

- [bibus](#) » [baseParameter](#) » [type](#)

Members

credential

Specifies an XML element that defines a single credential used for authentication with a data source.

For more information, see the [bibus](#) » [credential](#) » [credentials](#) property.

hierarchyUniqueName

Specifies a string that uniquely identifies the Hierarchy Unique Name (HUN) of a hierarchy to which a dimension member belongs.

New in Version 10.2.0 – “Support for SAP BW hierarchy variables” on page 1839

This value was added.

memberUniqueName

Specifies a string that uniquely identifies a member of a dimension. Objects of this type are used to refer to the dimension member in a data or metadata query.

Values of this type contain data to be used only by IBM Cognos software. The structure of this data is subject to change without notice. SDK applications should not attempt to interpret or modify this data.

xsdByte

Specifies a signed 8-bit integer.

xsdDate

Specifies a date.

The date is specified in the UTC (Coordinated Universal Time) format YYYY-MM-DD.

xsdDateTime

Specifies a date and time.

The date and time are specified using the UTC format YYYY-MM-DDThh:mm:ss.fff. Fractional seconds are optional.

For example, 2005-12-22T07:52:40.65 represents 52 minutes and 40.65 seconds past 7 o'clock a.m. on December 22, 2005.

xsdDecimal

Specifies a decimal number.

xsdDouble

Specifies a double-precision floating point number.

xsdDuration

Specifies a time interval.

The time interval is specified using the ISO 8601 format PnYnMnDTnHnMnS. In this format

- nY represents the number of years

- the first nM represents the number of months
- nD represents the number of days
- T is the date-time separator
- nH represents the number of hours
- the second nM represents the number of minutes
- nS represents the number of seconds

The number of seconds can include decimal digits to arbitrary precision.

A preceding minus sign may be used to indicate a negative interval.

For example, P1Y2M3DT10H30M indicates a duration of 1 year, 2 months, 3 days, 10 hours, and 30 minutes.

xsdFloat

Specifies a single-precision floating point number.

xsdInt

Specifies a signed 32-bit integer.

xsdLong

Specifies a signed 64-bit integer.

xsdShort

Specifies a signed 16-bit integer.

xsdString

Specifies a string.

The string may contain any Unicode character.

xsdTime

Specifies a time.

The time is specified using the UTC format hh:mm:ss.fff. Fractional seconds are optional.

xsdUnsignedByte

Specifies an unsigned 8-bit integer.

xsdUnsignedInt

Specifies an unsigned 32-bit integer.

xsdUnsignedLong

Specifies an unsigned 64-bit integer.

xsdUnsignedShort

Specifies an unsigned 16-bit integer.

pdfCharacterEncodingEnum

Provides the list of character encoding options for generated PDF documents.

References

Used by the following properties:

- [bibus](#) » [batchReportService](#) » [brsPDFCharacterEncoding](#)
- [bibus](#) » [configuration](#) » [brsPDFCharacterEncoding](#)

- [bibus » configurationFolder » brsPDFCharacterEncoding](#)
- [bibus » dispatcher » brsPDFCharacterEncoding](#)
- [bibus » configuration » rsPDFCharacterEncoding](#)
- [bibus » configurationFolder » rsPDFCharacterEncoding](#)
- [bibus » dispatcher » rsPDFCharacterEncoding](#)
- [bibus » reportService » rsPDFCharacterEncoding](#)
- [bibus » pdfCharacterEncodingEnumProp » value](#)

What's new

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This enumeration set was added.

Members

auto

Specifies that characters will be encoded using the windows-1252 character encoding if all characters can be represented by that encoding. Otherwise, characters will be encoded using utf-16. This value results in slower document generation due to the additional processing required, but provides the most flexibility.

font

Specifies that characters will be encoded using the utf-16 character encoding unless the font used to render the characters is a built-in PDF font. In this case the characters will be encoded using the windows-1252 character encoding.

windows1252

Specifies that the windows-1252 character encoding will be used to store all characters in the document.

Characters that are not represented in the windows-1252 character encoding will be mapped to an arbitrary code point.

pdfCompressionTypeEnum

Provides the list of compression types for generated PDF documents. The compression type determines which objects in the PDF document are compressed. It takes longer to create a PDF document with a higher compression type but the document size is reduced.

The compression type also determines which version of Adobe Reader can be used to open the PDF document.

References

Used by the following properties:

- [bibus » batchReportService » brsPDFCompressionType](#)
- [bibus » configuration » brsPDFCompressionType](#)
- [bibus » configurationFolder » brsPDFCompressionType](#)
- [bibus » dispatcher » brsPDFCompressionType](#)
- [bibus » configuration » rsPDFCompressionType](#)
- [bibus » configurationFolder » rsPDFCompressionType](#)
- [bibus » dispatcher » rsPDFCompressionType](#)
- [bibus » reportService » rsPDFCompressionType](#)

- [bibus](#) » [pdfCompressionTypeEnumProp](#) » [value](#)

What's new

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This enumeration set was added.

Members

advanced

The second highest type of content compression. Requires Adobe Reader 6 or later.

basic

The second lowest type of content compression. Requires Adobe Reader 6 or later.

classic

The lowest type of content compression. Requires Adobe Reader 5 or later.

full

The highest type of content compression. Requires Adobe Reader 6 or later.

improved

The third lowest type of content compression. Requires Adobe Reader 6 or later.

pdfFontEmbeddingEnum

Provides the list of PDF font embedding configuration parameter values.

Note that licensed fonts are never embedded in the document.

References

Used by the following properties:

- [bibus](#) » [batchReportService](#) » [brsPDFEmbedFonts](#)
- [bibus](#) » [configuration](#) » [brsPDFEmbedFonts](#)
- [bibus](#) » [configurationFolder](#) » [brsPDFEmbedFonts](#)
- [bibus](#) » [dispatcher](#) » [brsPDFEmbedFonts](#)
- [bibus](#) » [configuration](#) » [rsPDFEmbedFonts](#)
- [bibus](#) » [configurationFolder](#) » [rsPDFEmbedFonts](#)
- [bibus](#) » [dispatcher](#) » [rsPDFEmbedFonts](#)
- [bibus](#) » [reportService](#) » [rsPDFEmbedFonts](#)
- [bibus](#) » [pdfFontEmbeddingEnumProp](#) » [value](#)

What's new

New in Version 8.3 — “PDF Configuration Parameters” on page 1915

This enumeration set was added.

Members

allow

Specifies that a font used in the document will be embedded unless it is included in the non-embeddable font list for the service generating the document.

auto

Specifies that a font used in the document will be embedded unless it is included in the non-embeddable font list for the service generating the document.

Fonts that are not listed in the embeddable font list for the service generating the document will be embedded only if characters that are not part of the windows-1252 character encoding are rendered in the document.

disallow

Specifies that a font used in the document will only be embedded if it is included in the embeddable font list for the service generating the document.

pdfOptionEnum

Provides the list of PDF options.

For more information, see [“Setting PDF Options” on page 69](#).

References

Used by the following properties:

- [bibus](#) » [pdfOption](#) » [name](#)

What's new

New in Version 8.3 — [“PDF Options - Password Protection” on page 1910](#)

This enumeration set was added.

Members

allowAccessibilitySupport

Specifies whether text in the document can be extracted for screen reader devices. To specify this option, use the [bibus](#) » [pdfOptionBoolean](#) class.

Default: false

allowAnnotations

Specifies whether text annotations can be added or modified. To specify this option, use the [bibus](#) » [pdfOptionBoolean](#) class.

Default: false

allowAssembly

Specifies whether document assembly is permitted. Document assembly functions include:

- inserting a page into the document
- deleting a page from the document
- rotating a page in the document
- the addition of navigation controls, such as bookmarks

To specify this option, use the [bibus](#) » [pdfOptionBoolean](#) class.

Default: false

allowContentCopy

Specifies whether images or text in the document can be copied. To specify this option, use the [bibus](#) » [pdfOptionBoolean](#) class.

Default: false

allowFieldCompletion

Specifies whether form fields can be filled and whether the document can be signed. To specify this option, use the [bibus » pdfOptionBoolean](#) class.

Default: false

allowModifications

Specifies whether the document content can be modified. To specify this option, use the [bibus » pdfOptionBoolean](#) class.

Default: false

allowPrintQuality

Specifies the allowed PDF print qualities. To specify this option, use the [bibus » pdfOptionPrintQuality](#) class.

Default: none

ownerPassword

Specifies the password required to open the document. To specify this option, use the [bibus » pdfOptionXMLEncodedXML](#) class.

If a password is not provided the document is not protected and can be opened by anyone.

Passwords are stored in a credential. For information about the credential format, see the [bibus » credential » credentials](#) property.

userPassword

Specifies the password required to access document options. To specify this option, use the [bibus » pdfOptionXMLEncodedXML](#) class.

If a password is not provided, any of the values specified for PDF options can be overridden in a consuming application.

Passwords are stored in a credential. For information about the credential format, see the [bibus » credential » credentials](#) property.

pdfPrintQualityEnum

Provides the list of PDF print qualities.

References

Used by the following properties:

- [bibus » pdfOptionPrintQuality » value](#)

What's new

New in Version 8.3 — “PDF Options - Password Protection” on page 1910

This enumeration set was added.

Members

low

Specifies that the document can be printed, but with low quality.

lowOrHigh

Specifies that the document can be printed either with low or with high quality.

none

Specifies that the document cannot be printed.

planningStateEnum

Reserved.

References

Used by the following properties:

- [bibus](#) » [planningApplication](#) » [applicationState](#)
- [bibus](#) » [planningStateEnumProp](#) » [value](#)

Members

development

Reserved.

production

Reserved.

portalDisplayModeEnum

Provides the list of portal display modes.

References

Used by the following properties:

- [bibus](#) » [portalOptionDisplayMode](#) » [value](#)

Members

detail

Specifies that a detail display should be used.

list

Specifies that a list display should be used.

portalListSeparatorEnum

Provides the list of portal list view separator options.

References

Used by the following properties:

- [bibus](#) » [portalOptionListSeparator](#) » [value](#)

Members

background

Specifies that background highlights should be used.

line

Specifies that lines should be used.

none

Specifies that no separator should be used.

portalOptionEnum

Provides the list of portal options.

References

Used by the following properties:

- [bibus](#) » [portalOption](#) » [name](#)

Members

automaticPageRefresh

Specifies the number of seconds that elapse between requests to refresh the current page. To specify this option, use the [bibus](#) » [portalOptionInt](#) class.

Default: 30

Use a value of 0 if you want disable automatic page refresh.

cmmPortal

Specifies CMM-specific portal options. To specify this option, use the [bibus](#) » [portalOptionXMLEncodedXML](#) class.

columnsPerPage

Specifies the number of columns in a portal page. To specify this option, use the [bibus](#) » [portalOptionInt](#) class.

Default: 3. The specified value must be no less than 1 and no greater than 5.

displayMode

Specifies the portal display mode. To specify this option, use the [bibus](#) » [portalOptionDisplayMode](#) class.

Default: [list](#)

homePage

Specifies the user's home page in the portal. To specify this option, use the [bibus](#) » [portalOptionString](#) class.

The specified string must conform to the URL query string syntax.

linesPerPage

Specifies the maximum number of lines in a portal page. To specify this option, use the [bibus](#) » [portalOptionInt](#) class.

Default: 15

listViewSeparator

Specifies the list view separator. To specify this option, use the [bibus](#) » [portalOptionListSeparator](#) class.

Default: [none](#)

pages

Reserved.

showGroupAndRolePages

Reserved.

showHiddenObjects

Specifies whether hidden objects should be shown in the portal. To specify this option, use the [bibus » portalOptionBoolean](#) class.

Default: false

showOptionSummary

Specifies whether the options used to run a task should be shown on a summary page. To specify this option, use the [bibus » portalOptionBoolean](#) class.

Default: true

showWelcomePage

Specifies whether the portal's welcome page should be shown. To specify this option, use the [bibus » portalOptionBoolean](#) class.

Default: true

skin

Specifies the portal skin. To specify this option, use the [bibus » portalOptionSearchPathSingleObject](#) class.

powerPlay8DataEnum

Provides the list of PowerPlay 8 data option values.

References

Used by the following properties:

- [bibus » powerPlay8OptionData » value](#)

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This enumeration set was added.

Members

previewWithNoData

Specifies that PowerPlay 8 will not retrieve data when running the report.

runWithAllData

Specifies that PowerPlay 8 will retrieve data when running the report.

powerPlay8OptionEnum

Provides the list of PowerPlay 8 options.

References

Used by the following properties:

- [bibus » powerPlay8Option » name](#)

What's new

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

This enumeration set was added.

Members

data

Specifies whether PowerPlay 8 will retrieve data when running the report. To specify this option, use the [bibus » powerPlay8OptionData](#) class. Possible choices are defined in the [bibus » powerPlay8DataEnum](#) enumeration set.

Default: [runWithAllData](#)

email

Specifies whether the output will be distributed by email. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

Default: `false`

emailAsAttachment

Specifies whether the output will be distributed as an email attachment. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

Default: `false`

emailAsURL

Specifies whether a link to the output will be sent as an email attachment. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

Default: `false`

outputFormat

Specifies the output format. To specify this option, use the [bibus » powerPlay8OptionOutputFormat](#) class. Possible choices are defined in the [bibus » powerPlay8OutputFormatEnum](#) enumeration set.

Default: [PDF](#)

outputLocale

Specifies the language of the output. The value is expressed as a hyphenated language-region pair, in accordance with the RFC3066 standard.

To specify a locale other than the default output locale, specify a single or multiple locales in conjunction with any of the following run options:

- [email](#)
- [print](#)
- [saveAs](#)
- [saveOutput](#)

To specify this option, use the [bibus » powerPlay8OptionLanguageArray](#) class.

Default: an array with the single value `en-us`

outputLocation

Specifies the location for storing temporary objects created by [powerPlayService](#).

To specify this option, use the [bibus » powerPlay8OptionAnyURI](#) class and values from the [bibus » temporaryObjectLocationEnum](#) enumeration set.

Default: [contentStore](#)

New in Version 10.1.0 – “Storing Temporary Output Objects Outside the Content Store” on page 1857

This value was added.

print

Specifies whether the new output will be printed. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

Default: `false`

printer

Identifies the instance of the [bibus » printer](#) class that will be used to print the report. To specify this option, use the [bibus » powerPlay8OptionSearchPathSingleObject](#) class.

printerAddress

Identifies the network address of the printer that will be used to print the report. To specify this option, use the [bibus » powerPlay8OptionString](#) class.

prompt

Specifies whether PowerPlay 8 will issue prompts, so that users can enter report option values. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

Default: `true`

saveAs

Contains the information required to create a new object in Content Manager based on the object to be run. To specify this option, use the [bibus » powerPlay8OptionSaveAs](#) class.

This option cannot be used with the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

saveOutput

Specifies whether the output should be saved to the content store. To specify this option, use the [bibus » powerPlay8OptionBoolean](#) class.

This option cannot be used with the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

Default: `false`

powerPlay8OutputFormatEnum

Provides the list of PowerPlay 8 output format option values.

References

Used by the following properties:

- [bibus » basePowerPlay8Report » defaultOutputFormat](#)
- [bibus » powerPlay8OptionOutputFormat » value](#)
- [bibus » powerPlay8OutputFormatEnumArrayProp » value](#)
- [bibus » powerPlay8OutputFormatEnumProp » value](#)

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This enumeration set was added.

Members

PDF

Specifies the Acrobat Portable Document Format (PDF).

powerPlay8SaveAsEnum

Lists the classes of objects that can be created as a result of a [saveAs](#) option for a [bibus » powerPlay8Report](#).

References

Used by the following properties:

- [bibus » powerPlay8OptionSaveAs » objectClass](#)

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This enumeration set was added.

Members

powerPlay8ReportView

Specifies that the [bibus » basePowerPlay8Report](#) is saved as an instance of the [bibus » powerPlay8ReportView](#) class, capturing all of the parameters and options specified in the [asynch » run\(objectPath, parameterValues, options\)](#) method call.

powerPlayDataBlockEnum

Specifies the set of [powerPlayDataBlock](#) data types.

References

Used by the following properties:

- [bibus » powerPlayDataBlock » id](#)

Members

parameters

Specifies that the block contains parameter values.

reportConfiguration

Specifies that the block contains the report configuration.

reportSpecification

Specifies that the block contains the report specification.

powerPlayDataEnum

Provides the list of PowerPlay IBM Cognos PowerPlay data option values.

References

Used by the following properties:

- [bibus » powerPlayOptionData » value](#)

Members

previewWithNoData

Specifies that PowerPlay will not retrieve data when running the report.

runWithAllData

Specifies that PowerPlay will retrieve data when running the report.

powerPlayOptionEnum

Provides the list of PowerPlay options.

References

Used by the following properties:

- [bibus » powerPlayOption » name](#)

Members

data

Specifies whether PowerPlay will retrieve data when running the report. To specify this option, use the [bibus » powerPlayOptionData](#) class. Possible choices are defined in the [bibus » powerPlayDataEnum](#) enumeration set.

Default: [runWithAllData](#)

outputFormat

Specifies the output format. To specify this option, use the [bibus » powerPlayOptionOutputFormat](#) class. Possible choices are defined in the [bibus » powerPlayOutputFormatEnum](#) enumeration set.

Default: [HTML](#)

prompt

Specifies whether PowerPlay will issue prompts, so that users can enter report option values. To specify this option, use the [bibus » powerPlayOptionBoolean](#) class.

Default: `true`

powerPlayOutputFormatEnum

Provides the list of IBM Cognos PowerPlay output format option values.

References

Used by the following properties:

- [bibus » powerPlayOptionOutputFormat » value](#)

Members

HTML

HyperText Markup Language (HTML) format.

PDF

Acrobat Portable Document Format (PDF).

powerPlayReportActionEnum

Provides the list of portal actions for IBM Cognos PowerPlay reports.

This enumeration set provides the capability to specify a default action to perform on a PowerPlay report in the portal.

References

Used by the following properties:

- [bibus](#) » [powerPlayReport](#) » [defaultPortalAction](#)
- [bibus](#) » [powerPlayReportActionEnumProp](#) » [value](#)

Members

edit

Specifies that the PowerPlay report should be edited with the associated studio if the user has the necessary permissions and capabilities.

run

Specifies that the PowerPlay report should be run if the user has the necessary permissions and capabilities.

promptCacheModeEnum

Provides the list of actions the report service can perform on a report's prompt cache.

References

Used by the following properties:

- [bibus](#) » [runOptionPromptCacheMode](#) » [value](#)

What's new

New in Version 8.4 — [“On Demand Refresh of Prompt Cache” on page 1884](#)

This enumeration set was added.

Members

create

Specifies that the report's prompt cache should be created.

If the report's prompt cache already exists it is replaced.

Cache entries for the locales specified by the [outputLocale](#) option are added to the report's prompt cache.

none

Specifies that no action be performed on the report's prompt cache.

refresh

Specifies that the report's prompt cache entries for specified locales should be refreshed. If an existing prompt cache entry for a specified locale has expired, it is replaced. If a report's prompt cache entry for a specified locale does not exist, it is created.

Specify the prompt cache entries to be refreshed using the `outputLocale` run option. Report prompt cache entries for locales not specified by this run option are not affected when the report's prompt cache is refreshed.

update

Specifies that the report's prompt cache should be updated.

If the report's prompt cache does not exist, it is created.

Cache entries for the locales specified by the `outputLocale` option are added to the report's prompt cache, replacing any cache entries for the specified locales. Cache entries for locales not specified by the `outputLocale` option are not affected when the report's prompt cache is updated.

promptTypeEnum

Defines the list of IBM Cognos Analytics prompt types.

References

Used by the following properties:

- [bibus](#) » [baseParameter](#) » [promptType](#)

Members

editBox

Specifies that the prompt will be displayed as an edit box.

selectDate

Specifies that the prompt will display a selectable date.

selectDateTime

Specifies that the prompt will display a selectable date and time.

selectInterval

Specifies that the prompt will display a selectable interval.

selectTime

Specifies that the prompt will display a selectable time.

selectValue

Specifies that the prompt will display a list of values from which the user can select.

selectWithSearch

Specifies that the prompt will be displayed as a user-defined search.

selectWithTree

Specifies that the prompt will be displayed as a tree.

propEnum

Defines the set of properties for all classes stored in the content store.

References

Used by the following properties:

- [bibus](#) » [retentionRule](#) » [prop](#)
- [bibus](#) » [queryRequest](#) » [properties](#)

- [bibus » refProp » properties](#)
- [bibus » sort » propName](#)
- [bibus » refProp » refPropName](#)
- [bibus » addOptions » returnProperties](#)
- [bibus » updateOptions » returnProperties](#)
- [bibus » contentManagerQueryOptionPropEnumArray » value](#)

Used by the following method parameters:

- [content » query\(searchPath, properties, sortBy, options\) » properties](#)

Members

aasAffineConnections

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasAuditLevel

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasExecutionTimeLimit

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasMaximumProcesses

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasNonAffineConnections

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasPeakAffineConnections

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasPeakMaximumProcesses

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

aasPeakNonAffineConnections

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

action

Defined by the class [bibus » drillPath](#).

active

Defined by the classes [bibus » namespace](#) and [bibus » schedule](#).

activeJMXProxyURI

Defined by the class [bibus » configuration](#).

actualCompletionTime

Defined by the class [bibus » history](#).

actualExecutionTime

Defined by the class [bibus » history](#).

addedObjectCount

Defined by the class [bibus » historyDetailDeploymentSummary](#).

advancedSettings

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » agentService](#), [bibus » annotationService](#), [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerCacheService](#), [bibus » contentManagerService](#), [bibus » dataAdvisorService](#), [bibus » dataIntegrationService](#), [bibus » dataMovementService](#), [bibus » deliveryService](#), [bibus » dimensionManagementService](#), [bibus » dispatcher](#), [bibus » eventManagementService](#), [bibus » EVService](#), [bibus » graphicsService](#), [bibus » humanTaskService](#), [bibus » idVizService](#), [bibus » indexDataService](#), [bibus » indexSearchService](#), [bibus » indexUpdateService](#), [bibus » jobService](#), [bibus » logService](#), [bibus » metadataService](#), [bibus » metricsManagerService](#), [bibus » migrationService](#), [bibus » mobileService](#), [bibus » monitorService](#), [bibus » planningAdministrationConsoleService](#), [bibus » planningDataService](#), [bibus » planningRuntimeService](#), [bibus » planningTaskService](#), [bibus » powerPlayService](#), [bibus » presentationService](#), [bibus » queryService](#), [bibus » relationalMetadataService](#), [bibus » reportDataService](#), [bibus » reportService](#), [bibus » repositoryService](#), [bibus » saCAMService](#), and [bibus » systemService](#).

aggregates

Defined by the class [bibus » rolapDataSource](#).

allowAnnotations

Defined by the classes [bibus » agentOutputHotList](#) and [bibus » reportVersion](#).

allowNotification

Defined by the classes [bibus » baseAgentDefinition](#) and [bibus » baseReport](#).

allowSubscription

Defined by the class [bibus » baseReport](#).

ancestors

Defined by the class [bibus » baseClass](#).

ansAnnotationLifetime

Defined by the classes [bibus » annotationService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

ansAuditLevel

Defined by the classes [bibus » annotationService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

applicationGUID

Defined by the class [bibus » planningApplication](#).

applicationID

Defined by the class [bibus » planningApplication](#).

applicationState

Defined by the class [bibus » planningApplication](#).

applicationURL

Defined by the class [bibus » planningApplication](#).

asAuditLevel

Defined by the classes [bibus » agentService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

asConnections

Defined by the classes [bibus » agentService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

asMaximumEMailAttachmentSize

Defined by the classes [bibus » agentService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

asOfTime

Defined by the class [bibus » reportVersion](#).

asPeakConnections

Defined by the classes [bibus » agentService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

base

Defined by the classes [bibus » agentDefinitionView](#), [bibus » dataMovementTaskAlias](#), [bibus » personalization](#), [bibus » portalSkin](#), [bibus » powerPlay8ReportView](#), and [bibus » reportView](#).

binding

Defined by the class [bibus » portletProducer](#).

bindingName

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

bookmarkItem

Defined by the class [bibus » drillPath](#).

bookmarkText

Defined by the class [bibus » drillPath](#).

bpmRestURI

Defined by the class [bibus » configuration](#).

brsAffineConnections

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsAuditLevel

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsAuditNativeQuery

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsChartHotspotLimit

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsDataSourceChange

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsExecutionTimeLimit

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsMaximumEMailAttachmentSize

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsMaximumProcesses

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsNonAffineConnections

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPDFCharacterEncoding

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPDFCompressionLevel

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPDFCompressionType

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPDFEmbedFonts

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPeakAffineConnections

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPeakMaximumProcesses

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

brsPeakNonAffineConnections

Defined by the classes [bibus » batchReportService](#), [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

bulkEvents

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

burstID

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

burstKey

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

businessPhone

Defined by the classes [bibus » account](#) and [bibus » contact](#).

canBurst

Defined by the classes [bibus » authoredReport](#) and [bibus » reportVersion](#).

canCustomize

Defined by the class [bibus » portlet](#).

capabilities

Defined by the classes [bibus » baseROLAPDataSource](#), [bibus » dataSource](#), and [bibus » namespace](#).

capacity

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

channel

Defined by the class [bibus » baseRSSTask](#).

closeConnectionCommands

Defined by the classes [bibus » dataSource](#) and [bibus » dataSourceConnection](#).

closeSessionCommands

Defined by the classes [bibus » dataSource](#) and [bibus » dataSourceConnection](#).

cmcsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerCacheService](#), and [bibus » dispatcher](#).

cmcsHeapLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerCacheService](#), and [bibus » dispatcher](#).

cmsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerService](#), and [bibus » dispatcher](#).

cmsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerService](#), and [bibus » dispatcher](#).

cmsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » contentManagerService](#), and [bibus » dispatcher](#).

collaborationDiscoveryURI

Defined by the class [bibus » configuration](#).

componentID

Defined by the class [bibus » installedComponent](#).

configuration

Defined by the class [bibus » uiProfile](#).

connections

Defined by the class [bibus » baseDataIntegrationTask](#).

connectionString

Defined by the class [bibus » dataSourceConnection](#).

consumers

Defined by the class [bibus » dataSourceSignon](#).

contact

Defined by the classes [bibus » adminFolder](#), [bibus » aliasRoot](#), [bibus » annotation](#), [bibus » annotationFolder](#), [bibus » archiveLocation](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay8Report](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » baseROLAPDataSource](#), [bibus » contentTask](#), [bibus » dashboard](#), [bibus » dataSource](#), [bibus » distributionList](#), [bibus » document](#), [bibus » documentVersion](#), [bibus » drillPath](#), [bibus » favoritesFolder](#), [bibus » folder](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » launchable](#), [bibus » migrationTask](#), [bibus » model](#), [bibus » mruFolder](#), [bibus » namespace](#), [bibus » package](#), [bibus » pagelet](#), [bibus » pageletFolder](#), [bibus » personalization](#), [bibus » personalizationFolder](#), [bibus » planningApplication](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » portalPackage](#), [bibus » portalSkin](#), [bibus » portlet](#), [bibus » portletFolder](#), [bibus » portletProducer](#), [bibus » printer](#), [bibus » queryServiceTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), [bibus » resource](#), [bibus » subscriptionFolder](#), [bibus » tenant](#), [bibus » uiProfile](#), [bibus » uiProfileFolder](#), and [bibus » URL](#).

contactEMail

Defined by the classes [bibus » adminFolder](#), [bibus » aliasRoot](#), [bibus » annotation](#), [bibus » annotationFolder](#), [bibus » archiveLocation](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay8Report](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » baseROLAPDataSource](#), [bibus » contentTask](#), [bibus » dashboard](#), [bibus » dataSource](#), [bibus » distributionList](#), [bibus » document](#), [bibus » documentVersion](#), [bibus » drillPath](#), [bibus » favoritesFolder](#), [bibus » folder](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » launchable](#), [bibus » migrationTask](#), [bibus » model](#), [bibus » mruFolder](#), [bibus » namespace](#), [bibus » package](#), [bibus » pagelet](#), [bibus » pageletFolder](#), [bibus » personalization](#), [bibus » personalizationFolder](#), [bibus » planningApplication](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » portalPackage](#), [bibus » portalSkin](#), [bibus » portlet](#), [bibus » portletFolder](#), [bibus » portletProducer](#), [bibus » printer](#), [bibus » queryServiceTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), [bibus » resource](#), [bibus » subscriptionFolder](#), [bibus » tenant](#), [bibus » uiProfile](#), [bibus » uiProfileFolder](#), and [bibus » URL](#).

contentLocale

Defined by the class [bibus » account](#).

contentType

Defined by the class [bibus » URL](#).

context

Defined by the classes [bibus » dashboard](#), [bibus » documentContent](#), [bibus » output](#), [bibus » pageletInstance](#), [bibus » personalization](#), [bibus » portlet](#), and [bibus » portletInstance](#).

contextBlockCount

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

cookieCAMPassportHttpOnly

Defined by the class [bibus » configuration](#).

creationTime

Defined by the class [bibus » baseClass](#).

credential

Defined by the class [bibus » schedule](#).

credentialNamespaces

Defined by the class [bibus » dataSourceConnection](#).

credentials

Defined by the classes [bibus » credential](#), [bibus » dataSourceCredential](#), and [bibus » dataSourceSignon](#).

crosstabItemDisplayCountDefault

Defined by the class [bibus » packageConfiguration](#).

crosstabItemDisplayCountLimit

Defined by the class [bibus » packageConfiguration](#).

cubeContentName

Defined by the class [bibus » basePowerPlayClass](#).

dailyPeriod

Defined by the class [bibus » schedule](#).

dasAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataAdvisorService](#), and [bibus » dispatcher](#).

data

Defined by the classes [bibus » cacheOutput](#), [bibus » dashboard](#), [bibus » documentContent](#), [bibus » graphic](#), [bibus » historyDetailDataMovementService](#), [bibus » output](#), [bibus » page](#), and [bibus » visualization](#).

dataBlocks

Defined by the class [bibus » basePowerPlayClass](#).

dataModificationTime**dataPathInfo**

Defined by the classes [bibus » graphic](#), [bibus » output](#), and [bibus » visualization](#).

dataSetBase**dataSetSize****dataSize**

Defined by the classes [bibus » cacheOutput](#), [bibus » dashboard](#), [bibus » documentContent](#), [bibus » graphic](#), [bibus » historyDetailDataMovementService](#), [bibus » output](#), [bibus » page](#), and [bibus » visualization](#).

dataSourceAccessAccount

Defined by the class [bibus » baseROLAPDataSource](#).

dataSourceConnectionName

Defined by the class [bibus » dataSourceCredential](#).

dataSourceName

Defined by the class [bibus » dataSourceCredential](#).

dataType

Defined by the classes [bibus » cacheOutput](#), [bibus » dashboard](#), [bibus » documentContent](#), [bibus » graphic](#), [bibus » historyDetailDataMovementService](#), [bibus » output](#), [bibus » page](#), and [bibus » visualization](#).

defaultAnalysis

Defined by the class [bibus » packageConfiguration](#).

defaultDescription

Defined by the class [bibus » uiClass](#).

defaultName

Defined by the class [bibus » baseClass](#).

defaultOutputFormat

Defined by the classes [bibus » basePowerPlay8Report](#) and [bibus » baseReport](#).

defaultPortalAction

Defined by the classes [bibus » baseAgentDefinition](#), [bibus » basePowerPlay8Report](#), [bibus » baseReport](#), [bibus » package](#), and [bibus » powerPlayReport](#).

defaultScreenTip

Defined by the class [bibus » uiClass](#).

defaultTriggerDescription

Defined by the class [bibus » authoredReport](#).

definition

Defined by the classes [bibus » agentState](#) and [bibus » agentTaskState](#).

deletedObjectCount

Defined by the class [bibus » historyDetailDeploymentSummary](#).

deployedObject

Defined by the class [bibus » deploymentDetail](#).

deployedObjectAncestorDefaultNames

Defined by the class [bibus » deploymentDetail](#).

deployedObjectClass

Defined by the class [bibus » deploymentDetail](#).

deployedObjectDefaultName

Defined by the class [bibus » deploymentDetail](#).

deployedObjectStatus

Defined by the class [bibus » deploymentDetail](#).

deployedObjectUsage

Defined by the class [bibus » deploymentDetail](#).

deploymentReferences

Defined by the classes [bibus » dashboard](#), [bibus » drillPath](#), [bibus » launchable](#), [bibus » pageletInstance](#), [bibus » personalization](#), [bibus » portlet](#), and [bibus » portletInstance](#).

description

Defined by the class [bibus » uiClass](#).

detail

Defined by the class [bibus » baseHistoryDetail](#).

detailTime

Defined by the class [bibus » baseHistoryDetail](#).

dimsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

dimsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dimensionManagementService](#), and [bibus » dispatcher](#).

disabled

Defined by the class [bibus » baseClass](#).

disAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataIntegrationService](#), and [bibus » dispatcher](#).

disConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataIntegrationService](#), and [bibus » dispatcher](#).

dispatcherAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

dispatcherID

Defined by the classes [bibus » dispatcher](#) and [bibus » history](#).

dispatcherPath

Defined by the class [bibus » dispatcher](#).

disPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataIntegrationService](#), and [bibus » dispatcher](#).

displaySequence

Defined by the class [bibus » uiClass](#).

distributionMembers

Defined by the classes [bibus » group](#) and [bibus » role](#).

dmsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

dmsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dataMovementService](#), and [bibus » dispatcher](#).

documentType

Defined by the class [bibus » document](#).

dsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » deliveryService](#), and [bibus » dispatcher](#).

dsCompressAttachmentLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » deliveryService](#), and [bibus » dispatcher](#).

dsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » deliveryService](#), and [bibus » dispatcher](#).

dsMaximumEMailSize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » deliveryService](#), and [bibus » dispatcher](#).

dsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » deliveryService](#), and [bibus » dispatcher](#).

edition

Defined by the class [bibus » dispatcher](#).

editions

Defined by the class [bibus » configuration](#).

effectiveUserCapabilities

Defined by the classes [bibus » content](#), [bibus » folder](#), and [bibus » package](#).

email

Defined by the classes [bibus » account](#) and [bibus » contact](#).

emsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » eventManagementService](#).

endDate

Defined by the class [bibus » schedule](#).

endType

Defined by the class [bibus » schedule](#).

eventID

Defined by the class [bibus » history](#).

eventKey

Defined by the classes [bibus » agentTaskDefinition](#) and [bibus » authoredAgentDefinition](#).

eventTypes

Defined by the class [bibus » agentTaskDefinition](#).

everyNPeriods

Defined by the class [bibus » schedule](#).

evsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » EVService](#).

executionFormat

Defined by the class [bibus » baseReport](#).

executionLocale

Defined by the class [bibus » baseReport](#).

executionPageDefinition

Defined by the class [bibus » baseReport](#).

executionPageOrientation

Defined by the class [bibus » baseReport](#).

executionPrompt

Defined by the class [bibus » baseReport](#).

expirationTime

Defined by the classes [bibus » agentOutputHotList](#), [bibus » documentVersion](#), [bibus » history](#), [bibus » model](#), [bibus » reportCache](#), [bibus » reportVersion](#), and [bibus » shortcut](#).

faxPhone

Defined by the classes [bibus » account](#) and [bibus » contact](#).

filterDataItemName

Defined by the class [bibus » agentTaskDefinition](#).

format

Defined by the classes [bibus » account](#), [bibus » contact](#), [bibus » documentContent](#), and [bibus » output](#).

gateway

Defined by the class [bibus » basePowerPlayClass](#).

givenName

Defined by the classes [bibus » account](#) and [bibus » contact](#).

glossaryURI

Defined by the class [bibus » configuration](#).

governors

Defined by the classes [bibus » group](#) and [bibus » role](#).

groupAndRoleSettings

Defined by the class [bibus » session](#).

gsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

gsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » graphicsService](#).

handle

Defined by the class [bibus » portlet](#).

hasChildren

Defined by the class [bibus » baseClass](#).

hasMessage

Defined by the classes [bibus » deploymentDetail](#) and [bibus » historyDetailMigrationService](#).

hasPrompts

Defined by the class [bibus » powerPlayReport](#).

height

Defined by the class [bibus » pageDefinition](#).

hidden

Defined by the class [bibus » uiClass](#).

homePhone

Defined by the classes [bibus » account](#) and [bibus » contact](#).

horizontalElementsRenderingLimit

Defined by the classes [bibus » account](#) and [bibus » contact](#).

htsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » humanTaskService](#).

htsCompletedTaskLifetime

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » humanTaskService](#).

iconURI

Defined by the class [bibus » uiClass](#).

identity

Defined by the class [bibus » session](#).

idsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexDataService](#).

idsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexDataService](#).

idsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexDataService](#).

idVizAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » idVizService](#).

inputMessageName

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

intradayRecurrenceEnd

Defined by the class [bibus » schedule](#).

intradayRecurrenceInterval

Defined by the class [bibus » schedule](#).

intradayRecurrenceStart

Defined by the class [bibus » schedule](#).

isolationLevel

Defined by the class [bibus » dataSourceConnection](#).

issAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexSearchService](#).

issConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexSearchService](#).

issPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexSearchService](#).

iusAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexUpdateService](#).

iusConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexUpdateService](#).

iusPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » indexUpdateService](#).

jmxProxyHostDispatchers

Defined by the class [bibus » configuration](#).

jsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » jobService](#).

jsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » jobService](#).

jsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » jobService](#).

lastConfigurationModificationTime

Defined by the class [bibus » configuration](#).

lastPage

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

launchableType

Defined by the class [bibus » launchable](#).

layout

Defined by the class [bibus » pagelet](#).

link

Defined by the classes [bibus » shortcutAgentRSSTask](#), [bibus » shortcutRSSTask](#), and [bibus » urlRSSTask](#).

linkPaths

Defined by the class [bibus » historyDetailRelatedReports](#).

loadBalancingMode

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

locale

Defined by the classes [bibus » cacheOutput](#), [bibus » contact](#), [bibus » documentContent](#), and [bibus » output](#).

location

Defined by the class [bibus » printer](#).

maximumDetailSeverity

Defined by the class [bibus » history](#).

mbsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » mobileService](#).

mbsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » mobileService](#).

mbsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » mobileService](#).

mdsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

mdsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metadataService](#).

memberDisplayCountDefault

Defined by the class [bibus » packageConfiguration](#).

memberDisplayCountLimit

Defined by the class [bibus » packageConfiguration](#).

members

Defined by the classes [bibus » distributionList](#), [bibus » group](#), and [bibus » role](#).

message

Defined by the classes [bibus » deploymentDetail](#) and [bibus » historyDetailMigrationService](#).

metadata

Defined by the class [bibus » pagelet](#).

metadataInformationURI

Defined by the class [bibus » configuration](#).

metadataModel

Defined by the classes [bibus » authoredPowerPlay8Report](#), [bibus » authoredReport](#), [bibus » baseDataIntegrationTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), and [bibus » storedProcedureTask](#).

metadataModelPackage

Defined by the classes [bibus » authoredPowerPlay8Report](#), [bibus » authoredReport](#), [bibus » baseDataIntegrationTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), and [bibus » storedProcedureTask](#).

migratedObject

Defined by the class [bibus » historyDetailMigrationService](#).

misAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » migrationService](#).

misConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » migrationService](#).

misPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » migrationService](#).

mmsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metricsManagerService](#).

mmsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metricsManagerService](#).

mmsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » metricsManagerService](#).

mobileConfiguration

Defined by the class [bibus » configuration](#).

mobileDeviceID

Defined by the class [bibus » account](#).

mobilePhone

Defined by the classes [bibus » account](#) and [bibus » contact](#).

model

Defined by the class [bibus » model](#).

modelName

Defined by the class [bibus » printer](#).

modificationTime

Defined by the class [bibus » baseClass](#).

monthlyAbsoluteDay

Defined by the class [bibus » schedule](#).

monthlyRelativeDay

Defined by the class [bibus » schedule](#).

monthlyRelativeWeek

Defined by the class [bibus » schedule](#).

mostRecentEventList

Defined by the class [bibus » baseAgentDefinition](#).

msAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » monitorService](#).

name

Defined by the class [bibus » baseClass](#).

namespaceFormat

Defined by the class [bibus » namespace](#).

nonPeakDemandBeginHour

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

notificationEMail

Defined by the class [bibus » account](#).

notificationList

Defined by the classes [bibus » baseAgentDefinition](#) and [bibus » baseReport](#).

notificationListIsEmpty

Defined by the classes [bibus » baseAgentDefinition](#) and [bibus » baseReport](#).

objectClass

Defined by the class [bibus » baseClass](#).

openConnectionCommands

Defined by the classes [bibus » dataSource](#) and [bibus » dataSourceConnection](#).

openSessionCommands

Defined by the classes [bibus » dataSource](#) and [bibus » dataSourceConnection](#).

operationName

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

options

Defined by the classes [bibus » account](#), [bibus » agentTaskDefinition](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay&Report](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » drillPath](#), [bibus » exportDeployment](#), [bibus » group](#), [bibus » historyDetailRequestArguments](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » jobStepDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » model](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » queryServiceTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), [bibus » role](#), [bibus » schedule](#), [bibus » storedProcedureTask](#), and [bibus » webServiceTask](#).

output

Defined by the classes [bibus » historyDetailAgentService](#) and [bibus » historyDetailReportService](#).

outputMessageName

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

outputs

Defined by the class [bibus » annotation](#).

overrideOptions

Defined by the class [bibus » configuration](#).

owner

Defined by the class [bibus » baseClass](#).

ownerEventID

Defined by the class [bibus » history](#).

ownerPassport

Defined by the classes [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), and [bibus » baseReport](#).

packageBase

Defined by the classes [bibus » agentTaskDefinition](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » basePowerPlayClass](#), [bibus » baseReport](#), [bibus » folder](#), [bibus » jobDefinition](#), [bibus » jobStepDefinition](#), [bibus » model](#), [bibus » package](#), [bibus » planningApplication](#), and [bibus » URL](#).

pacsauditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningAdministrationConsoleService](#).

pacsaconnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningAdministrationConsoleService](#).

pacsapeakconnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningAdministrationConsoleService](#).

page

Defined by the class [bibus » account](#).

pagelet

Defined by the class [bibus » pageletInstance](#).

pageorientation

Defined by the class [bibus » account](#).

pagerPhone

Defined by the classes [bibus » account](#) and [bibus » contact](#).

parameterAssignments

Defined by the classes [bibus » agentTaskDefinition](#) and [bibus » drillPath](#).

parameters

Defined by the classes [bibus » account](#), [bibus » agentTaskDefinition](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay8Report](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » drillPath](#), [bibus » exportDeployment](#), [bibus » historyDetailRelatedReports](#), [bibus » historyDetailRequestArguments](#),

[bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » jobStepDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » queryServiceTask](#), [bibus » reportCache](#), [bibus » reportVersion](#), [bibus » schedule](#), [bibus » storedProcedureTask](#), and [bibus » webServiceTask](#).

parent

Defined by the class [bibus » baseClass](#).

paths

Defined by the class [bibus » authoredReport](#).

pdsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsEListAccessCacheLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

pdsShowCellAnnotations

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningDataService](#).

peakDemandBeginHour

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), and [bibus » dispatcher](#).

periodicalDocumentVersionRetentionAge

Defined by the class [bibus » configuration](#).

periodicalDocumentVersionRetentionCount

Defined by the class [bibus » configuration](#).

periodicalProducer

Defined by the class [bibus » schedule](#).

permissions

Defined by the class [bibus » baseClass](#).

policies

Defined by the class [bibus » baseClass](#).

portalPage

Defined by the class [bibus » account](#).

portalPages

Defined by the class [bibus » account](#).

portlet

Defined by the class [bibus » portletInstance](#).

position

Defined by the class [bibus » baseClass](#).

postalAddress

Defined by the classes [bibus » account](#) and [bibus » contact](#).

powerPlay8Configuration

Defined by the classes [bibus » authoredPowerPlay8Report](#), [bibus » content](#), [bibus » folder](#), and [bibus » package](#).

ppsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsMaximumEMailAttachmentSize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

ppsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » powerPlayService](#).

previewImageLocation

Defined by the class [bibus » portalSkin](#).

printerAddress

Defined by the class [bibus » printer](#).

priority

Defined by the class [bibus » schedule](#).

producer

Defined by the class [bibus » periodical](#).

productLocale

Defined by the class [bibus » account](#).

properties

Defined by the class [bibus » systemMetricThresholds](#).

prsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningRuntimeService](#).

prsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningRuntimeService](#).

prsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningRuntimeService](#).

psAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » presentationService](#).

ptsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningTaskService](#).

ptsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningTaskService](#).

ptsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » planningTaskService](#).

published

Defined by the class [bibus » portalSkin](#).

qsAdditionalJVMArguments

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsDiagnosticsEnabled

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsDisableQueryPlanCache

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsDisableVerboseGCLogging

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsDumpModelToFile

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsGCPolicy

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsGenerateCommentsInNativeSQL

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsIdleConnectionTimeout

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsInitialJVMHeapSize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsInitialJVMNurserySize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsJVMHeapSizeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsJVMNurserySizeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsManualCubeStart

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsMetricsEnabled

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsMultiDimensionalQuerySizeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsQueryExecutionTrace

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsQueryPlanningTrace

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsResultSetCacheQueryTimeThreshold

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsROLAPCubeAdministrationCommandTimeout

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsROLAPCubeConfigurations

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsROLAPMemberCacheAliasRoot

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qsVerboseGCLogLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » queryService](#).

qualifier

Defined by the class [bibus » dataSourceNameBinding](#).

queryMode

Defined by the class [bibus » authoredReport](#).

rdsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportDataService](#).

rdsGatewayMappings

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportDataService](#).

rdsMaximumDataSize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportDataService](#).

recipients

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

recipientsEMail

Defined by the classes [bibus » documentContent](#) and [bibus » output](#).

registration

Defined by the class [bibus » portletProducer](#).

related

Defined by the classes [bibus » agentTaskState](#) and [bibus » historyDetailRelatedHistory](#).

replacedObjectCount

Defined by the class [bibus » historyDetailDeploymentSummary](#).

replacement

Defined by the class [bibus » dataSourceNameBinding](#).

report

Defined by the class [bibus » reportDataServiceAgentDefinition](#).

reposAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » repositoryService](#).

reposCacheObjTTL

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » repositoryService](#).

repositoryRules

Defined by the classes [bibus » account](#), [bibus » content](#), [bibus » folder](#), [bibus » namespace](#), [bibus » namespaceFolder](#), and [bibus » package](#).

reposNumObjDisk

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » repositoryService](#).

reposNumObjMem

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » repositoryService](#).

requestedExecutionTime

Defined by the class [bibus » history](#).

resourceLocation

Defined by the class [bibus » portalSkin](#).

resourceType

Defined by the class [bibus » resource](#).

restartEventID

Defined by the class [bibus » history](#).

retentions

Defined by the classes [bibus » agentState](#), [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » basePowerPlay8Report](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » document](#), [bibus » exportDeployment](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » mruFolder](#), [bibus » package](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » storedProcedureTask](#), and [bibus » webServiceTask](#).

rmdsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsPeakConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

rmdsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » relationalMetadataService](#).

routingHints

Defined by the classes [bibus » account](#), [bibus » dashboard](#), [bibus » group](#), [bibus » package](#), [bibus » portletProducer](#), and [bibus » role](#).

routingServerGroup

Defined by the classes [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay&Report](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » dashboard](#), [bibus » exportDeployment](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » package](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » portletProducer](#), [bibus » queryServiceTask](#), [bibus » storedProcedureTask](#), and [bibus » webServiceTask](#).

routingServerGroupEdition

Defined by the classes [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), [bibus » basePowerPlay&Report](#), [bibus » baseReport](#), [bibus » baseRSSTask](#), [bibus » contentTask](#), [bibus » dashboard](#), [bibus » exportDeployment](#), [bibus » humanTask](#), [bibus » importDeployment](#), [bibus » indexUpdateTask](#), [bibus » jobDefinition](#), [bibus » memo](#), [bibus » migrationTask](#), [bibus » package](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » portalSkin](#), [bibus » portletProducer](#), [bibus » queryServiceTask](#), [bibus » storedProcedureTask](#), and [bibus » webServiceTask](#).

routingTable

Defined by the class [bibus » configuration](#).

rsAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsAuditNativeQuery

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsChartHotspotLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsDataSourceChange

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsExecutionTimeLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsMaximumEMailAttachmentSize

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPDFCharacterEncoding

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPDFCompressionLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPDFCompressionType

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPDFEmbedFonts

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPeakAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPeakMaximumProcesses

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsPeakNonAffineConnections

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

rsQueueLimit

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » reportService](#).

runAsOwner

Defined by the classes [bibus » baseAgentDefinition](#), [bibus » baseDataIntegrationTask](#), [bibus » baseDataMovementTask](#), and [bibus » baseReport](#).

runCondition

Defined by the class [bibus » agentTaskDefinition](#).

runningState

Defined by the classes [bibus » adaptiveAnalyticsService](#), [bibus » agentService](#), [bibus » annotationService](#), [bibus » batchReportService](#), [bibus » contentManagerCacheService](#), [bibus » contentManagerService](#), [bibus » dataAdvisorService](#), [bibus » dataIntegrationService](#), [bibus » dataMovementService](#), [bibus » deliveryService](#), [bibus » dimensionManagementService](#), [bibus » dispatcher](#), [bibus » eventManagementService](#), [bibus » EVService](#), [bibus » graphicsService](#), [bibus » humanTaskService](#), [bibus » idVizService](#), [bibus » indexDataService](#), [bibus » indexSearchService](#), [bibus » indexUpdateService](#), [bibus » jobService](#), [bibus » logService](#), [bibus » metadataService](#), [bibus » metricsManagerService](#), [bibus » migrationService](#), [bibus » mobileService](#), [bibus » monitorService](#), [bibus » planningAdministrationConsoleService](#), [bibus » planningDataService](#), [bibus » planningRuntimeService](#), [bibus » planningTaskService](#), [bibus » powerPlayService](#), [bibus » presentationService](#), [bibus » queryService](#), [bibus » relationalMetadataService](#), [bibus » reportDataService](#), [bibus » reportService](#), [bibus » repositoryService](#), [bibus » saCAMService](#), and [bibus » systemService](#).

runWithOwnerCapabilities

Defined by the classes [bibus » authoredReport](#) and [bibus » dataMovementTask](#).

saCAMAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » saCAMService](#).

scheduleTriggerName

Defined by the class [bibus » history](#).

scheduleType

Defined by the class [bibus » history](#).

scope

Defined by the class [bibus » drillPath](#).

screenTip

Defined by the class [bibus » uiClass](#).

searchPath

Defined by the class [bibus » baseClass](#).

securityMembers

Defined by the classes [bibus » group](#) and [bibus » role](#).

selectionContext

Defined by the class [bibus » annotation](#).

sequencing

Defined by the classes [bibus » authoredAgentDefinition](#) and [bibus » jobDefinition](#).

serverGroup

Defined by the classes [bibus » baseReport](#), [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » content](#), [bibus » dispatcher](#), [bibus » folder](#), [bibus » jobDefinition](#), [bibus » package](#), and [bibus » reportVersion](#).

serviceDefaultOptions

Defined by the class [bibus » configuration](#).

serviceDescription

Defined by the class [bibus » portletProducer](#).

serviceName

Defined by the classes [bibus » humanTask](#) and [bibus » webServiceTask](#).

severity

Defined by the class [bibus » baseHistoryDetail](#).

shown

Defined by the class [bibus » uiClass](#).

source

Defined by the class [bibus » SQL](#).

specification

Defined by the classes [bibus » authoredPowerPlay8Report](#), [bibus » authoredReport](#), [bibus » baseDataIntegrationTask](#), [bibus » dataMovementTask](#), [bibus » drillPath](#), [bibus » indexUpdateTask](#), [bibus » launchable](#), [bibus » migrationTask](#), [bibus » planningMacroTask](#), [bibus » planningTask](#), [bibus » portalSkin](#), [bibus » reportCache](#), [bibus » reportVersion](#), and [bibus » resource](#).

ssAuditLevel

Defined by the classes [bibus » configuration](#), [bibus » configurationFolder](#), [bibus » dispatcher](#), and [bibus » systemService](#).

startAsActive

Defined by the class [bibus » contentManagerService](#).

startDate

Defined by the class [bibus » schedule](#).

state

Defined by the class [bibus » runTimeState](#).

status

Defined by the class [bibus » history](#).

stepObject

Defined by the class [bibus » jobStepDefinition](#).

storedProcedureName

Defined by the class [bibus » storedProcedureTask](#).

storeID

Defined by the class [bibus » baseClass](#).

surname

Defined by the classes [bibus » account](#) and [bibus » contact](#).

systemMetric

Defined by the class [bibus » systemMetricThresholds](#).

target

Defined by the classes [bibus » drillPath](#) and [bibus » shortcut](#).

targetOptions

Defined by the class [bibus » drillPath](#).

targetParameters

Defined by the class [bibus » drillPath](#).

taskID

Defined by the class [bibus » schedule](#).

taskObject

Defined by the class [bibus » agentTaskDefinition](#).

temporaryObjectLifetime

Defined by the class [bibus » configuration](#).

temporaryObjectLocation

Defined by the class [bibus » configuration](#).

tenantID

Defined by the class [bibus » baseClass](#).

tenantMembers

Defined by the class [bibus » baseClass](#).

timeZoneID

Defined by the classes [bibus » account](#), [bibus » contact](#), and [bibus » schedule](#).

title

Defined by the class [bibus » baseRSSTask](#).

triggerDescription

Defined by the class [bibus » authoredReport](#).

triggerName

Defined by the classes [bibus » authoredReport](#) and [bibus » schedule](#).

type

Defined by the class [bibus » schedule](#).

unit

Defined by the class [bibus » pageDefinition](#).

unixRepositoryURI

Defined by the class [bibus » configuration](#).

unixURI

Defined by the class [bibus » aliasRoot](#).

updatedObjectCount

Defined by the class [bibus » historyDetailDeploymentSummary](#).

uri

Defined by the classes [bibus » archiveLocation](#), [bibus » humanTask](#), [bibus » URL](#), and [bibus » webServiceTask](#).

usage

Defined by the class [bibus » baseClass](#).

useAccessibilityFeatures

Defined by the class [bibus » account](#).

user

Defined by the class [bibus » history](#).

userCapabilities

Defined by the classes [bibus » content](#), [bibus » folder](#), [bibus » package](#), and [bibus » session](#).

userCapability

Defined by the classes [bibus » securedFeature](#) and [bibus » securedFunction](#).

userCapabilityPolicies

Defined by the classes [bibus » content](#), [bibus » folder](#), and [bibus » package](#).

userInterface

Defined by the class [bibus » uiProfile](#).

userInterfaces

Defined by the class [bibus » package](#).

userName

Defined by the classes [bibus » account](#) and [bibus » contact](#).

version

Defined by the class [bibus » baseClass](#).

verticalElementsRenderingLimit

Defined by the classes [bibus » account](#) and [bibus » contact](#).

viewed

Defined by the class [bibus » uiClass](#).

weeklyFriday

Defined by the class [bibus » schedule](#).

weeklyMonday

Defined by the class [bibus » schedule](#).

weeklySaturday

Defined by the class [bibus » schedule](#).

weeklySunday

Defined by the class [bibus » schedule](#).

weeklyThursday

Defined by the class [bibus » schedule](#).

weeklyTuesday

Defined by the class [bibus » schedule](#).

weeklyWednesday

Defined by the class [bibus » schedule](#).

width

Defined by the class [bibus » pageDefinition](#).

windowsRepositoryURI

Defined by the class [bibus](#) » [configuration](#).

windowsURI

Defined by the class [bibus](#) » [aliasRoot](#).

yearlyAbsoluteDay

Defined by the class [bibus](#) » [schedule](#).

yearlyAbsoluteMonth

Defined by the class [bibus](#) » [schedule](#).

yearlyRelativeDay

Defined by the class [bibus](#) » [schedule](#).

yearlyRelativeMonth

Defined by the class [bibus](#) » [schedule](#).

yearlyRelativeWeek

Defined by the class [bibus](#) » [schedule](#).

queryModeEnum

Defines the query mode identifiers.

What's new

New in Version 10.1.0 – “Query Modes” on page 1865

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/queryModeEnum#compatible>

Specifies that the query is run in compatible query mode.

<http://developer.cognos.com/ceba/constants/queryModeEnum#dynamic>

Specifies that the query is run in dynamic query mode.

queryOptionEnum

Defines the query mode options with possible values defined in the [bibus](#) » [queryModeEnum](#) enumeration set enumeration set.

What's new

New in Version 10.1.0 – “Query Modes” on page 1865

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/queryOptionEnum#mode>

Specifies the query mode.

Default: [compatible](#)

queryProcessingEnum

Lists the set of query processing options.

This enumeration set

- defines values that must be specified as string constants

Members

databaseOnly

Specifies that all processing occurs only in the database. While processing in the database may be more efficient, the query may fail if some of the resources required to process the query exist outside of the database.

limitedLocal

Specifies that limited local non-database processing is allowed.

queryTaskOptionEnum

Provides the list of query service task options.

What's new

New in Version 10.1.0 – [“Query Service Administration Task” on page 1869](#)

This enumeration set was added.

Related information:

- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

Members

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#clearCache>

Specifies whether query service cache entries should be cleared.

An administrator would typically create or schedule a cache clearing task so that it is synchronized with the refresh rate of the data or metadata in the underlying data source. This ensures that the data held in the cache does not become stale. Clearing the cache also manages memory usage by the specified cache entries, such as data sources and cubes.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: false.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#clearCacheContext>

Specifies the set of entries that the [clearCache](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All entries.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#clearROLAPCubeWorkloadLog>

Specifies whether aggregate workload log data, for selected ROLAP cubes, is cleared.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: false.

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#clearROLAPCubeWorkloadLogContext>

Specifies the set of ROLAP cubes that the [clearROLAPCubeWorkloadLog](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) receiving the request.

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#dumpCache>

Specifies the creation of an administrator report that identifies the cache state (hits and misses) of the query service cache entries specified by [dumpCacheContext](#).

This information can help administrators with scheduling other task maintenance activities.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#dumpCacheContext>

Specifies the set of entries that the [dumpCache](#) option acts upon.

To specify this option, use the [bibus](#) » [genericOptionStringArray](#) class.

Default: All entries.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#getROLAPCubeState>

Specifies whether the state of ROLAP cubes should be interrogated.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#getROLAPCubeStateContext>

Specifies the set of ROLAP cubes that the [getROLAPCubeState](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) service receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#incrementallyLoadROLAPCube>

Specifies whether ROLAP cubes should be incrementally loaded.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#incrementallyLoadROLAPCubeContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#incrementallyLoadROLAPCubeContext)**

Specifies the set of ROLAP cubes that the [incrementallyLoadROLAPCube](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) service receiving the request.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#pauseROLAPCubes](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#pauseROLAPCubes)**

Specifies whether ROLAP cubes should be paused.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#pauseROLAPCubesContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#pauseROLAPCubesContext)**

Specifies the set of ROLAP cubes that the [pauseROLAPCubes](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) service receiving the request.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeDataCache](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeDataCache)**

Specifies that the ROLAP cube data cache should be refreshed.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeDataCacheContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeDataCacheContext)**

Specifies the set of ROLAP cubes that the [refreshROLAPCubeDataCache](#) option acts upon.

To specify this option, use an option of class [bibus](#) » [genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeMemberCache](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeMemberCache)**

Specifies that the ROLAP cube member cache should be refreshed.

To specify this option, use an option of class [bibus](#) » [genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeMemberCacheContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeMemberCacheContext)**

Specifies the set of ROLAP cubes that the [refreshROLAPCubeMemberCache](#) option acts upon.

To specify this option, use an option of class [bibus » genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeSecurity](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeSecurity)**

Specifies that the ROLAP cube security should be refreshed.

To specify this option, use an option of class [bibus » genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#refreshROLAPCubeSecurityContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#refreshROLAPCubeSecurityContext)**

Specifies the set of ROLAP cubes that the [refreshROLAPCubeSecurity](#) option acts upon.

To specify this option, use an option of class [bibus » genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#restartROLAPCubes](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#restartROLAPCubes)**

Specifies whether ROLAP cubes should be restarted.

To specify this option, use an option of class [bibus » genericOptionBoolean](#) class.

Default: `false`.

Use of this option requires either the [canUseDataSourcesTool](#) or the [canUseServerAdministrationTool](#) user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

**[http://developer.cognos.com/ceba/constants/
queryTaskOptionEnum#restartROLAPCubesContext](http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#restartROLAPCubesContext)**

Specifies the set of ROLAP cubes that the [restartROLAPCubes](#) option acts upon.

To specify this option, use an option of class [bibus » genericOptionStringArray](#) class.

Default: All ROLAP cubes configured for the [queryService](#) receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#startROLAPCubes>

Specifies whether ROLAP cubes should be started.

To specify this option, use an option of class `bibus » genericOptionBoolean` class.

Default: `false`.

Use of this option requires either the `canUseDataSourcesTool` or the `canUseServerAdministrationTool` user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#startROLAPCubesAndSourceCubes>

Specifies whether the virtual ROLAP cube, and the source cubes it depends on, must be started. In contrast, using the `startROLAPCubes` option on a virtual cube alone will start the virtual cube and not its source cubes. Requests will fail if at least one source cube is not running.

To specify this option, use an option of class `bibus » genericOptionBoolean` class.

Default: `false`.

New in Version 10.2.0 – “IBM Cognos Dynamic Cubes Aggregate Advisor configuration” on page 1836

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#startROLAPCubesContext>

Specifies the set of ROLAP cubes that the `startROLAPCubes` option acts upon.

To specify this option, use an option of class `bibus » genericOptionStringArray` class.

Default: All ROLAP cubes configured for the `queryService` receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#stopROLAPCubes>

Specifies whether ROLAP cubes should be stopped. To stop cubes immediately, the `stopROLAPCubesImmediately` option must be set to `true`.

To specify this option, use an option of class `bibus » genericOptionBoolean` class.

Default: `false`.

Use of this option requires either the `canUseDataSourcesTool` or the `canUseServerAdministrationTool` user capability.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#stopROLAPCubesContext>

Specifies the set of ROLAP cubes that the `stopROLAPCubes` option acts upon.

To specify this option, use an option of class `bibus » genericOptionStringArray` class.

Default: All ROLAP cubes configured for the `queryService` receiving the request.

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This value was added.

<http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#stopROLAPCubesImmediately>

Specifies whether the ROLAP cubes should be stopped immediately.

In contrast, the [stopROLAPCubes](#) option stops ROLAP cubes only after current requests have completed.

To specify this option, use an option of class [bibus » genericOptionBoolean](#) class.

Default: `false`.

New in Version 10.1.1 – “[IBM Cognos ROLAP Administration](#)” on page 1847

This value was added.

reportSaveAsEnum

Lists the classes of objects that can be created as a result of a `saveAsrun` option for a report.

References

Used by the following properties:

- [bibus » runOptionSaveAs » objectClass](#)

Members

reportView

Specifies that the executed report, query, or custom view is saved as an instance of the [bibus » reportView](#) class, which captures all the parameters and run options specified with the `execute` action.

reportServiceQueryDrillPathOptionEnum

Provides the list of options to be used with the [drillThrough » queryDrillPath\(objectPath, parameterValues, options\)](#) method.

References

Used by the following properties:

- [bibus » reportServiceQueryDrillPathOption » name](#)

What's new

New in Version 8.4 – “[Supporting New Drill-through Targets](#)” on page 1890

This enumeration set was added.

Members

upgrade

Specifies whether the drill-through specification will be upgraded to the current version. To specify this option, use the [bibus » reportServiceQueryDrillPathOptionBoolean](#) class.

Default: `false`

reportServiceQueryOptionEnum

Provides the list of report service query options.

References

Used by the following properties:

- [bibus » reportServiceQueryOption » name](#)

Members

specificationFormat

Specifies the format of the returned report specification. To specify this option, use the [bibus » reportServiceQueryOptionSpecificationFormat](#) class. Possible choices are defined in the [bibus » specificationFormatEnum](#) enumeration set.

upgrade

Specifies whether the report specification will be upgraded to the current version. If the value specified for `specificationFormat` causes a conversion to occur, the report specification will be upgraded automatically. To specify this option, use the [bibus » reportServiceQueryOptionBoolean](#) class.

Default: false

reportStudioOptionEnum - deprecated

Provides the list of IBM Cognos Analytics - Reporting options.

References

Used by the following properties:

- [bibus » reportStudioOption » name](#)

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This enumeration set is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This enumeration set was added.

Members

profile

Specifies the [bibus » uiProfile](#) location.

Use this option to specify whether Reporting or Workspace Advanced is specified as the user's profile preference.

To specify this option, use the [bibus » reportStudioOptionSearchPathSingleObject](#) class.

rolapDataSourceStateEnum

Defines the ROLAP data source state identifiers.

References

Used by the following properties:

- [bibus » asynchDetailROLAPDataSourceState » state](#)

What's new

New in Version 10.1.1 – [“IBM Cognos ROLAP Administration” on page 1847](#)

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#disabled>
Specifies that the ROLAP data source is disabled.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#paused>
Specifies that the ROLAP data source is paused.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#pausing>
Specifies that the ROLAP data source is pausing.

New in Version 10.2.2 – New options to manage dynamic cubes

This value was added.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#running>
Specifies that the ROLAP data source is running.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#starting>
Specifies that the ROLAP data source is in the process of starting.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#stopped>
Specifies that the ROLAP data source has stopped.

<http://developer.cognos.com/ceba/constants/rolapDataSourceStateEnum#stopping>
Specifies that the ROLAP data source is in the process of stopping.

rolapOptionEnum

Provides the list of ROLAP cube method options.

What's new

New in Version 10.1.1 – “IBM Cognos ROLAP Administration” on page 1847

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/rolapOptionEnum#immediately>

Specifies whether ROLAP cubes should stop immediately.

Default: false

rssOptionEnum

Defines Rich Site Summary (RSS) options.

References

Used by the following properties:

- [bibus](#) » [rssOption](#) » [name](#)

Members

createItem

Specifies the location of the parent object of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionSearchPathSingleObject](#) class.

deleteItem

Specifies the location of the RSS item to be deleted.

To specify this option, use the [bibus](#) » [rssOptionSearchPathSingleObject](#) class.

itemDescription

Specifies the description of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionString](#) class.

itemName

Specifies the name of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionString](#) class.

itemScreenTip

Specifies the screen tip of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionString](#) class.

itemTarget

Specifies the target of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionSearchPathSingleObject](#) class.

itemURI

Specifies the Universal Resource Identifier (URI) of the RSS item to be created.

To specify this option, use the [bibus](#) » [rssOptionAnyURI](#) class.

runConditionEnum

Provides the list of conditions used to determine whether a task is run.

References**Used by the following properties:**

- [bibus](#) » [agentTaskDefinition](#) » [runCondition](#)
- [bibus](#) » [runConditionEnumProp](#) » [value](#)

What's new**New in Version 8.3 — “[Run Agent Tasks on Failure](#)” on page 1912**

This enumeration set was added.

Members**runAlways**

Specifies that the task should be run independent of the [bibus](#) » [runStatusEnum](#) of other tasks.

runOnFailure

Specifies that the task should be run if a related task has [bibus](#) » [runStatusEnum](#) » [failed](#).

runOnSuccess

Specifies that the task should be run if a related task has [bibus](#) » [runStatusEnum](#) » [succeeded](#).

runningStateEnum

Specifies the running state of an IBM Cognos Analytics service.

References

Used by the following properties:

- [bibus » adaptiveAnalyticsService » runningState](#)
- [bibus » agentService » runningState](#)
- [bibus » annotationService » runningState](#)
- [bibus » batchReportService » runningState](#)
- [bibus » contentManagerCacheService » runningState](#)
- [bibus » contentManagerService » runningState](#)
- [bibus » dataAdvisorService » runningState](#)
- [bibus » dataIntegrationService » runningState](#)
- [bibus » dataMovementService » runningState](#)
- [bibus » deliveryService » runningState](#)
- [bibus » dimensionManagementService » runningState](#)
- [bibus » dispatcher » runningState](#)
- [bibus » eventManagementService » runningState](#)
- [bibus » EVService » runningState](#)
- [bibus » graphicsService » runningState](#)
- [bibus » humanTaskService » runningState](#)
- [bibus » idVizService » runningState](#)
- [bibus » indexDataService » runningState](#)
- [bibus » indexSearchService » runningState](#)
- [bibus » indexUpdateService » runningState](#)
- [bibus » jobService » runningState](#)
- [bibus » logService » runningState](#)
- [bibus » metadataService » runningState](#)
- [bibus » metricsManagerService » runningState](#)
- [bibus » migrationService » runningState](#)
- [bibus » mobileService » runningState](#)
- [bibus » monitorService » runningState](#)
- [bibus » planningAdministrationConsoleService » runningState](#)
- [bibus » planningDataService » runningState](#)
- [bibus » planningRuntimeService » runningState](#)
- [bibus » planningTaskService » runningState](#)
- [bibus » powerPlayService » runningState](#)
- [bibus » presentationService » runningState](#)
- [bibus » queryService » runningState](#)
- [bibus » relationalMetadataService » runningState](#)
- [bibus » reportDataService » runningState](#)
- [bibus » reportService » runningState](#)
- [bibus » repositoryService » runningState](#)

- [bibus](#) » [saCAMService](#) » [runningState](#)
- [bibus](#) » [systemService](#) » [runningState](#)
- [bibus](#) » [runningStateEnumProp](#) » [value](#)

Members

disabled

Specifies that the service was disabled in the IBM Cognos Analytics environment on this server, and will not be loaded at startup. Although it remains physically present on the IBM Cognos Analytics server, the service must be explicitly enabled, and the IBM Cognos Analytics environment on this server must be restarted for it to process requests.

running

Specifies that the service is running in the IBM Cognos Analytics environment on this server. For example, in its normal operational state, the dispatcher processes all existing requests and accepts new ones.

standby

Specifies that the service is running in the IBM Cognos Analytics environment on this server, but is not processing requests. In this state, the service can only transition to the [running](#) state as a result of an automatic failover or use of the [content](#) » [activate\(searchPath\)](#) method.

suspended

Specifies that the service is running in the IBM Cognos Analytics environment on this server, but that it was suspended. In this state, the service can no longer process normal requests. However, it can accept and process other requests to provide certain services, such as dispatch, to existing in-progress requests.

runOptionEnum

Provides the list of run options.

References

Used by the following properties:

- [bibus](#) » [runOption](#) » [name](#)

Members

advancedOutput

Specifies whether advanced features are enabled for output. Advanced features include the ability to evaluate conditional subscriptions and to import saved output versions into Office Connection. To specify this option, use the [bibus](#) » [runOptionBoolean](#) class.

Default: `false`

New in Version 8.3 — “Advanced Features for Report Output” on page 1912

This value was added.

allowAnnotations

Specifies whether saved output can be annotated by report consumers.

This option can be used only when output is saved in the content store using the [bibus](#) » [runOptionEnum](#) » [saveOutput](#) value option.

To specify this option, use the [bibus](#) » [runOptionBoolean](#) class.

Default: `false`

New in Version 8.4 – “Report Output Annotations” on page 1895

This value was added.

archive

Specifies whether the output will be archived. To specify this option, use the `bibus » runOptionBoolean` class.

Default: `false`

New in Version 8.3 – “Saving Report Output to File System” on page 1901

This value was added.

burst

Specifies whether the report is to run as a burst report. To specify this option, use the `bibus » runOptionBoolean` class.

Default: `false`

burstDistribution

Breaks and executes the burst report into several chunks and executes them in parallel.

This should result in the burst outputs being produced much more efficiently.

To specify this option, use the `bibus » runOptionString` class.

Default: `Default`

The default value observes the system wide default value, which is set to `Disabled` in the `RSVP.BURST_DISTRIBUTION` advanced setting. If this is set to `Disabled`, burst reports will not be distributed and run sequentially in one process.

New in Version 10.2.1 – “New configuration options for burst distribution” on page 1832

This value was added.

burstKeyLimit

Sets the maximum number of burst keys per distributed chunk.

Depending on the data source type, it might be necessary to set this limit in order to prevent the generation of lengthy or complex SQL. If `bibus » runOptionEnum » burstDistribution` is `Disabled`, this value has no effect.

If this value is blank (""), the value of the `BDS.split.maxKeysPerChunk` advanced setting is used. The default value for `BDS.split.maxKeysPerChunk` is `1000`. Use a value of `0` for this property, or the `BDS.split.maxKeysPerChunk` setting, to specify that the burst key limit is unlimited.

To specify this option, use the `bibus » runOptionInt` class.

Default: `1000`

New in Version 10.2.1 – “New configuration options for burst distribution” on page 1832

This value was added.

burstQueryPrefetch

When enabled, this allows DQM packages (and only DQM packages) to pre-fetch the burst queries while the outputs are being rendered. The result should be that the burst outputs will be produced faster because the queries are executing in parallel with the report rendering. To specify this option, use the `bibus » runOptionString` class.

Default: Default

The default value observes the system wide default value, which is set to Disabled by default in the RSVP . BURST _ QUERY _ PREFETCH setting. If this is set to Disabled, burst queries run sequentially.

New in Version 10.2.1 – “New configuration options for burst distribution” on page 1832

This value was added.

connection

continueConversation

Specifies that the asynchronous conversation should continue after the request has executed. To specify this option, use the [bibus » runOptionBoolean](#) class. This option can be used with the [report » deliver\(conversation, parameterValues, options\)](#) method

Default: false

credentialParameters

Specifies that parameters related to data source connections should be returned by the [parameter » getParameters\(objectPath, parameterValues, options\)](#) method or the [parameter » getParametersSpecification\(specification, parameterValues, options\)](#) method. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

cssURL

Specifies the Web location of the user CSS style sheet to use when rendering the report. This style sheet is used in conjunction with the style sheet specified by the report author. In adherence to the CSS 2.1 standard from W3C, this style sheet must include the keyword !important on each declaration that is intended to override the author's style sheet behavior. For more information, see the W3C website.

To specify this option, use the [bibus » runOptionAnyURI](#) class.

This option is validated by CAF if CAF is enabled and the RSVP . RENDER . VALIDATEURL advanced setting is set to 'true':

New in Version 8.4 – “URL Validation Rules” on page 1893

CAF validation rules were added.

data

Specifies whether Report Server will retrieve data when running the report. To specify this option, use the [bibus » runOptionData](#) class.

Default: [bibus » dataEnum » runWithAllData](#) value

email

Specifies whether the output will be distributed by email. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

emailAsAttachment

Specifies whether the output will be distributed as an email attachment. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

emailAsURL

Specifies whether a link to the output will be distributed as an email attachment. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

layoutParameters

Specifies that all parameters defined in the layout section of a report specification should be returned by the [parameter » getParameters\(objectPath, parameterValues, options\)](#) method or the [parameter » getParametersSpecification\(specification, parameterValues, options\)](#) method regardless of whether the parameter is used when the report is executed. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

maximumValueCount

Specifies the maximum number of candidate prompt values in the result set. Use with [skipValueCount](#) to partition a large result set.

For example, you can use this option to select the set of candidate prompt values when scrolling in your Web portal.

To specify this option, use the [bibus » runOptionInt](#) class.

Default: 0 (Returns all values)

metadataModel

Specifies the metadata model used to execute the report. This option should only be used if the report was not saved in Content Manager. To specify this option, use the [bibus » runOptionString](#) class.

mobile

Specifies whether the output will be distributed to mobile devices. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: false

outputEncapsulation

Specifies whether the server returns output documents in the response, or caches them in the user's session object and returns a reference to the documents instead.

To use this option, the report output must be returned in the SOAP response. For this to occur, the value of the [saveAs](#) option must be unspecified and the values of the following options must be false:

- [archive](#)
- [email](#)
- [print](#)
- [saveOutput](#)

To specify this option, use the [bibus » runOption](#) class. Possible choices are defined in the [bibus » outputEncapsulationEnum](#) enumeration set .

Default: none

outputFormat

Specifies the output format. To specify this option, use the [bibus » runOptionStringArray](#) class. Possible choices are defined in the [bibus » outputFormatEnum](#) enumeration set.

Default: [HTML](#)

outputLocale

Specifies the language of the report output. The value is expressed as a hyphenated language-region pair, in accordance with the RFC3066 standard.

To specify a locale other than the default output locale, you can specify a single locale or you can specify multiple locales in conjunction with any of the following run options:

- [email](#)
- [print](#)
- [saveAs](#)
- [saveOutput](#)
- [savePromptCache](#)

To specify this option, use the [bibus » runOptionLanguageArray](#) class.

Default: an array with the single value en-us

outputLocation

Specifies the location for storing temporary objects created by [reportService](#) and [batchReportService](#).

To specify this option, use the [bibus » runOptionAnyURI](#) class and values from the [bibus » temporaryObjectLocationEnum](#) enumeration set.

Default: [contentStore](#)

New in Version 10.1.0 – “Storing Temporary Output Objects Outside the Content Store” on page 1857

This value was added.

outputPageDefinition

Specifies the [bibus » pageDefinition](#) attributes of the output report. To specify this option, use the [bibus » runOptionString](#) class.

outputPageOrientation

Specifies the page orientation of the output report. To specify this option, use the [bibus » runOptionString](#) class. Possible choices are defined in the [bibus » pageOrientationEnum](#) enumeration set.

Default: [portrait](#)

print

Specifies whether the new output will be printed. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: `false`

printer

Identifies the instance of the [bibus » printer](#) class that will be used to print the report. To specify this option, use the [bibus » runOptionString](#) class.

printerAddress

Identifies the network address of the printer that will be used to print the report. To specify this option, use the [bibus » runOptionString](#) class.

prompt

Specifies whether the report service will issue prompts, so that users can enter report option values. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: `true`

promptCacheMode

Specifies the action that the report service will perform on the prompt cache. To specify this option, use the [bibus » runOptionPromptCacheMode](#) class.

Default: [none](#)

New in Version 8.4 – “On Demand Refresh of Prompt Cache” on page 1884

This value was added. It replaces the [bibus » runOptionEnum » savePromptCache](#) value.

promptFormat

Specifies the format produced for prompt pages.

If the value of the [outputFormat](#) run option is [HTML](#), the default value is [HTML](#). Otherwise, the default value is [XHTML](#).

Use an option of class [bibus » runOptionString](#) to specify different values.

returnOutputWhenAvailable

Specifies that available output should be returned in the response ([bibus » asynchReply » details](#) property) to a [wait\(conversation, parameterValues, options\)](#) request.

By using this option, your application does not need to call the [report » getOutput\(conversation, parameterValues, options\)](#) method to retrieve report output from the [reportService](#) service. When you use this option, the [reportService](#) service never returns [responseReady](#) in the [bibus » asynchDetailReportStatus » status](#) property.

To specify this option, use the [bibus » runOptionBoolean](#).

Default: `false`

New in Version 10.1.0 – “Batch Report Service/Report Service Optimizations” on page 1870

This value was added.

saveAs

Contains the information required to create a new object in Content Manager based on the object to be executed. To specify this option, use the [bibus » runOptionSaveAs](#) class.

This option cannot be used with the [asynch » runSpecification\(specification, parameterValues, options\)](#) method .

saveOutput

Specifies whether the output should be saved to the content store. To specify this option, use the [bibus » runOptionBoolean](#) class.

This option cannot be used with the [asynch » runSpecification\(specification, parameterValues, options\)](#) method .

Default: `false`

savePromptCache

Specifies whether the prompt data should be saved in the content store. To specify this option, use the [bibus » runOptionBoolean](#) class.

Existing cache data is deleted each time the cache is updated. Prompt controls that rely on parameter values, other controls, or additional data do not generate data in the cache.

This option is used in conjunction with the [prePopulateLevels](#) option on tree prompt controls to control the amount of data stored in the cache.

Default: `false`

New in Version 8.4 – “On Demand Refresh of Prompt Cache” on page 1884

This value is deprecated. Use the [bibus » runOptionEnum » promptCacheMode](#) value instead.

secondaryWaitThreshold – deprecated

Specifies the maximum amount of time, in seconds, that the server can use to process the request before sending a response to the client. To specify this option, use the [bibus » runOptionInt](#) class.

Use a value of 0 when you want the client to wait indefinitely.

Default: 30

This value

- is deprecated and will be removed in a future version of the product

selectionBasedFeatures

Specifies whether selection-based features are enabled in report output. To specify this option, use the [bibus » runOptionBoolean](#) class.

Default: true

Disabling this option turns off package drill-through and IBM Cognos Search features.

New in Version 8.3 — “Disabling Selection-based Interactive Report Output Features” on page 1911

This value was added.

skipValueCount

Specifies the number of candidate prompt values to be skipped in the result set. Use with [maximumValueCount](#) to partition a large set of prompt values.

For example, you can use this option to select the set of candidate prompt values when scrolling in your Web portal.

To specify this option, use the [bibus » runOptionInt](#) class.

Default: 0

verticalElements

Specifies the number of vertical elements to be rendered on an infinitely sized canvas, such as an HTML page. To specify this option, use the [bibus » runOptionInt](#) class.

Use a value of 0 if you want all vertical elements to be rendered on one page.

xslParameters

Specifies values for the parameters defined in the XSL style sheet specified by [xslURL](#). To specify this option, use the [bibus » runOptionNameValueArray](#) class.

xslURL

Specifies the name of an XSL style sheet that can be applied to the report to render it in the requested format. The value of this option must match the name of a file in the `templates/rsvp/xsl` directory of the IBM Cognos Analytics installation. The value must not contain a path qualifier.

To specify this option, use the [bibus » runOptionAnyURI](#) class.

This option is valid only for the [HTML](#), [HTMLFragment](#), and [XHTML](#) output formats. It is ignored for non-HTML output formats.

The default value is `v5html.xsl` and should not be changed. Specifying a value other than the default may reduce the functionality of the output. For example, drill through may not work properly.

runStatusEnum

Specifies the values used to describe job status when a task is run.

The following diagram shows the various state transitions of an event, indicating which methods invoke a change in state:

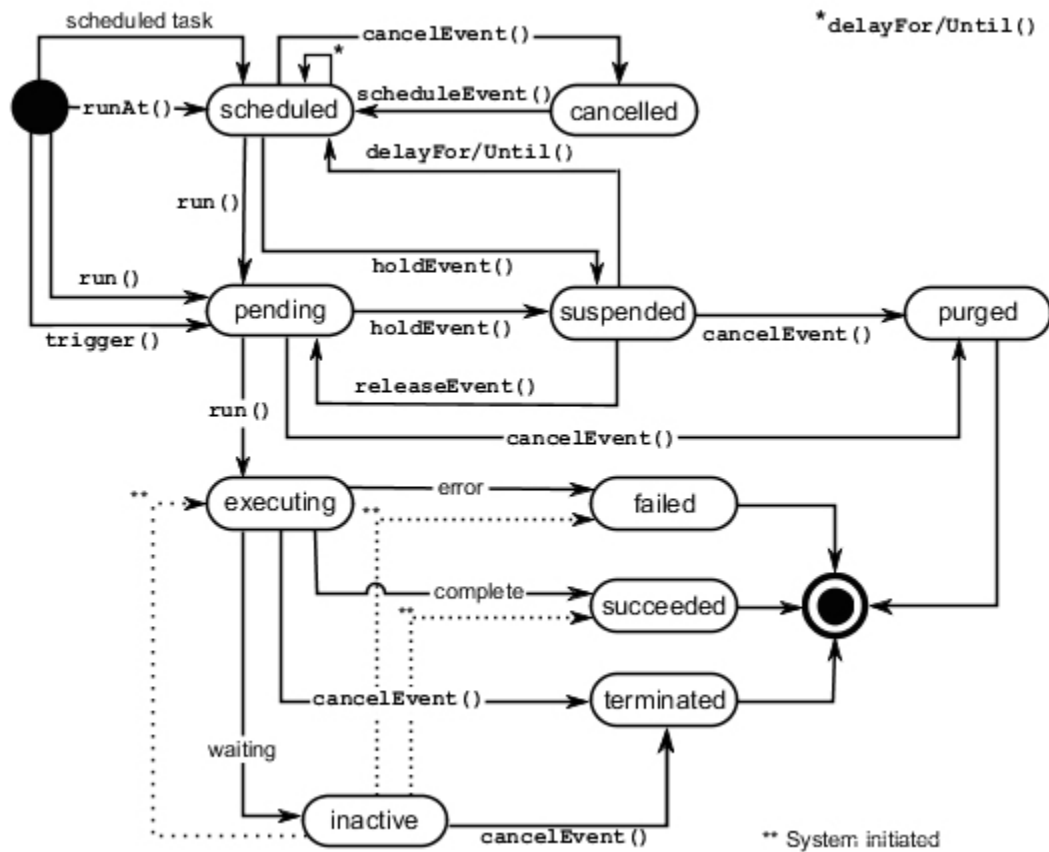


Figure 10. State transitions for an event

For more information, see:

- [event](#) method set
- [asynch](#) method set
- [“Running Jobs” on page 76](#)

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus » asynchDetailEventRecord » status](#)
- [bibus » eventRecord » status](#)
- [bibus » history » status](#)

Members

cancelled

Specifies that a scheduled task was canceled.

executing

Specifies that a task is running.

failed

Specifies that the task did not end successfully. The error log provides details about why the task was not successful.

inactive

Specifies that a task is inactive.

New in Version 8.3 — “E-Mail Activity and Administration” on page 1911

This value was added.

pending

Specifies that a task is waiting to run.

purged

Specifies that the pending task was removed from the schedule and delivery service queue.

scheduled

Specifies that the task is scheduled for execution.

succeeded

Specifies that the task ended successfully.

suspended

Specifies that the task remains on hold in the queue. The user must change the status for the task to proceed.

terminated

Specifies that the executing task was canceled.

scheduleDailyPeriodEnum

Specifies the time units that can be used when scheduling recurring tasks.

This enumeration set

- defines values that must be specified as string constants

References**Used by the following properties:**

- [bibus](#) » [schedule](#) » [dailyPeriod](#)

Members**day**

Specifies day as the unit of time.

hour

Specifies hour as the unit of time.

minute

Specifies minute as the unit of time.

scheduleEndTypeEnum

Specifies the set of options that can be used to define the last occurrence of a scheduled task.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [schedule](#) » [endType](#)

Members

indefinite

Specifies that there is no end date for the task. Therefore, the task recurs indefinitely.

onDate

Specifies that there is an end date for the recurring task. The task will stop recurring when the current time exceeds the date and time specified by the [bibus](#) » [schedule](#) » [endDate](#) property.

scheduleTypeEnum

Specifies the recurrence type for a schedule, such as daily or weekly.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [asynchDetailEventRecord](#) » [scheduleType](#)
- [bibus](#) » [history](#) » [scheduleType](#)
- [bibus](#) » [schedule](#) » [type](#)

Members

daily

Specifies that the task occurs daily.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [dailyPeriod](#) property
- [bibus](#) » [schedule](#) » [everyNPeriods](#) property

dailyWithIntradayRecurrence

Specifies that the task occurs daily, with recurrences at set intervals.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [everyNPeriods](#) property

This value is always interpreted as if [bibus](#) » [schedule](#) » [dailyPeriod](#) property were set to [bibus](#) » [scheduleDailyPeriodEnum](#) » [day](#))

- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

monthlyAbsolute

Specifies that the task occurs on a specific day each month.

When using this value, set the [bibus » schedule » monthlyAbsoluteDay](#) property.

monthlyAbsoluteWithIntradayRecurrence

Specifies that the task occurs on a specific day of each month, with recurrences at set intervals.

When using this value, set the following properties:

- [bibus » schedule » monthlyAbsoluteDay](#) property
- [bibus » schedule » intradayRecurrenceStart](#) property
- [bibus » schedule » intradayRecurrenceEnd](#) property
- [bibus » schedule » intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

monthlyRelative

Specifies that the task occurs on a relative day of each month, such as the first Monday.

When using this value, set the following properties:

- [bibus » schedule » monthlyRelativeDay](#) property
- [bibus » schedule » monthlyRelativeWeek](#) property

monthlyRelativeWithIntradayRecurrence

Specifies that the task occurs on a relative day of each month (for example, Monday), with recurrences at set intervals.

When using this value, set the following properties:

- [bibus » schedule » monthlyRelativeDay](#) property
- [bibus » schedule » monthlyRelativeWeek](#) property
- [bibus » schedule » intradayRecurrenceStart](#) property
- [bibus » schedule » intradayRecurrenceEnd](#) property
- [bibus » schedule » intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

once

Specifies that the task occurs once.

subscription

Specifies that the task occurs as a result of a subscription.

New in Version 8.3 – “Conditional Subscriptions” on page 1899

This value was added.

trigger

Specifies that the task occurs as a result of an external occurrence. Use the [event » trigger\(triggerName\)](#) method to advise IBM Cognos Analytics of the external occurrence.

weekly

Specifies that the task occurs weekly.

When using this value, set one or more of the following properties:

- [bibus](#) » [schedule](#) » [weeklySunday](#) property
- [bibus](#) » [schedule](#) » [weeklyMonday](#) property
- [bibus](#) » [schedule](#) » [weeklyTuesday](#) property
- [bibus](#) » [schedule](#) » [weeklyWednesday](#) property
- [bibus](#) » [schedule](#) » [weeklyThursday](#) property
- [bibus](#) » [schedule](#) » [weeklyFriday](#) property
- [bibus](#) » [schedule](#) » [weeklySaturday](#) property

weeklyWithIntradayRecurrence

Specifies that the task occurs one or more days a week, with recurrences at set intervals for each of the days specified.

When using this value, set one or more of the following properties:

- [bibus](#) » [schedule](#) » [weeklySunday](#) property
- [bibus](#) » [schedule](#) » [weeklyMonday](#) property
- [bibus](#) » [schedule](#) » [weeklyTuesday](#) property
- [bibus](#) » [schedule](#) » [weeklyWednesday](#) property
- [bibus](#) » [schedule](#) » [weeklyThursday](#) property
- [bibus](#) » [schedule](#) » [weeklyFriday](#) property
- [bibus](#) » [schedule](#) » [weeklySaturday](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

yearlyAbsolute

Specifies that the task occurs on a specific day of the year, such as January 1.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [yearlyAbsoluteDay](#) property
- [bibus](#) » [schedule](#) » [yearlyAbsoluteMonth](#) property

yearlyAbsoluteWithIntradayRecurrence

Specifies that the task occurs on a specific day of the year (for example, January 1st), with recurrences at set intervals.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [yearlyAbsoluteDay](#) property
- [bibus](#) » [schedule](#) » [yearlyAbsoluteMonth](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

yearlyRelative

Specifies that the task occurs on a relative day of the year, such as the first Monday in January.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [yearlyRelativeWeek](#) property
- [bibus](#) » [schedule](#) » [yearlyRelativeDay](#) property
- [bibus](#) » [schedule](#) » [yearlyRelativeMonth](#) property

yearlyRelativeWithIntradayRecurrence

Specifies that the task occurs on a relative day of the year (for example, such as the first Monday in January), with recurrences at set intervals.

When using this value, set the following properties:

- [bibus](#) » [schedule](#) » [yearlyRelativeWeek](#) property
- [bibus](#) » [schedule](#) » [yearlyRelativeDay](#) property
- [bibus](#) » [schedule](#) » [yearlyRelativeMonth](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceInterval](#) property

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

selectionContextFormatEnum

Provides the list of formats for selection contexts.

What's new

New in Version 8.4 – “Drill-Through Improvements” on page 1889

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/selectionContextFormatEnum#cognos8>

The universal selection context format as defined by IBM Cognos Analytics.

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This value was added.

<http://developer.cognos.com/ceba/constants/selectionContextFormatEnum#identity>

This URI is not a format. Use it to specify an identity transformation for the [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#) method.

<http://developer.cognos.com/ceba/constants/selectionContextFormatEnum#parameterValues>

A list of [bibus](#) » [parameterValue](#) class objects.

<http://developer.cognos.com/ceba/constants/selectionContextFormatEnum#powerPlay8>

The filter format as defined by IBM Cognos PowerPlay.

sequencingEnum

Specifies whether steps in a job can run at the same time.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [authoredAgentDefinition](#) » [sequencing](#)
- [bibus](#) » [jobDefinition](#) » [sequencing](#)

Members

parallel

Steps can run at the same time.

Use this value if the reports in a job are not dependent on one another.

When set to run at the same time, all the steps are submitted at the same time and the job is successful when all the steps run.

sequential

Steps must run one after the other.

Use this value if the reports in a job are dependent on one another.

When the steps are executed in sequence, you can specify the order in which the steps run by setting the [bibus](#) » [uiClass](#) » [displaySequence](#) property. A step is submitted only after the preceding step has run successfully.

severityEnum

Specifies the severity levels assigned to system messages issued during error and exception handling. The possible choices, in decreasing order of severity, are `fatal`, `error`, `warning`, `info`, and `debug`.

References

Used by the following properties:

- [bibus](#) » [history](#) » [maximumDetailSeverity](#)
- [bibus](#) » [baseHistoryDetail](#) » [severity](#)
- [bibus](#) » [CAMException](#) » [severity](#)
- [bibus](#) » [severityEnumProp](#) » [value](#)

Members

debug

Specifies that error messages may be at any severity level, from `debug` to `fatal`.

error

Specifies that error messages are at a severity level of `error` or `fatal`.

fatal

Specifies that error messages are at a severity level of `fatal`.

info

Specifies that error messages are at a severity level of `info`, `warn`, `error`, or `fatal`.

warn

Specifies that error messages are at a severity level of `warn`, `error`, or `fatal`.

smtpContentDispositionEnum

Provides the list of SMTP content disposition types.

References

Used by the following properties:

- [bibus](#) » [memoPart](#) » [contentDisposition](#)

Members

attachment

Specifies that the [bibus](#) » [memoPart](#) object should be delivered as an attachment.

inline

Specifies that the [bibus](#) » [memoPart](#) object should be delivered inline.

smtpContentTypeEnum

Provides the list of SMTP content types.

References

Used by the following properties:

- [bibus](#) » [memoPartComposite](#) » [contentType](#)

Members

alternative

Specifies that all contained parts are alternatives. Each part includes the same content in a different format. Each recipient sees only one of the alternatives, based on email package settings and capabilities. For example, a message may include both HTML and plain text versions. If the recipient's email package cannot display messages in HTML format, the plain text version will be used.

mixed

Specifies that the contained parts are simply co-located. There are no explicit relationships between any of the parts.

related

Specifies that there are explicit relationships between the contained parts. For example, one part may reference the graphic in another part.

specificationFormatEnum

Provides the list of report specification formats.

References

Used by the following properties:

- [bibus](#) » [reportServiceQueryOptionSpecificationFormat](#) » [value](#)

Members

analysis

Specifies the format used by IBM Cognos Analysis Studio to create instances of the [bibus » analysis](#) class.

interactiveReport

Specifies the format used by IBM Cognos Analytics - Reporting to create instances of the [bibus » interactiveReport](#) class.

New in Version 10.1.0 – “IBM Cognos Active Report” on page 1869

This value was added.

query

Specifies the format used by IBM Cognos Query Studio to create instances of the [bibus » query](#) class.

report

Specifies the format used by IBM Cognos Analytics - Reporting to create instances of the [bibus » report](#) class.

reportTemplate

Specifies the format used by IBM Cognos Analytics - Reporting to create instances of the [bibus » reportTemplate](#) class.

specificationOptionEnum

Defines the specification options.

References

Used by the following properties:

- [bibus » specificationOption » name](#)

What's new

New in Version 8.4 – “Dynamic Filtering of Report Data” on page 1889

This enumeration set was added.

Members

editSpecification

Specifies the set of actions, such as additional filtering, to be applied to the specification before it is run.

To specify this option, use the [bibus » specificationOptionXMLEncodedXML](#) class.

useStyleVersion

Specifies the version of the styles the [reportService](#) service and [batchReportService](#) service should use to render the report. To specify this option, use the [bibus » specificationOptionString](#) class.

Default: Use the style that the `useStyleVersion` attribute of the `report` element specifies in the report specification. For example, `GlobalReportStyles_none.css`.

New in Version 10.1.0 – “Batch Report Service/Report Service Optimizations” on page 1870

This value was added.

subscriptionOptionEnum

Provides the list of subscription options.

References

Used by the following properties:

- [bibus](#) » [subscriptionOption](#) » [name](#)

What's new

New in Version 8.3 — [“Conditional Subscriptions” on page 1899](#)

This enumeration set was added.

Members

documentVersion

Specifies the document version containing the output that should be used to process the subscription. To specify this option, use the [bibus](#) » [subscriptionOptionSearchPathSingleObject](#) class. This option is generated and maintained by the [eventManagementService](#) service. SDK applications should not use options of this type.

systemMetricEnum

Defines the system metrics.

References

Used by the following properties:

- [bibus](#) » [systemMetricThresholds](#) » [systemMetric](#)

What's new

New in Version 8.3 — [“System Metrics” on page 1908](#)

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/systemMetricEnum#averageTimeInQueue>

Specifies the average length of time, in milliseconds, that a request has spent in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#failedExternalizedDocumentsPercent>

Specifies the percentage of failed requests to save objects from the content store to an external repository.

[New in Version 10.1.0](#) — [“IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866](#)

This value was added.

[New in Version 10.1.1](#) — [“Support for IBM Cognos Content Archival” on page 1845](#)

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#failedRequestPercent>

Specifies the percentage of requests that have failed.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#heapSize>

Specifies the current size of the JVM heap (in bytes).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#heapSizeHighWaterMark>

Specifies the maximum amount of memory allocated to the JVM heap (in bytes).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#heapSizeLowerBound>

Specifies the minimum allowable size that has been configured for the JVM heap (in bytes).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#heapSizeLowWaterMark>

Specifies the minimum amount of memory allocated to the JVM heap (in bytes).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#heapSizeUpperBound>

Specifies the maximum allowable size that has been configured for the JVM heap (in bytes).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#lastResponseTime>

Specifies the time taken for the most recent request.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#millisecondsPerSuccessfulRequest>

Specifies the average number of milliseconds required to process a successful request.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfFailedExternalizedDocuments>

Specifies the number of failed requests to save content store objects in an external repository.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfFailedRequests>

Specifies the number of requests that resulted in an exception (or fault).

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfProcessedRequests>

Specifies the number of requests processed.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfProcesses>

Specifies the number of server processes currently running for a particular service. For example, for the report service, specifies the number of report server processes running.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfProcessesHighWaterMark>

Specifies the maximum number of server processes for a particular service. For example, for the report service, specifies the number of report server processes.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfProcessesLowWaterMark>

Specifies the minimum number of server processes for a particular service. For example, for the report service, specifies the number of report server processes.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfRequests>

Specifies the number of requests that have passed through the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfSessions>

Specifies the number of currently active user sessions.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfSessionsHighWaterMark>

Specifies the maximum number of active user sessions.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfSessionsLowWaterMark>

Specifies the minimum number of active user sessions.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfSuccessfulExternalizedDocuments>

Specifies the number of requests successfully processed to save content store objects in an external repository.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#numberOfSuccessfulRequests>

Specifies the number of requests processed successfully.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#operationalStatus>

Specifies whether the server is available, unavailable, or partially available.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#queueLength>

Specifies the number of items currently in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#queueLengthHighWaterMark>

Specifies the maximum number of items in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#queueLengthLowWaterMark>

Specifies the minimum number of items in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#requestsPerSecond>

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#requestsPerSecondHighWaterMark>

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#requestsPerSecondLowWaterMark>

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#responseTime>

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#responseTimeHighWaterMark>

Specifies the maximum time taken for a request.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#responseTimeLowWaterMark>

Specifies the minimum time taken for a request.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#serviceTime>

Specifies the total time taken to process all requests.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#serviceTimeAllRequests>

Specifies the time taken to process all requests in milliseconds.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#serviceTimeFailedRequests>

Specifies the total time taken to process all failed requests.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#serviceTimeSuccessfulRequests>

Specifies the total time taken to process all successful requests.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#successfulExternalizedDocumentsPercent>

Specifies the percentage of requests successfully processed to save content store objects in an external repository.

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#successfulRequestPercent>

Specifies the percentage of requests that have succeeded.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#successfulRequestsPerMinute>

Specifies the average number of requests that have been processed in one minute.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#timeInQueue>

Specifies the total, combined amount of time that requests have been in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#timeInQueueHighWaterMark>

Specifies the maximum time that a request has been in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#timeInQueueLowWaterMark>

Specifies the minimum time that a request has been in the queue.

<http://developer.cognos.com/ceba/constants/systemMetricEnum#upTime>

Specifies the amount of time the JVM has been running.

systemMetricThresholdsPropertyEnum

Defines the system metric thresholds properties.

References

Used by the following properties:

- [bibus](#) » [uriValue](#) » [uri](#)

What's new

New in Version 8.3 — “System Metrics” on page 1908

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#lowerSideAverageThreshold>

Specifies the lower side average range threshold value. A metric \leq this value is considered to be in the lower side average range unless it is in the lower side poor range.

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#lowerSideAverageThresholdExclusive>

Specifies whether a value matching the [lowerSideAverageThreshold](#) value is considered to be part of the good range.

Default: false

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#lowerSidePoorThreshold>

Specifies the lower side poor range threshold value. A metric value \leq this value is considered to be in the lower side poor range.

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#lowerSidePoorThresholdExclusive>

Specifies whether a value matching the [lowerSidePoorThreshold](#) value is considered to be part of the lower side average range.

Default: false

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#upperSideAverageThreshold>

Specifies the upper side average range threshold value. A metric value \geq this value is considered to be in the upper side average range unless it is in the upper side poor range.

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#upperSideAverageThresholdExclusive>

Specifies whether a value matching the [upperSideAverageThreshold](#) value is considered to be part of the good range.

Default: false

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#upperSidePoorThreshold>

Specifies the upper side poor range threshold value. A metric value \geq this value is considered to be in the upper side poor range.

<http://developer.cognos.com/ceba/constants/systemMetricThresholdsPropertyEnum#upperSidePoorThresholdExclusive>

Specifies whether a value matching the [upperSidePoorThreshold](#) value is considered to be part of the upper side average range.

Default: false

systemOptionEnum

Provides a list of options that define system-wide options.

What's new

New in Version 10.1.0 – [“Accessibility” on page 1859](#)

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/systemOptionEnum#accessibilityFeatures>

Specifies whether accessibility features are included in report output.

This option can also be used for conditional layouts in IBM Cognos Analytics - Reporting. When the option is set to `true`, report authors can specify that a crosstab is rendered as an accessible alternate to a graph or a chart.

To specify this option, use the [bibus » genericOptionBoolean](#) class.

Default: `false`

New in Version 10.1.0 – [“Accessibility” on page 1859](#)

This value was added.

temporaryObjectLocationEnum

Specifies the storage location of temporary objects created by IBM Cognos services.

References

Used by the following properties:

- [bibus » configuration » temporaryObjectLocation](#)

What's new

New in Version 10.1.0 – [“Storing Temporary Output Objects Outside the Content Store” on page 1857](#)

This enumeration set was added.

Members

<http://developer.cognos.com/ceba/constants/temporaryObjectLocationEnum#contentStore>

Specifies that temporary objects are stored in the content store.

<http://developer.cognos.com/ceba/constants/temporaryObjectLocationEnum#serverFileSystem>

Specifies that temporary objects are stored in the configured directory on the file server.

uiComponentEnum

Provides the list of IBM Cognos Analytics user interface components that consume [bibus » package](#) objects.

References

Used by the following properties:

- [bibus » uiProfile » userInterface](#)
- [bibus » package » userInterfaces](#)
- [bibus » uiComponentEnumArrayProp » value](#)
- [bibus » uiComponentEnumProp » value](#)

Members

adaptiveAnalytics

Adaptive Analytics

New in Version 8.4 – “Adaptive Analytics Integration” on page 1893

This value was added.

analysisStudio

IBM Cognos Analysis Studio

applicationStudio

Application Studio

controllerStudio

IBM Cognos Controller Studio

eventStudio

IBM Cognos Event Studio

EVStudio

New in Version 9.0.0 – “Support for IBM Cognos Express” on page 1879

This value was added.

metricStudio

IBM Cognos Metric Studio

planningContributor

IBM Cognos Planning - Contributor

powerPlayStudio

PowerPlay Studio

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This value was added.

queryStudio

IBM Cognos Query Studio

reportStudio

IBM Cognos Analytics - Reporting

tm1Studio

IBM Cognos TM1 Studio

New in Version 10.1.0 – “IBM Cognos TM1 Packages” on page 1858

This value was added.

updateActionEnum

Defines how conflicts are resolved when a Content Manager request is processed.

References

Used by the following properties:

- [bibus](#) » [addOptions](#) » [updateAction](#)

- [bibus » copyOptions » updateAction](#)
- [bibus » moveOptions » updateAction](#)

Members

fail

Specifies that the request should fail if a naming conflict exists.

replace

Specifies that if an object already exists, the existing object and any descendants are deleted before the request is processed.

Adding, copying, or moving objects with this option deletes the original source object and creates a new one when there is a naming conflict, for example, if an object is copied to the same location using the [content » copy\(objects, targetPath, options\)](#) method instead of the [content » copyRename\(objects, targetPath, newNames, options\)](#) method. By default, any ID-based references to the object are cleared. You can set the [bibus » addOptions » faultIfObjectReferenced](#), [bibus » copyOptions » faultIfObjectReferenced](#), [bibus » deleteOptions » faultIfObjectReferenced](#), and [bibus » moveOptions » faultIfObjectReferenced](#) to alter the default behavior, depending on the operation.

A [bibus » shortcut](#) refers to an object based on a search path, not an ID, and will still function if the target object still exists.

When replacing connections to databases, any database metadata that was previously imported into Framework Manager packages is not impacted. However, the action impacts existing Data Modules that referenced the connection. Any imported schemata is deleted and will not be re-assembled automatically when the new object is created.

update

Specifies that if an object already exists, the existing ID of the object is maintained and references to the object remain intact.

usageEnum

Specifies how a portal can represent the object.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus » deploymentDetail » deployedObjectUsage](#)
- [bibus » baseClass » usage](#)

Members

folder

Specifies that a portal can represent the object as a structural element, much like a non-leaf node in a tree structure. For example, a folder is a structural element that is used to organize other objects, such as reports.

object

Specifies that the portal can represent the object as an element of a structural element, much like a leaf node in a tree structure. The object can be manipulated independent of the structural element. For example, a report is an element contained within a folder. Each report can be manipulated individually.

subObject

Specifies that the portal can represent the object as an aspect of its containing object. This `subObject` may be accessed by opening a property sheet or launching an editor.

userCapabilityEnum

Defines the user capabilities that control access to an IBM Cognos Analytics functional area or tool, such as IBM Cognos Analytics Reporting.

When you run an object, the `bibus » account` used by the capabilities to determine access to secured functions and features of IBM Cognos Analytics depends on the value of the following properties:

- `bibus » baseAgentDefinition » runAsOwner` property
- `bibus » baseDataIntegrationTask » runAsOwner` property
- `bibus » baseDataMovementTask » runAsOwner` property
- `bibus » baseReport » runAsOwner` property
- `bibus » authoredReport » runWithOwnerCapabilities` property

The following sections describe additional details related to user capabilities. These sections correspond to the columns in the User Capabilities Details table.

Related Objects

Each capability is related to an object that initially controls it. The referenced objects are typically created during content store initialization, however IBM Cognos Analytics creates some objects when required.

Capability Cookie Bit Index

Each bit in the capability cookie value maps to a specific capability. Bits are ordered from right to left, starting with bit 0.

A user has a global capability if the corresponding bit in the cookie value is 1.

A user does not have a global capability if the corresponding bit in the cookie value is 0.

For example, in the value 11001010, bits 0, 2, 4, and 5 have the value 0, and the bits 1, 3, 6, and 7 have the value 1.

Undefined bits in the capability cookie value are reserved and should not be used.

For more information, see [“Querying Capabilities Using the Capability Cookie Bit Index”](#) on page 52.

Security Context Used to Determine Access

- When `runAsOwner` and `runWithOwnerCapabilities` are `false`, the capabilities of the session account determine access.
- When `runAsOwner` or `runWithOwnerCapabilities` is `true`, the User Capabilities Details table specifies the account that the capabilities use to determine access.

User Capability Details

The following table provides specific details for each user capability.

Table 227. User capability details

Capability	Related Object	Cookie Bit Index	Account (either runAsOwner or runWithOwnerCapabilities is true)
canCollaborate	“Collaborate” on page 1635	92	Session
canImportRelationalMetadata	“Import relational metadata” on page 1649	88	Session
canGenerateCSVOutput	“Generate CSV Output” on page 1645	50	
canGeneratePDFOutput	“Generate PDF Output” on page 1646	51	
canGenerateXLSOutput	“Generate XLS Output” on page 1647	52	
canGenerateXMLOutput	“Generate XML Output” on page 1647	53	
canLaunchCollaborationTools	“Launch collaboration tools” on page 1636	93	Session
canOpenPowerPlayInAnalysisStudio	“Open PowerPlay Reports with Analysis Studio” on page 1630	2	Session
canOpenPowerPlayInReportStudio	“Open PowerPlay Reports with Reporting” on page 1660	36	Session
canReceiveDetailedErrors	“Detailed Errors” on page 1637	10	Session
canUpdateRepositoryRules	“Manage repository connections” on page 1644	95	Session
canUseAdaptiveAnalytics	“Adaptive Analytics” on page 1616	0	Session
canUseAdaptiveAnalyticsAdministration	“Adaptive Analytics Administration” on page 1618	65	Session
canUseAdministrationPortal	“Administration” on page 1617	64	Session
canUseAdvancedDashboardFeatures	“Use Advanced Dashboard Features” on page 1641	98	Session
canUseAnalysisStudio	“Analysis Studio” on page 1629	1	Session
canUseBursting	“Bursting” on page 1658	33	Owner
canUseCapabilitiesTool	“Set capabilities and manage UI profiles” on page 1627	76	Session

Table 227. User capability details (continued)

Capability	Related Object	Cookie Bit Index	Account (either runAsOwner or runWithOwnerCapabilities is true)
canUseCognosInsight	“Cognos Insight” on page 1631	61	Session
canUseCognosViewer	“Cognos Viewer” on page 1631	3	Session
canUseCognosViewerContextMenu	“Context Menu” on page 1632	4	Session
canUseCognosViewerRunWithOptions	“Run With Options” on page 1633	5	Session
canUseCognosViewerSelection	“Selection” on page 1634	6	Session
canUseCognosViewerToolbar	“Toolbar” on page 1634	7	Session
canUseCollaborationFeatures	“Allow collaboration features” on page 1635	94	Session
canUseConditionalSubscriptions	“Watch Rules” on page 1669	31	Session
canUseContentStoreTool	“Administration tasks” on page 1618	66	Session
canUseControllerAdministration	“Controller Administration” on page 1620	68	Session
canUseControllerStudio	“Controller Studio” on page 1636	8	Session
canUseDashboardViewer	“Executive Dashboard” on page 1640	14	Session
canUseDashboardViewerFileManagement	“Use the Edit Features” on page 1643	15	Session
canUseDataManager	“Data Manager” on page 1637	9	Session
canUseDataSourcesTool	“Data Source Connections” on page 1620	69	Session
canUseDescriptiveStatistics	“Statistics” on page 1668	38	Session
canUseDistributionListsAndContactsTool	“Distribution Lists and Contacts” on page 1621	70	Session
canUseDrillThroughAssistant	“Drill Through Assistant” on page 1638	11	Session
canUseEV	“EVStudio” on page 1639	86	Session
canUseEventStudio	“Event Studio” on page 1639	12	Session
canUseExternalData	“Allow External Data” on page 1657	90	Session
canUseGlossary	“Glossary” on page 1648	16	Session

Table 227. User capability details (continued)

Capability	Related Object	Cookie Bit Index	Account (either <code>runAsOwner</code> or <code>runWithOwnerCapabilities</code> is true)
canUseHTML	“HTML Items in Report” on page 1659	35	Owner
canUseIndexSearch	“Execute Indexed Search” on page 1640	13	Session
canUseInteractiveDashboardFeatures	“Use Interactive Dashboard Features” on page 1642	99	Session
canUseLineage	“Lineage” on page 1650	18	Session
canUseMetricsManagerAdministration	“Metric Studio Administration” on page 1622	71	Session
canUseMetricStudio	“Metric Studio” on page 1651	19	Session
canUseMetricStudioEditView	“Edit View” on page 1651	20	Session
canUseMobileService	“Mobile” on page 1652	62	Session
canUseMonitorActivityTool	“Run activities and schedules” on page 1626	75	Session
canUseMyDataSets	“My Data Sets” on page 1653	54	
canUseMyDataSetsAdministration	“My Data Sets Administration” on page 1623	55	
canUseObjectCapabilities	“Set Entry-Specific Capabilities” on page 1667	29	Session
canUsePackageDataSources	“Package Data Sources” on page 1653	21	Session
canUsePersonalDataSourceCredentials	“Manage own data source signons” on page 1650	87	Session
canUsePlanningAdministration	“Planning Administration” on page 1624	72	Session
canUsePlanningContributor	“Planning Contributor” on page 1654	22	Session
canUsePortalAdministrationTool	“Styles and portlets” on page 1628	77	Session
canUsePowerPlay	“PowerPlay Studio” on page 1655	23	Session
canUsePowerPlayAdministration	“PowerPlay Servers” on page 1625	73	Session
canUsePrintersTool	“Printers” on page 1625	74	Session

Table 227. User capability details (continued)

Capability	Related Object	Cookie Bit Index	Account (either runAsOwner or runWithOwnerCapabilities is true)
canUseQueryServiceTool	“Query Service Administration” on page 1626	63	Session
canUseQueryStudio	“Query Studio” on page 1655	24	Session
canUseQueryStudioAdvancedMode	“Advanced” on page 1656	25	Session
canUseQueryStudioFileManagement	“Create” on page 1656	26	Session
canUseReportStudio	“Reporting” on page 1657	32	Session
canUseReportStudioFileManagement	“Create/Delete” on page 1659	34	Session
canUseRepository	“External Repositories” on page 1644	96	Session
canUseScheduling	“Scheduling” on page 1661	27	Session
canUseSchedulingByDay	“Schedule by day” on page 1662	79	Session
canUseSchedulingByHour	“Schedule by hour” on page 1663	80	Session
canUseSchedulingByMinute	“Schedule by minute” on page 1663	81	Session
canUseSchedulingByMonth	“Schedule by month” on page 1664	82	Session
canUseSchedulingByTrigger	“Schedule by trigger” on page 1664	83	Session
canUseSchedulingByWeek	“Schedule by week” on page 1665	84	Session
canUseSchedulingByYear	“Schedule by year” on page 1665	85	Session
canUseSchedulingPriority	“Scheduling Priority” on page 1665	28	Session
canUseSelfServicePackageWizard	“Self Service Package Wizard” on page 1667	91	Session
canUseServerAdministrationTool	“Configure and manage the system” on page 1619	67	Session
canUseShowHiddenObjectsPreference	“Hide Entries” on page 1648	17	Session
canUseSpecifications	“Specification Execution” on page 1668	30	Session
canUseUserDefinedSQL	“User Defined SQL” on page 1661	37	Owner

Table 227. User capability details (continued)

Capability	Related Object	Cookie Bit Index	Account (either runAsOwner or runWithOwnerCapabilities is true)
canUseUsersGroupsAndRolesTool	“Users, Groups, and Roles” on page 1629	78	Session
canViewContentInRepository	“View external documents” on page 1645	97	Session

11.1.2

The new `userCapabilitiesEx` cookie includes all the encoded capabilities found in `userCapabilities` cookie. The `userCapabilitiesEx` cookie, however, extends cookie bit index support above 127. Cognos Analytics sets both cookies at the same time. The following table provides specific details for user capabilities with a bit index of 128 and higher.

Table 228. User capability details

Capability	Related Object	Cookie Bit Index	Account (either runAsOwner or runWithOwnerCapabilities is true)
canEditJupyterNotebook	“Collaborate” on page 1635	128	Session
canUseJob	“Import relational metadata” on page 1649	129	Session

References

Used by the following properties:

- [bibus » content » effectiveUserCapabilities](#)
- [bibus » folder » effectiveUserCapabilities](#)
- [bibus » package » effectiveUserCapabilities](#)
- [bibus » content » userCapabilities](#)
- [bibus » folder » userCapabilities](#)
- [bibus » package » userCapabilities](#)
- [bibus » session » userCapabilities](#)
- [bibus » userCapabilityCache » userCapabilities](#)
- [bibus » securedFeature » userCapability](#)
- [bibus » securedFunction » userCapability](#)
- [bibus » userCapabilityPermission » userCapability](#)
- [bibus » userCapabilityEnumArrayProp » value](#)
- [bibus » userCapabilityEnumProp » value](#)

What's new

New in Version 8.4 — [“Object Capabilities” on page 1885](#)

Documentation for user capabilities has been updated to reflect whether a user capability can be set at global scope only, or at both global and object scope.

New in Version 8.4 GA – “Capability Cookie Format Changed” on page 1881

Documentation has been updated to indicate each capability's assigned capability cookie bit index.

New in Version 10.1.0 – “Run with Owner Capabilities” on page 1871

Reserved.

New in Version 10.1.0 – “Run with Owner Capabilities” on page 1871

The User Capability Details table has been updated to include context information for the [bibus](#) » [authoredReport](#) » [runWithOwnerCapabilities](#) property.

Related information

- *IBM Cognos IBM Cognos Analytics Administration and Security Guide*

Members

canCollaborate

Specifies that the user can use collaboration tools.

This capability acts as a container only and is not enforced by IBM Cognos Analytics software.

This value

- defines a capability that is initially controlled by the [“Collaborate” on page 1635](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 92 in the capability cookie value

New in Version 10.1.0 – “Collaboration Tool Integration” on page 1867

This value was added.

canGenerateCSVOutput

Specifies that the user can use generate CSV output.

This capability acts as a container only and is not enforced by IBM Cognos Analytics software.

This value

- defines a capability that is initially controlled by [“Generate CSV Output” on page 1645](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 50 in the capability cookie value

New in Version 10.2.2 – Report output format restriction

This value was added.

canGeneratePDFOutput

Specifies that the user can use generate PDF output.

This capability acts as a container only and is not enforced by IBM Cognos Analytics software.

This value

- defines a capability that is initially controlled by [“Generate PDF Output” on page 1646](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 51 in the capability cookie value

New in Version 10.2.2 – Report output format restriction

This value was added.

canGenerateXLSOutput

Specifies that the user can use generate XLS output.

This capability acts as a container only and is not enforced by IBM Cognos Analytics software.

This value

- defines a capability that is initially controlled by [“Generate XLS Output” on page 1647](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 52 in the capability cookie value

New in Version 10.2.2 – Report output format restriction

This value was added.

canGenerateXMLOutput

Specifies that the user can use generate XML output.

This capability acts as a container only and is not enforced by IBM Cognos Analytics software.

This value

- defines a capability that is initially controlled by [“Generate XML Output” on page 1647](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 53 in the capability cookie value

New in Version 10.2.2 – Report output format restriction

This value was added.

canImportRelationalMetadata

Specifies that the user can import relational metadata from data sources.

This value

- defines a capability that is initially controlled by the [“Import relational metadata” on page 1649](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 88 in the capability cookie value

New in Version 10.1.0 – “Relational Metadata Service” on page 1858

This value was added.

canLaunchCollaborationTools

Specifies that the user can launch collaboration tools.

This value

- defines a capability that is initially controlled by the [“Launch collaboration tools” on page 1636](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 93 in the capability cookie value

New in Version 10.1.0 – “Collaboration Tool Integration” on page 1867

This value was added.

canOpenPowerPlayInAnalysisStudio – deprecated

Specifies that the user can open PowerPlay 8 objects in Analysis Studio.

This value

- is deprecated and will be removed in a future version of the product
- defines a capability that is initially controlled by the [“Open PowerPlay Reports with Analysis Studio” on page 1630](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 2 in the capability cookie value

New in Version 8.4 – “Migration Capabilities” on page 1891

This value was added.

canOpenPowerPlayInReportStudio – deprecated

Specifies that the user can open PowerPlay 8 objects in Reporting.

This value

- is deprecated and will be removed in a future version of the product
- defines a capability that is initially controlled by the [“Open PowerPlay Reports with Reporting” on page 1660](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 36 in the capability cookie value

New in Version 8.4 – “Migration Capabilities” on page 1891

This value was added.

canReceiveDetailedErrors

Specifies that the user can receive detailed errors. Detailed errors can expose sensitive information such as the names of servers and dispatchers.

This value

- defines a capability that is initially controlled by the [“Detailed Errors” on page 1637](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 10 in the capability cookie value

canUpdateRepositoryRules

Specifies that the user can edit properties of objects based on the [bibus » repositoryRule](#) class.

This value

- defines a capability that is initially controlled by the [“Manage repository connections” on page 1644](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 95 in the capability cookie value

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

canUseAdaptiveAnalytics

Specifies that the user can consume Adaptive Analytics interactive reports.

This value

- defines a capability that is initially controlled by the [“Adaptive Analytics” on page 1616](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 0 in the capability cookie value

New in Version 8.4 – “Adaptive Analytics Integration” on page 1893

This value was added.

canUseAdaptiveAnalyticsAdministration

Specifies that the user can use Adaptive Analytics administrative functions to design interactive reports.

This value

- defines a capability that is initially controlled by the [“Adaptive Analytics Administration” on page 1618](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 65 in the capability cookie value

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

This value was added.

canUseAdministrationPortal

Specifies that the user can access the IBM Cognos Analytics administration portal.

A user with this capability can cancel requests not owned by the user.

This value

- defines a capability that is initially controlled by the [“Administration” on page 1617](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 64 in the capability cookie value

canUseAdvancedDashboardFeatures

Specifies that the user can use advanced dashboard features, such as adding content.

This value

- defines a capability that is initially controlled by the [“Use Advanced Dashboard Features” on page 1641](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 98 in the capability cookie value

New in Version 10.2.0 – [“Graduated dashboard capabilities” on page 1835](#)

This value was added.

canUseAnalysisStudio

Specifies that the user can use IBM Cognos Analysis Studio.

This value

- defines a capability that is initially controlled by the [“Analysis Studio” on page 1629](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 1 in the capability cookie value

canUseBursting

Specifies that the user can edit burst specifications for a report, and limits the ability to execute a report with the [burst](#) run option set to `true`. For the burst run option to have any effect, there must be a burst specification in the report specification.

This capability is based on the report owner when `runAsOwner` is enabled.

This value

- defines a capability that is assigned using the owner's account when the object's `runAsOwner` property or `runWithOwnerCapabilities` property is set to `true`
- defines a capability that is initially controlled by the [“Bursting” on page 1658](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 33 in the capability cookie value

canUseCapabilitiesTool

Specifies that the user can use the Capabilities tool. Use this tool to view and set access permissions to secured functions and features.

This value

- defines a capability that is initially controlled by the [“Set capabilities and manage UI profiles” on page 1627](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 76 in the capability cookie value

canUseCognosInsight

Specifies that the user can use Cognos Insight features in IBM Cognos Administration.

This value

- defines a capability that is initially controlled by the [“Cognos Insight” on page 1631](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 61 in the capability cookie value

New in Version 10.2.0 – [“Restricting access to Cognos Insight in IBM Cognos Analytics” on page 1841](#)

This value was added.

canUseCognosViewer

Provides a container capability for secured features related to IBM Cognos Viewer.

This capability acts as a container only and is not enforced by IBM Cognos software.

This value

- defines a capability that is initially controlled by the [“Cognos Viewer” on page 1631](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 3 in the capability cookie value

canUseCognosViewerContextMenu

Gives the user access to the Context menu in IBM Cognos Viewer.

This value

- defines a capability that is initially controlled by the [“Context Menu” on page 1632](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 4 in the capability cookie value

canUseCognosViewerRunWithOptions

Specifies that the user can access the Run with Options page in IBM Cognos Viewer.

This value

- defines a capability that is initially controlled by the [“Run With Options” on page 1633](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 5 in the capability cookie value

canUseCognosViewerSelection

Specifies that the user can select report data in IBM Cognos Viewer.

This value

- defines a capability that is initially controlled by the [“Selection” on page 1634](#)
- defines a capability that has global scope

- defines a capability that is represented by bit 6 in the capability cookie value

canUseCognosViewerToolbar

Specifies that the user can access the toolbar in IBM Cognos Viewer.

This value

- defines a capability that is initially controlled by the [“Toolbar” on page 1634](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 7 in the capability cookie value

canUseCollaborationFeatures

Specifies that the user can use collaboration features.

This value

- defines a capability that is initially controlled by the [“Allow collaboration features” on page 1635](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 94 in the capability cookie value

New in Version 10.1.0 – [“Collaboration Tool Integration” on page 1867](#)

This value was added.

canUseConditionalSubscriptions

Specifies that the user can use the conditional subscription features.

This value

- defines a capability that is initially controlled by the [“Watch Rules” on page 1669](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 31 in the capability cookie value

New in Version 8.3 – [“Conditional Subscriptions” on page 1899](#)

This value was added.

canUseContentStoreTool

Specifies that the user can use the content store tool. Use this tool to deploy data from one environment to another and to monitor the consistency of data.

This value

- defines a capability that is initially controlled by the [“Administration tasks” on page 1618](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 66 in the capability cookie value

canUseControllerAdministration

Specifies that the user can use the IBM Cognos Controller administration functions.

This value

- defines a capability that is initially controlled by the [“Controller Administration” on page 1620](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 68 in the capability cookie value

canUseControllerStudio

Specifies that the user can use IBM Cognos Controller Studio Controller Studio.

This value

- defines a capability that is initially controlled by the [“Controller Studio” on page 1636](#)

- defines a capability that has global scope
- defines a capability that is represented by bit 8 in the capability cookie value

canUseDashboardViewer

Specifies that the user can consume dashboard content in IBM Cognos Workspace.

This value

- defines a capability that is initially controlled by the [“Executive Dashboard” on page 1640](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 14 in the capability cookie value

New in Version 8.4 – [“Dashboards” on page 1894](#)

This value was added.

canUseDashboardViewerFileManagement – deprecated

Specifies that the user can use the Dashboard Viewer file management features.

This value

- is deprecated and will be removed in a future version of the product
- defines a capability that is initially controlled by the [“Use the Edit Features” on page 1643](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 15 in the capability cookie value

New in Version 8.4 – [“Dashboards” on page 1894](#)

This value was added.

New in Version 10.1.0 – [“IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858](#)

This value is deprecated.

canUseDataManager

Specifies that the user can use Data Manager.

This value

- defines a capability that is initially controlled by the [“Data Manager” on page 1637](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 9 in the capability cookie value

canUseDataSourcesTool

Specifies that the user can use the Data Sources tool. Use this tool to define data sources and connections.

This value

- defines a capability that is initially controlled by the [“Data Source Connections” on page 1620](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 69 in the capability cookie value

New in Version 8.3 – [“Capabilities Refinements” on page 1918](#)

This value was added. It replaces the [bibus](#) » [userCapabilityEnum](#) » [canUseDirectoryTool](#) value.

canUseDescriptiveStatistics

Specifies that the user can use the statistics feature.

The statistics feature uses a third-party application to generate statistical output including charts and tables that can be used within IBM Cognos Analytics applications.

Reserved.

This value

- defines a capability that is initially controlled by the [“Statistics” on page 1668](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 38 in the capability cookie value

New in Version 8.4 GA – [“Statistics” on page 1881](#)

This value was added.

canUseDirectoryTool – obsolete

New in Version 8.3 – [“Capabilities Refinements” on page 1918](#)

This value is obsolete and was removed. Use the following values instead:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDataSourcesTool](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseDistributionListsAndContactsTool](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUsePrintersTool](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseUsersGroupsAndRolesTool](#) value

canUseDistributionListsAndContactsTool

Specifies that the user can use the Distribution Lists and Contacts tool. Use this tool to define distribution lists and contacts.

This value

- defines a capability that is initially controlled by the [“Distribution Lists and Contacts” on page 1621](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 70 in the capability cookie value

New in Version 8.3 – [“Capabilities Refinements” on page 1918](#)

This value was added. It replaces the [bibus](#) » [userCapabilityEnum](#) » [canUseDirectoryTool](#) value.

canUseDrillThroughAssistant

Specifies that the user can use the Drill-through Assistant. Use this tool to assist in the creation of drill-through actions.

This value

- defines a capability that is initially controlled by the [“Drill Through Assistant” on page 1638](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 11 in the capability cookie value

New in Version 8.3 – [“Drill Through Assistant Capability” on page 1913](#)

This value was added.

canUseEV

This value

- defines a capability that is initially controlled by the [“EVStudio” on page 1639](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 86 in the capability cookie value

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

This value was added.

canUseEventStudio

Specifies that the user can use IBM Cognos Event Studio.

This value

- defines a capability that is initially controlled by the [“Event Studio” on page 1639](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 12 in the capability cookie value

canUseExternalData

Specifies that the user can use external data that a package references.

External data refers to data that is not included in a package managed by an administrator. For example, a resource on a user's local file system.

This value

- defines a capability that is initially controlled by the [“Allow External Data” on page 1657](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 90 in the capability cookie value

New in Version 10.1.0 – [“External Data” on page 1851](#)

This value was added.

canUseGlossary

Specifies that the user can use the glossary features of IBM Cognos Analytics.

This value

- defines a capability that is initially controlled by the [“Glossary” on page 1648](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 16 in the capability cookie value

New in Version 8.4 – [“Support for IBM® WebSphere® Business Glossary” on page 1897](#)

This value was added.

canUseHTML

Specifies that the user can use the `button`, `HTMLItem`, and `hyperlink` elements of the report specification.

This capability is based on the report owner when `runAsOwner` is enabled.

This value

- defines a capability that is assigned using the owner's account when the object's `runAsOwner` property or `runWithOwnerCapabilities` property is set to `true`
- defines a capability that is initially controlled by the [“HTML Items in Report” on page 1659](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 35 in the capability cookie value

canUseIndexSearch

Reserved.

This value

- defines a capability that is initially controlled by the [“Execute Indexed Search” on page 1640](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 13 in the capability cookie value

canUseInteractiveDashboardFeatures

Specifies that the user can use interactive dashboard features.

Interactive features include all of the functionality available with the [canUseDashboardViewer](#) capability, in addition to access to the **Actions Toolbar**.

This value

- defines a capability that is initially controlled by the [“Use Interactive Dashboard Features” on page 1642](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 99 in the capability cookie value

New in Version 10.2.0 – “Graduated dashboard capabilities” on page 1835

This value was added.

canUseJob

11.1.2

Specifies that the user can manage jobs.

This value

- defines a capability that is initially controlled by the [“Job” on page 1762](#) object
- defines a capability that has global scope
- defines a capability that is represented by bit 128 in the user capability cookie value

canUseLineage

Specifies that the user can use the lineage features of IBM Cognos Analytics.

This value

- defines a capability that is initially controlled by the [“Lineage” on page 1650](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 18 in the capability cookie value

New in Version 8.4 – “Lineage Metadata” on page 1891

This value was added.

canUseMetricsManagerAdministration

Specifies that the user can use the IBM Cognos Metric Studio administration functions.

This value

- defines a capability that is initially controlled by the [“Metric Studio Administration” on page 1622](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 71 in the capability cookie value

canUseMetricStudio

Specifies that the user can use IBM Cognos Metric Studio.

This value

- defines a capability that is initially controlled by the [“Metric Studio” on page 1651](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 19 in the capability cookie value

canUseMetricStudioEditView

Specifies that the user can edit views in IBM Cognos Metric Studio.

This value

- defines a capability that is initially controlled by the [“Edit View” on page 1651](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 20 in the capability cookie value

canUseMobileAdministration

Reserved.

This value

- defines a capability that has global scope
- defines a capability that is represented by bit 60 in the capability cookie value

New in Version 10.2.1 – “Administrative changes for Cognos Analytics Mobile Reports” on page 1830

This value was added.

canUseMobileService

Specifies that the user can use IBM Cognos Analytics Mobile Reports administration functions.

This value

- defines a capability that is initially controlled by the [“Mobile” on page 1652](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 62 in the capability cookie value

New in Version 10.2.0 – “Restricting access to Cognos Analytics Mobile Reports Service in IBM Cognos Analytics” on page 1842

This value was added.

canUseMonitorActivityTool

Specifies that the user can use the Monitor Activity tool. Use this tool to monitor schedules and past, current and upcoming activities.

This value

- defines a capability that is initially controlled by the [“Run activities and schedules” on page 1626](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 75 in the capability cookie value

New in Version 8.3 – “Capabilities Refinements” on page 1918

This value was added.

canUseMyDataSets

Specifies that the user can use IBM Cognos My Data Sets functions.

This value

- defines a capability that is initially controlled by the [“My Data Sets” on page 1653](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 54 in the capability cookie value

New in Version 10.2.2 – “My data sets” on page 1821

This value was added.

canUseMyDataSetsAdministration

Specifies that the user can use IBM Cognos My Data Sets Administration functions.

This value

- defines a capability that is initially controlled by the [“My Data Sets Administration” on page 1623](#)
- defines a capability that has global scope

- defines a capability that is represented by bit 55 in the capability cookie value

New in Version 10.2.2 – “My data sets” on page 1821

This value was added.

canUseNotebook

11.1.2

Specifies that the user can create, edit, and run a notebook. Users that don't have this capability will only be able to view a notebook.

This value

- defines a capability that is initially controlled by the “Notebook” on page 1763 object
- defines a capability that has global scope
- defines a capability that is represented by bit 129 in the user capability cookie value

canUseObjectCapabilities

Specifies that the user can use the property pages to define user capability policies for individual objects.

This value

- defines a capability that is initially controlled by the “Set Entry-Specific Capabilities” on page 1667
- defines a capability that has global scope
- defines a capability that is represented by bit 29 in the capability cookie value

New in Version 8.4 – “Object Capabilities” on page 1885

This value was added.

canUsePackageDataSources

Reserved.

This value

- defines a capability that is initially controlled by the “Package Data Sources” on page 1653
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 21 in the capability cookie value

New in Version 8.4 – “Package Data Sources” on page 1895

This value was added.

canUsePersonalDataSourceCredentials

Specifies that the user can create and maintain data source credentials associated with their account.

This value

- defines a capability that is initially controlled by the “Manage own data source signons” on page 1650
- defines a capability that has global scope
- defines a capability that is represented by bit 87 in the capability cookie value

New in Version 10.1.0 – “Personal Data Source Credentials” on page 1851

This value was added.

canUsePlanningAdministration

Specifies that the user can use the Planning administration functions.

This value

- defines a capability that is initially controlled by the “Planning Administration” on page 1624

- defines a capability that has both global and object scope
- defines a capability that is represented by bit 72 in the capability cookie value

canUsePlanningContributor

Specifies that the user can use IBM Cognos Contributor.

This value

- defines a capability that is initially controlled by the [“Planning Contributor” on page 1654](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 22 in the capability cookie value

canUsePortalAdministrationTool

Specifies that the user can use the Portal Administration tool. Use this tool to manage portlets and portal layout.

This value

- defines a capability that is initially controlled by the [“Styles and portlets” on page 1628](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 77 in the capability cookie value

canUsePowerPlay

Specifies that the user can use PowerPlay.

This value

- defines a capability that is initially controlled by the [“PowerPlay Studio” on page 1655](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 23 in the capability cookie value

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This value was added.

canUsePowerPlayAdministration

Specifies that the user can use the PowerPlay administration functions.

This value

- defines a capability that is initially controlled by the [“PowerPlay Servers” on page 1625](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 73 in the capability cookie value

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This value was added.

canUsePrintersTool

Specifies that the user can use the Printers tool. Use this tool to define printers.

This value

- defines a capability that is initially controlled by the [“Printers” on page 1625](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 74 in the capability cookie value

New in Version 8.3 – [“Capabilities Refinements” on page 1918](#)

This value was added. It replaces the [bibus](#) » [userCapabilityEnum](#) » [canUseDirectoryTool](#) value.

canUseQueryServiceTool

Specifies that the user can use **Query Service** features in IBM Cognos Administration.

This value

- defines a capability that is initially controlled by the [“Query Service Administration” on page 1626](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 63 in the capability cookie value

New in Version 10.1.0 – [“Query Service Administration Task” on page 1869](#)

This value was added.

canUseQueryStudio

Specifies that the user can access IBM Cognos Query Studio and its secured features.

This value

- defines a capability that is initially controlled by the [“Query Studio” on page 1655](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 24 in the capability cookie value

canUseQueryStudioAdvancedMode

Specifies that the user can access the following advanced features of IBM Cognos Query Studio:

- combined filters and advanced expression editing
- multilingual support
- styles, such as borders, colors, and conditional formatting

This value

- defines a capability that is initially controlled by the [“Advanced” on page 1656](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 25 in the capability cookie value

canUseQueryStudioFileManagement

Specifies that the user can create new queries in IBM Cognos Query Studio and use the Save as function for new reports and custom views.

This value

- defines a capability that is initially controlled by the [“Create” on page 1656](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 26 in the capability cookie value

canUseReportStudio

Specifies that the user can access IBM Cognos Analytics - Reporting and its secured features.

This value

- defines a capability that is initially controlled by the [“Reporting” on page 1657](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 32 in the capability cookie value

canUseReportStudioFileManagement

Specifies that the user can access IBM Cognos Analytics - Reporting object query management operations New, Open, Save As, and Model Connections.

This value

- defines a capability that is initially controlled by the [“Create/Delete” on page 1659](#)

- defines a capability that has global scope
- defines a capability that is represented by bit 34 in the capability cookie value

canUseRepository

Specifies that the user can access external repositories and secured features related to archiving content.

This value

- defines a capability that is initially controlled by the [“External Repositories” on page 1644](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 96 in the capability cookie value

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

canUseScheduling

Specifies that the user can use the scheduling facilities. Use the Scheduling tool to set schedules, and to view schedules and run histories.

This value

- defines a capability that is initially controlled by the [“Scheduling” on page 1661](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 27 in the capability cookie value

canUseSchedulingByDay

Specifies that options to set daily schedules are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by day” on page 1662](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 79 in the capability cookie value

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

canUseSchedulingByHour

Specifies that options to set hourly schedules are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by hour” on page 1663](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 80 in the capability cookie value

New in Version 10.1.0 – “Flexible Scheduling” on page 1851

This value was added.

canUseSchedulingByMinute

Specifies that options to set by minute scheduling are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by minute” on page 1663](#)

- defines a capability that has global scope
- defines a capability that is represented by bit 81 in the capability cookie value

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This value was added.

canUseSchedulingByMonth

Specifies that options to set monthly schedules are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by month” on page 1664](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 82 in the capability cookie value

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This value was added.

canUseSchedulingByTrigger

Specifies that the user or group can set schedules according to a [bibus](#) » [schedule](#) » [triggerName](#).

This value

- defines a capability that is initially controlled by the [“Schedule by trigger” on page 1664](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 83 in the capability cookie value

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This value was added.

canUseSchedulingByWeek

Specifies that options to set weekly schedules are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by week” on page 1665](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 84 in the capability cookie value

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This value was added.

canUseSchedulingByYear

Specifies that options to set yearly schedules are visible to the user or group.

This value

- defines a capability that is initially controlled by the [“Schedule by year” on page 1665](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 85 in the capability cookie value

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This value was added.

canUseSchedulingPriority

Specifies that the user can use the Scheduling tool to view and change the priority on any scheduled task to something other than the default.

Users with this capability will see a Priority option on the schedule dialog where they can filter and run group actions by priority. This functionality will appear in all schedule dialogs. Priorities for pending tasks will appear in the current activities dialog.

The priority of a queued task can be updated using the event » updateEvents(events) method.

This value

- defines a capability that is initially controlled by the “Scheduling Priority” on page 1665
- defines a capability that has global scope
- defines a capability that is represented by bit 28 in the capability cookie value

New in Version 8.3 – “Schedule Priority” on page 1912

This value was added.

canUseSDK – obsolete

canUseSelfServicePackageWizard

Specifies that the user can use the New Package wizard to create new packages.

This value

- defines a capability that is initially controlled by the “Self Service Package Wizard” on page 1667
- defines a capability that has global scope
- defines a capability that is represented by bit 91 in the capability cookie value

New in Version 10.1.0 – “Personal Packages” on page 1863

This value was added.

canUseServerAdministrationTool

Specifies that the user can use the Server Administration tool. Use this tool to configure your environment, perform load balancing, and monitor dispatchers and services.

This value

- defines a capability that is initially controlled by the “Configure and manage the system” on page 1619
- defines a capability that has global scope
- defines a capability that is represented by bit 67 in the capability cookie value

canUseShowHiddenObjectsPreference

Specifies that the user can use the showHiddenObjects preference.

This value

- defines a capability that is initially controlled by the “Hide Entries” on page 1648
- defines a capability that has global scope
- defines a capability that is represented by bit 17 in the capability cookie value

New in Version 8.4 – “Hiding Objects in the Portal” on page 1883

This value was added.

canUseSpecifications

Specifies that the user can use inline specifications in methods such as the asynch » runSpecification(specification, parameterValues, options) method.

This value

- defines a capability that is initially controlled by the “Specification Execution” on page 1668
- defines a capability that has both global and object scope

- defines a capability that is represented by bit 30 in the capability cookie value

canUseUserDefinedSQL

Specifies that the user can edit SQL in a report, query, or report view. Also, enables execution of user-defined, direct-entry SQL.

This value

- defines a capability that is assigned using the owner's account when the object's `runAsOwner` property or `runWithOwnerCapabilities` property is set to `true`
- defines a capability that is initially controlled by the [“User Defined SQL” on page 1661](#)
- defines a capability that has both global and object scope
- defines a capability that is represented by bit 37 in the capability cookie value

canUseUsersGroupsAndRolesTool

Specifies that the user can use the Users, Groups and Roles tool. Use this tool to define groups and roles, and to view users, groups, and roles from external namespaces.

This value

- defines a capability that is initially controlled by the [“Users, Groups, and Roles” on page 1629](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 78 in the capability cookie value

New in Version 8.3 – “Capabilities Refinements” on page 1918

This value was added. It replaces the [bibus » userCapabilityEnum » canUseDirectoryTool](#) value.

canViewContentInRepository

Specifies that the user can view archived content in an external repository.

This value

- defines a capability that is initially controlled by the [“View external documents” on page 1645](#)
- defines a capability that has global scope
- defines a capability that is represented by bit 97 in the capability cookie value

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This value was added.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

validateHintEnum

Defines the hints for report validation.

References

Used by the following properties:

- [bibus » validateOptionHint » value](#)

Members

interactiveHTML

Specifies that the query is used to generate HTML for interactive use.

validateOptionEnum

Defines the options for validation of reports and report specifications.

References

Used by the following properties:

- [bibus](#) » [validateOption](#) » [name](#)

Members

checkMemberUniqueNames

Specifies that the member unique names should be checked.

To specify this option, use the [bibus](#) » [validateOptionBoolean](#) class.

Default: `false`

hint

Specifies a hint for validation.

To specify this option, use the [bibus](#) » [validateOptionHint](#) class.

severity

Specifies the minimum severity of defects to be reported.

To specify this option, use the [bibus](#) » [validateOptionValidateSeverity](#) class.

Default: [bibus](#) » [validateSeverityEnum](#) » [error](#) value

validateSeverityEnum

Defines the severity values for report validation defects.

In order of increasing verbosity, the settings are `error`, `warning`, `keyTransformation`, and `information`.

For more information, refer to [“Value keyTransformation of Enumeration Set validateSeverityEnum Only Partially Works”](#) on page 116.

References

Used by the following properties:

- [bibus](#) » [asynchDetailReportValidation](#) » [maximumSeverity](#)
- [bibus](#) » [validateOptionValidateSeverity](#) » [value](#)

Members

error

Reports all errors returned from the query.

information

Reports errors, warnings, key transformations, and other information messages related to query planning and execution.

keyTransformation

In addition to all errors and warnings, reports messages describing important transformation steps from the report specification to the native query sent to the data source. These messages can be helpful to understand the cause of errors and warnings returned from the query.

warning

Reports all errors and warnings returned from the query.

weeksEnum

Specifies the weeks of a month.

This enumeration set

- defines values that must be specified as string constants

References

Used by the following properties:

- [bibus](#) » [schedule](#) » [monthlyRelativeWeek](#)
- [bibus](#) » [schedule](#) » [yearlyRelativeWeek](#)

Members

first

Specifies the first week of the month.

fourth

Specifies the fourth week of the month.

last

Specifies the last week of the month which, depending on the context, may be the fourth or the fifth week.

second

Specifies the second week of the month.

third

Specifies the third week of the month.

Chapter 17. Secondary requests

Each time you call an asynchronous method, the set of methods that you can use as secondary requests is returned in the `bibus » asynchReply » secondaryRequests` property. Which methods are included depends on the `bibus » asynchReply » status` property. The following table lists valid secondary requests for all methods that return an `bibus » asynchReply` object.

Table 229. Valid secondary requests for each status response

Status	Secondary requests
<code>complete</code>	<code>asynch » release(conversation)</code>
<code>conversationComplete</code>	no additional requests are possible
<code>stillWorking</code>	<code>asynch » cancel(conversation)</code> <code>asynch » wait(conversation, parameterValues, options)</code>
<code>working</code>	<code>asynch » cancel(conversation)</code> <code>asynch » wait(conversation, parameterValues, options)</code>

When the `bibus » asynchReply » status` property is `complete`, the report service may return additional secondary requests, depending on the report status. The report status is stored as the `bibus » asynchDetailReportStatus » status` property. An instance of this class may be returned in the `bibus » asynchReply » details` property.

Table 230. Valid secondary requests for each report status response

Report status	Secondary requests
<code>prompting</code>	<code>promptPaging » getPromptValues(conversation, parameterValues, options)</code> <code>promptPaging » back(conversation, parameterValues, options)</code> <code>promptPaging » forward(conversation, parameterValues, options)</code>
<code>responseReady</code>	<code>report » getOutput(conversation, parameterValues, options)</code>
<code>no report status</code>	<code>promptPaging » getPromptValues(conversation, parameterValues, options)</code> <code>paging » currentPage(conversation, parameterValues, options)</code> <code>report » deliver(conversation, parameterValues, options)</code> <code>paging » firstPage(conversation, parameterValues, options)</code> <code>paging » lastPage(conversation, parameterValues, options)</code> <code>report » lineage(conversation, parameterValues, options)</code> <code>paging » nextPage(conversation, parameterValues, options)</code> <code>paging » previousPage(conversation, parameterValues, options)</code> <code>report » render(conversation, parameterValues, options)</code>

Although most of these methods can be returned only when the `bibus » asynchReply » status` property is `complete`, if the option `alwaysIncludePrimaryRequest` is `true`, the following methods may also be returned when the `bibus » asynchReply » status` property is `working` or `stillWorking`:

- `report » deliver(conversation, parameterValues, options)`
- `report » render(conversation, parameterValues, options)`

Note that methods listed here are not always available. For example, the `promptPaging » back(conversation, parameterValues, options)` method is not available on the first prompt page and the `paging » previousPage(conversation, parameterValues, options)` method cannot follow

the `paging » firstPage(conversation, parameterValues, options)` method. The `promptPaging » getPromptValues(conversation, parameterValues, options)` method is returned only when the output contains a tree control that supports dynamic population of its data. Always check the `bibus » asynchReply » secondaryRequests` property for the current set of valid methods.

Example: Checking Valid Secondary Requests in Java

To see the code in context, view the sample in the following location:

`installation_location/sdk/java/SendEmail/Email.java`

This Java code sample demonstrates how to check whether a specific method is a valid secondary request:

```
public boolean hasSecondaryRequest(
    AsynchReply response,
    String secondaryRequest)
{
    AsynchSecondaryRequest[] secondaryRequests =
        response.getSecondaryRequests();
    for (int i = 0; i < secondaryRequests.length; i++)
    {
        if (secondaryRequests[i].getName().compareTo(secondaryRequest)
            == 0)
        {
            return true;
        }
    }
    return false;
}
```

Chapter 18. Code samples and language-specific coding practices

The IBM Cognos code samples encompass the following programming languages, each one designed to expose different aspects of the IBM Cognos Software Development Kit:

- active server pages (.asp files)
- Java
- C# .NET

These samples use the BI Bus API to read from and write to the content store. You can use them as learning tools or as examples to help you develop your own applications.

The IBM Cognos Software Development Kit installation also includes

- report samples that you can open using IBM Cognos Analytics - Reporting to view the XML of the underlying report specifications. For more information, see [Chapter 23, “Using report specifications,” on page 1457](#).
- additional XML code snippets for other common elements that you can use in a report specification.

Active Server Page Samples

The following commented Active Server Page (ASP) samples are included with IBM Cognos:

- URLReport.asp, which illustrates how to run and view predefined report samples, after selecting one from the list of available sample reports in the URLInput.html file
- URLEditReport.asp, which illustrates how to use the GET method to open a report for editing in IBM Cognos Analytics - Reporting or for editing a query in IBM Cognos Query Studio, after selecting it from the list of available sample queries in the URLEditReportSelect.html file

ASP Sample File Locations

The ASP sample programs are located in the directory *installation_location/webcontent/samples/sdk/asp*. The databases for the Great Outdoors sample reports on which they are based are located in the directory *installation_location/webcontent/samples/datasources/sqlserver*.

Java Samples

The IBM Cognos Software Development Kit includes Java program samples that show you some types of applications you can design. The samples include source files so that you can test making changes to the sample code, and batch files or shell scripts for compiling and running the samples.

The source files contain comments that describe the main purpose of each sample, including a summary of which BI Bus API methods are used. The batch files and shell scripts contain instructions that you must follow before you run them.

Before you modify any of the samples, we recommend that you become familiar with basic Java programming techniques.

The generated source code obtained from the IBM Cognos Web Service Definition Language (WSDL) file is installed in the .jar file *installation_location/sdk/java/src/axisCognosClientSrc.jar*.

The generated Java documentation that describes the classes and methods in the BI Bus API is installed in the folder *installation_location/sdk/java/doc/cognos8/javadoc*.

Java Sample File Locations

The sample files are installed in subdirectories under the *installation_location/sdk/java* directory. Each sample is in its own subdirectory. These subdirectories are as follows:

- Agents
This sample creates and runs an agent.
- Alerts
This sample demonstrates how to manage alert lists for reports.
- CancelExec
This sample cancels a running report.
- CapabilitiesGUI
This sample adds the `canUseReportStudio` capability to the selected account or role for the selected package.
This directory contains the user interface for the Capabilities sample. The methods used by this sample are in the Security directory.
- Common
This directory contains functionality that is used by a number of the Java samples. `CRNConnect.java` is used to establish a connection to IBM Cognos. `ReportWizardDialog.java` provides a user interface for selecting columns for a report.
- ContentStoreExplorer
This sample displays all the objects in the content store.
- CreateDrillThrough
This sample creates a drill-through definition and sets its properties.
- DeployPackage
This sample demonstrates how to specify values for the options that control Content Manager deployments.
- DispatcherControl
This sample displays dispatcher properties, starts and stops individual services, and sets dispatcher settings.
- EventTrigger
This sample notifies the IBM Cognos server that an external event has occurred, for the purpose of triggering trigger-based schedules.
To use trigger-based report scheduling, the user must schedule a report based on the trigger and you must set up the event on a server. Using this sample, you can link the occurrence of an external event, such as a database refresh or an email, with a trigger on the IBM Cognos server that causes the report to run.
- ExecReports
This sample contains methods for executing different types of reports.
- ExecReportsAt
This sample submits a report as a job. When the report executes, the output is saved in the content store as a report view, which is named after and located with the report.
- GroupsAndRolesGUI
This sample allows you to manipulate groups and roles within your secured namespace. You can add users to and delete users from groups and roles, and you can add and delete groups and roles within your namespace.

This directory contains the user interface for the GroupsAndRoles sample. The methods used by this sample are in the Security directory.

- HandlersCS

This directory contains methods that access the content store. Java methods in this directory are used by a number of the Java samples to query, add, move and delete content within Content Manager, and to handle exceptions.

- PermissionsGUI

This sample denies the current user read permission to that user's MyFolders folder.

This directory contains the user interface for the Permissions sample. The methods used by this sample are in the Security directory.

- PrintReport

This sample prints a report that you specify. You can also add entries to the list of printers, or remove existing ones from it.

- QueryCM

This sample gets the objects from the content store in the location specified by the search option. The application file contains comments to help you follow the sequence.

- RenderReport

This sample demonstrates how to run a report in HTML format and then render the report output in XML.

- ReportAdd

This sample demonstrates how to add a report to the content store based on an existing report specification.

- ReportCopyMove

This sample copies or moves a report to another location.

- ReportCreate

This sample allows you to create a new report. It prompts you to select the columns that you want to add to the report.

- ReportDelete

This sample deletes a report from the content store.

- ReportParams

This utility class is used by other samples to execute a prompted report. It retrieves the parameters required by the report, builds an array of parameter values, and then passes those parameters to the execute method to run the report.

- ReportSpec

When you run this sample, you will be prompted to select a column to add to a report. This sample modifies a report specification using a Document Object Model (DOM) parser. It retrieves the report specification from the IBM Cognos Analytics server, changes the specification using the DOM parser, and then saves the modified report back to the server.

- ReportUpgrade

This sample upgrades the specification property of report objects in the content store to IBM Cognos. It also allows report specifications to be extracted to the local file system, as they exist, or upgraded to IBM Cognos.

- runreport

This sample runs a report and saves the output as HTML. By default, the selected report is Show Detailed Rows and Summaries in the Documentation Report Samples folder of the GO Sales and Retailers package. You can change the report to run by editing the Java code.

When the report execution is complete, the report appears in the Java samples location *installation_location/webcontent/samples*. You can then open the file to view the result.

- SaveAs

This sample executes a report and saves either the report or the report output back in the content store.

- Scheduler

This sample creates a new schedule for a report.

- Security

These samples allow you to perform a number of security related tasks, such as logging on to a secured namespace, logging off, and displaying the current logon information. If you are not logged on when you attempt to display your logon information, you will be prompted to log on.

Although most of the samples will work when Anonymous access is enabled, they serve as more robust examples when a secured namespace is used.

- SendEmail

This sample demonstrates how to run a report and send the output as an email to a specific user. The IBM Cognos server must be configured with permissions to send email to a valid email server.

- Submit

This sample submits a report for execution as a job.

- TestDIMS

This sample illustrates running `dimensionManagementService` requests. IBM Cognos Business Viewpoint Server must be installed and running before using this sample.

- TesterCM

This sample tests that the content store is operational.

- ViewAll

This sample displays a list of all packages, reports, and queries in the content store.

- ViewCMPackages

This sample displays a list of all packages in the content store.

- ViewCMReports

This sample displays a list of all reports and queries in the content store.

This list is also available in the file *installation_location/sdk/java/JavaSamples.html*, which includes links to the explanation files for the samples.

Set Up Your System to Use the Java Samples

Each subdirectory in *installation_location/sdk/java* contains the following files.

build.bat

Builds the individual sample on Windows.

run.bat

Runs the individual sample on Windows.

build.sh

Builds the individual sample on UNIX or Linux.

run.sh

Runs the individual sample on UNIX or Linux.

In addition, the *installation_location/sdk/java* directory contains the following files.

build-samples.bat

Builds all the Java samples on Windows.

build-samples.sh

Builds all the Java samples on UNIX or Linux.

Before you modify any of the samples, we recommend that you become familiar with basic Java programming techniques.

Before you use these Java samples, you should identify whether you have Anonymous access enabled. Although the samples will work with anonymous access, security features will not be demonstrated if anonymous access is enabled. To use the security features of the samples, ensure that you have a secured NTLM, LDAP, or other namespace, and that you disable anonymous access.

Because some of the Java samples issue output to the command console, ensure that the console is visible when you run the samples.

Steps for Windows

1. Install a Java Development Kit (JDK). To determine the supported Java versions, visit the [Software Product Compatibility Reports web site](https://www.ibm.com/software/reports/compatibility/clarity/) (<https://www.ibm.com/software/reports/compatibility/clarity/>). Select your product and product version and create a report for related software. Choose Development Tools under Supported software.

The report that is generated lists the supported Java versions.

2. Ensure that your Windows PATH environment variable includes the location where the JDK is installed.
3. Edit *installation_location/sdk/java/Common/CRNConnect.java* by locating the line

```
public static String CM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
```

and replacing localhost with the name and, if necessary, port number of your IBM Cognos server.
4. If you want to compile all the samples, edit *installation_location/sdk/java/build-samples.bat* and change the lines

```
set JAVA_HOME=c:/jdk1.5
```

```
set CRN_HOME = ../..
```

so that they point to the locations where the JDK and IBM Cognos are installed, respectively.

5. If you want to compile individual samples, edit *installation_location/sdk/java/sample_name/build.bat* and change the lines

```
set JAVA_HOME=c:/jdk1.5
```

```
set CRN_HOME = ../..
```

so that they point to the locations where the JDK and IBM Cognos are installed, respectively.

6. Compile the Java samples by running *build-samples.bat* (to compile all samples) or *build.bat* (to compile an individual sample).
7. Use the *run.bat* script files to run the samples, as instructed in the code comments.

Steps for UNIX[®] or Linux

1. Install a Java Development Kit (JDK). To determine the supported Java versions, visit the [Software Product Compatibility Reports web site](https://www.ibm.com/software/reports/compatibility/clarity/) (<https://www.ibm.com/software/reports/compatibility/clarity/>). Select your product and product version and create a report for related software. Choose Development Tools under Supported software.

The report that is generated lists the supported Java versions.

2. Set the JAVA_HOME environment variable to point to the location where the JDK is installed.
3. Edit *installation_location/sdk/java/Common/CRNConnect.java* by locating the line

```
public static String CM_URL = "http://localhost:9300/p2pd/servlet/dispatch";
```

and replacing localhost with the name and, if necessary, port number of your IBM Cognos server.

4. If you want to compile all the samples, edit *installation_location/sdk/java/build-samples.sh* and change the lines

```
CRN_HOME=/usr/cognos/c10
```

```
JAVA_HOME=/c/j2sdk1.5
```

so that they point to the locations where IBM Cognos and the JDK are installed, respectively.

5. If you want to compile individual samples, edit *installation_location/sdk/java/sample_name/build.sh* and change the lines

```
CRN_HOME=/usr/cognos/c10
```

```
JAVA_HOME=/c/j2sdk1.5
```

so that they point to the locations where IBM Cognos and the JDK are installed, respectively.

6. Compile the Java samples by running *build-samples.sh* (to compile all samples) or *build.bat* (to compile an individual sample).

7. Use the *run.sh* script files to run the samples, as instructed in the code comments.

C# .NET Samples

Read the *csharp_sample_name_Explain.html* file before you run any of the C# .NET samples. These files provide instructions about running the sample programs, and contain information about associated files. They are located in *installation_location/sdk/csharp/sample_name*.

In addition, the source files contain comments to help you follow the sequence, and understand what each sample does. We recommend that you become familiar with programming techniques before you modify any of the samples.

The C# .NET samples include executable programs, located in *installation_location/sdk/csharp/bin*. If you receive an automation error when you launch an .exe program, you must recompile the program for your environment.

The C# .NET samples include the following applications

- AddReport

This sample demonstrates how to add a report to the content store based on an existing report specification.

- Cancel

This sample cancels a running report.

- Capabilities

This sample removes the Administration capability from the user running the sample.

This directory contains the user interface for the Capabilities sample. The methods used by this sample are in the Security directory.

- CMQuery

This sample gets the objects from the content store in the location specified by the search option.

- CMTester

This sample tests whether the content store is operational.

- ContentStoreExplorer

This sample displays all the objects in the content store.

- Copy

This sample copies an object in the content store.

- CreateDrillThroughTarget

This sample creates a drillthrough target.

- CreateReport

This sample creates a report.

- DeleteReport

This sample deletes a report that you specify.

- Email

This sample demonstrates how to run a report and send the output as an email to a specific user. The IBM Cognos server must be configured with permissions to send email to a valid email server.

- ExecuteReport

This sample contains methods for executing different types of reports.

- Move

This sample moves an object in the content store.

- Permissions

This sample denies the current user read permission to that user's MyFolders folder.

This directory contains the user interface for the Permissions sample. The methods used by this sample are in the Security directory.

- PrintReport

This sample prints a report that you specify. You can also add entries to the list of printers, or remove existing ones from it.

- ReportParameters

This sample gets or retrieves parameters associated with the specified report.

- reportrunner

This sample runs a report that you specify, and produces results that are saved in HTML format.

- SamplesCommon

This sample includes many common classes that are used by the other C# .NET samples. It does not include a sample executable program, but instead builds a DLL that the other samples use.

- Save

This sample runs a report that you specify, and saves the output or allows you to save it with a new name.

- Schedule

This sample creates a new schedule for a report.

- Security

These samples allow you to perform a number of security related tasks, such as logging on to a secured namespace, logging off, and displaying the current logon information. If you are not logged on when you attempt to display your logon information, you will be prompted to log on.

Although most of the samples will work when Anonymous access is enabled, they serve as more robust examples when a secured namespace is used.

- Submit

This sample creates a new job and submits it.

- ViewAll

This sample displays a list of all packages, reports, and queries in the content store.

- ViewPackages

This sample displays a list of all packages in the content store.

- ViewReports

This sample displays a list of all reports, and queries in the content store.

This list is also available in the file *installation_location/sdk/csharp/CSharpSamples.html*, which includes links to the explanation files for the samples.

C# .NET Sample File Locations

The sample files are installed in various subdirectories in this location *installation_location/sdk/csharp*. There is one subdirectory for each sample. All executable files are located in *installation_location/sdk/csharp/bin*.

Each of the subdirectories contain the following files:

- AssemblyInfo.cs
- *sample_name*.cs
- *sample_name*.csproj
- *sample_name*.sln
- build.bat
- *csharp_sample_name_*Explain.html

The following files are referenced by all the samples and are located in *installation_location/sdk*:

- cognosdotnet_10_2.dll
- cognosdotnetassembly_10_2.dll

The following considerations should be noted.:

- The solution (.sln) and project (.csproj) files require Visual Studio 2010.
- AllSamples.sln in *installation_location/sdk/csharp* includes all the sample projects. We recommend that you use this solution when you want to build all the C# .NET samples at the same time.
- The reportrunner sample also includes a file named Cognos8Exception.cs.

Set Up Your System to Use the C# .NET Samples

To run the toolkit samples with .NET, you must have Version 4.0 of the .NET Framework installed.

To modify or rebuild the C# .NET samples, you must have a C# development environment installed, such as Visual Studio 2010 or the .NET Framework Software Development Kit (SDK) v4.0.

In each sample, the build.bat script included with the sample code shows one way of building the application using the Visual Studio .NET compiler. For more information about using build.bat, see the comments in the file.

Chapter 19. Language-specific data type mappings

This chapter contains the language-specific data type mappings.

C# Equivalents

The following table maps XML data types used in the WSDL file to their equivalents in C#.

XML	C#
xsd:anytype	None
xsd:base64Binary	byte
xsd:boolean	bool
xsd:date	System.DateTime (Convert DateTime objects to UTC before using them.)
xsd:dateTime	System.DateTime (Convert DateTime objects to UTC before using them.)
xsd:decimal	decimal
xsd:float	float
xsd:int	int
xsd:integer	string
xsd:QName	None
xsd:string	string
xsd:time	System.DateTime (Convert DateTime objects to UTC before using them.)

Java Equivalents

The following table maps XML data types used in the WSDL file to their equivalents in Java.

XML	Java
xsd:anytype	None
xsd:base64Binary	byte[]
xsd:boolean	boolean

Table 232. Java-to-XML datatype reference (continued)

XML	Java
xsd:date	java.util.Calendar (Convert Calendar objects to UTC before using them.)
xsd:dateTime	java.util.Calendar (Convert Calendar objects to UTC before using them.)
xsd:decimal	decimal
xsd:float	float
xsd:int	int
xsd:integer	java.math.BigInteger
xsd:QName	None
xsd:string	java.lang.string
xsd:time	java.util.Calendar (Convert Calendar objects to UTC before using them.)

XML-SQL Data Type Mappings

Data types used in the IBM Cognos schema are mapped automatically, as mandatory attributes, to their equivalents in the database. If the built-in parser cannot recognize a data type, an exception occurs. For example, a Microsoft® SQL Server database returns a sqlUnknownType as the data type and stops processing the request.

The following table maps the XML data types used by IBM Cognos to their SQL database column equivalents. Notes about user-supplied scale, precision, and length values are included, where applicable.

Table 233. SQL-to-XML datatype reference

XML data type	SQL equivalent	Notes
xsd:base64Binary	sqlBlob	Not supported.
xsd:boolean	sqlSmallInt	Int16
xsd:date	sqlDate2	
xsd:dateTime	sqlTimestampTZ	Timestamp with time zone
xsd:dateTime	sqlTimestamp2	Date and timestamp. There are also separate data types for both date and time.
xsd:decimal	sqlDecimal (if scale is user-supplied)	If scale and precision are not user-supplied, converts to sqlQuad (defaults: 0; 19).
xsd:decimal	sqlNumeric	If scale and precision are not user-supplied, converts to sqlQuad (defaults: 0; 19).

Table 233. SQL-to-XML datatype reference (continued)

XML data type	SQL equivalent	Notes
xsd:duration	sqlInterval2	Interval day-to-second
xsd:duration	sqlIntervalYM	Interval year-to-month
xsd:float	sqlFloat	
xsd:int	sqlInteger	Int32
xsd:integer	sqlInteger	Int32
xsd:QName	sqlVarChar	Same for xsd:anyURI and xsd:NMTOKEN.
xsd:string	sqlLongVarChar	Converted to sqlVarChar based on user-supplied scale, precision, or length default precision = (32K-1).
xsd:string	sqlNChar	National character
xsd:string	sqlNVarChar	National character (varying). Same for xsd:anyURI and xsd:NMTOKEN.
xsd:time	sqlTimeTZ	Time with time zone
xsd:time	sqlTime2	
xsd:unknown	sqlUnknownType	Exception thrown; all processing stops.

Chapter 20. Performing tasks using URLs

The URLs provide a quick and efficient way to start components and open specified content, such as reports, metrics, folders, or pages.

You can use the URL Report sample program included with the IBM Cognos Software Development Kit to see examples that demonstrate how to perform basic tasks by clicking embedded links on an active server page. However, for complex tasks, such as scheduling, use the Software Development Kit to create a custom application.

You can use various declarations to identify the requested action, depending on the component.

Note: These declarations are not the same as the Software Development Kit methods exposed by the BI Bus API.

For more information on using URLs see the *Administration and Security Guide*.

Recommendations for URLs

Depending on your goals, some or all of the following URL-programming recommended practices described here may apply to your situation.

The recommended practices include the following:

- Ensure that your URLs do not include spaces. For example, if you use JavaScript for a post declaration, you must convert any spaces into the %20 form. (If you enter values using the get technique, the Web browser handles this encoding for you.)
- Use the equal sign (=) followed by a single-quotation-mark-enclosed space when submitting empty form variables to accommodate the notational requirements of all web servers. For example, to end a complex type array in a post declaration, use the following syntax: ...name='EA' value=' />
- Use &backURL= syntax to specify the URL location to return to, when users click Return in their output window.

Tip: To avoid launching a new browser window, you can specify a target name of "_self" as an attribute of the <a> anchor tag.

CGI Program and Alternative Gateways

All URL commands begin with a declaration of the end point for the request: either cognos.cgi or an alternative gateway.

For IBM Cognos Analytics, the complete syntax is `http://webservername:portnumber/ibmcognos/bi/v1/disp`.

If you configured IBM Cognos Analytics to use a gateway other than the default CGI program, type the URL that corresponds to your gateway. For information about configuring gateways, see the *IBM Cognos Analytics Installation and Configuration Guide*.

The supported gateways are listed in the following table:

Gateway	URL
ISAPI	<code>http://webservername/ibmcognos/isapi</code>
Apache Connector (Microsoft Windows operating system)	<code>http://webservername/ibmcognos/cgi-bin/mod_cognos.dll</code>

<i>Table 234. Supported gateways (continued)</i>	
Gateway	URL
Apache Connector (Solaris and AIX®)	http://webservername/ibmcognos/cgi-bin/mod_cognos.so
Apache Connector (HPUX)	http://webservername/ibmcognos/cgi-bin/mod_cognos.sl
Gateway Servlet	http://webservername:9300/ServletGateway/servlet/Gateway
CGI	http://webservername/ibmcognos

URL Methods

There are two methods to start most IBM Cognos components: parameterized URL and cognosLaunch. Both methods perform the same function and use the same parameters. You can use either method to perform many UI tasks.

Parameterized URL Method

The parameterized URL method performs tasks or starts specific components using parameters typed in the Web browser address bar.

Using both get and post methods, the `launch.xts` or `cc.xts` parameter starts the specified component.

This method requires that the parameters use the following format:

```
&ArgumentName=ArgumentValue
```

The `ArgumentName` parameter specifies the type, and the `ArgumentValue` parameter specifies the value of the called arguments. All names and values must be character strings.

This method is easier to begin using than the `cognosLaunch` method because it does not require advance preparation. However, the longer URL-encoded entries are restrained by browser character limits.

You can use these commands to browse content or view pages.

Mandatory Parameterized URL Parameters

If you are using the parameterized URL method to start any component, you must use the following parameters with this syntax:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/launch.xts
```

followed by the specific component parameters you want to use.

If you want to start the viewer using the parameterized URL method, use the following parameters with this syntax:

```
http://localhost/bi/v1/disp?b_action=cognosViewer
```

If you want to start the portal using the parameterized URL method, use the following URL:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/cc.xts
```

The mandatory building blocks for parameterized URL commands are as follows:

- `gateway`

This is the mandatory argument value that specifies the gateway. The gateway name in this example is `http://server/ibmcognos/bi/v1/disp?`.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#).

- `b_action=xts.run`

Identifies the action. To specify the viewer, `b_action=cognosViewer`. For more information see, [“Starting IBM Cognos Viewer” on page 1440](#).

- `m=portal/launch.xts&ui.tool=tool_name`

Identifies the component interface that displays the result. To start the portal, use `m=portal/cc.xts`.

- `ui.action`

Specifies the action to take. Acceptable values for the Studio components are `new` and `edit`. The default is `new`. Acceptable values for the viewer are `run` and `view`. The default is `view`.

cognosLaunch Method

The `cognosLaunch` method uses a JavaScript function to perform tasks and start components.

To use the launch utility in a Web page, you must first include the following statement in the HTML file from which you start the specific component:

```
<script language="JavaScript" src="CognosGateway/cognoslaunch.js">
</script>
```

`CognosGateway` is the main IBM Cognos Analytics gateway defined in IBM Cognos Configuration.

This statement enables the page to open a specified report in the chosen IBM Cognos component using the `cognosLaunch` JavaScript parameters.

This method requires that the parameters use the following format:

```
"Argument Name", "Argument Value"
```

The `Argument Name` parameter specifies the type, and the `Argument Value` parameter specifies the value of the called arguments. All names and values must be character strings.

The `cognosLaunch` method uses a simpler construction than the parameterized URL method, but requires an enabled starting page.

Mandatory cognosLaunch Parameters

If you are using the `cognosLaunch` method to start any component, use the following parameters with this syntax:

```
cognosLaunch("ui.gateway", " gateway ", "ui.tool", "
component")
```

- `"ui.gateway"`

This is the mandatory argument value that specifies the gateway.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#).

- `"ui.tool"`

This is the mandatory argument value that specifies the component.

- `ui.action`

Specifies the action to take. Acceptable values for the Studio components are `new` and `edit`. The default is `new`. Acceptable values for IBM Cognos Viewer are `run` and `view`. The default is `view`.

You cannot use this parameter with Metric Studio.

Common Optional Parameters

In addition to the mandatory parameters required by each IBM Cognos Analytics component, you can use the following optional parameters unless otherwise specified.

- `ui.object`

Specifies the path of the target object. Acceptable values are the Content Manager search path or store ID.

This parameter is mandatory for Event Studio, Analysis Studio, and Metric Studio.

- `ui.folder`

Specifies the target folder. Acceptable values are the Content Manager search path or store ID.

You cannot use this parameter with Analysis Studio.

- `ui.backURL`

Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#).

URL Validation

IBM Cognos Application Firewall validation is enforced on URLs using the following rules.

- Fully qualified, or absolute URLs:

```
protocol://host[:port]/path[?query]
```

Where `protocol` is either `http` or `https` and the `host` is validated against the valid domain list, which is specified by the administrator in IBM Cognos Configuration. For more information, see the *IBM Cognos Analytics Installation and Configuration Guide*.

- URLs relative to the IBM Cognos Analytics installation web root:

```
/<install root>/.*
```

Where `<install root>` is the gateway file path, taken from the Gateway URI from Cognos Configuration Tool. For example, `/ibmcognos/ps/portal/images/`.

- One of the following specifically allowed URLs:

- `about:blank` (case insensitive)
- `JavaScript>window.close()` (case insensitive, with or without trailing semi-colon)
- `JavaScript:parent.close()` (case insensitive, with or without trailing semi-colon)
- `JavaScript:history.back()` (case insensitive, with or without trailing semi-colon)
- `parent.cancelErrorPage()` (case insensitive, with or without trailing semi-colon)
- `doCancel()` (case insensitive, with or without trailing semi-colon)

In addition, an advanced configuration setting, `RSVP.RENDER.VALIDATEURL`, can be used to specify whether these rules are applied to values specified by any URL values contained within a report specification. CAF must be enabled for the `RSVP.RENDER.VALIDATEURL` setting to take effect.

Starting IBM Cognos Analytics Components

Use a URL to start IBM Cognos Analytics components and open specified content.

The components can be started from any enabled Web page.

You can use a URL to start the following components:

- Query Studio
- Analysis Studio
- Metric Studio
- Event Studio
- IBM Cognos Viewer
- PowerPlay Studio

Start Parameters

Before using either the parameterized URL or `cognosLaunch` method you must first locate the object that you want to access.

The easiest way to identify the location of an object, such as a saved report, is to copy the object search path into the required URL command. The full path must be copied, as listed in the report properties, including the relevant package name and report names.

For example, using the following parameterized URL starts IBM Cognos Viewer and runs the report named 2005 Sales Summary:

```
http://localhost/bi/v1/disp?b_action=cognosViewer&ui.  
action=run&ui.object=/content/folder  
[@name='Samples']/folder  
[@name='Models']/package  
[@name='GO DataWarehouse (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='EmployeeSatisfaction 2006']
```

The following script performs the same function using the `cognosLaunch` method in an enabled Web page:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos  
/bi/v1/disp','ui.tool','CognosViewer','ui.  
action','run','ui.object','/content/folder  
[@name='Samples']/folder[@name='Models']/package  
[@name='GO DataWarehouse (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='EmployeeSatisfaction 2006']')
```

The following topics describe each of the parameters used in this example, and examples of some other commonly used parameters.

Starting Query Studio

You can use a URL to quickly open a specific report in IBM Cognos Query Studio.

Use the following mandatory parameters to start Query Studio with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=  
portal/launch.xts&ui.tool=  
QueryStudio&ui.object=/content&ui.action=new
```

Use the following mandatory parameters to start Query Studio with the cognosLaunch method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.  
tool','QueryStudio','ui.action','new')
```

The `ui.action` parameter is a mandatory for both methods.

The following parameters are optional for Query Studio:

- `cv.header`
Specifies whether to display the header. Acceptable values are `true` and `false`.
- `ui.spec`
Specifies an XML document that contains an IBM Cognos Analytics report specification. For information about IBM Cognos Analytics report specifications, see the report specification topics in the IBM Cognos Software Development Kit *Developer Guide*.
- `run.outputLocale`
Specifies the output language. Acceptable values are expressed as a hyphenated language-region pair, in accordance with the RFC3066 standard. The default value is `en-us`.
- `run.prompt`
Specifies whether the report service issues prompts, so you can enter report option values. Acceptable values are `true` and `false`. The default value is `true`.
- `ui.object`
Specifies the path of the target object. Acceptable values are the Content Manager search path or store ID.
- `ui.folder`
Specifies the target folder. Acceptable values are the Content Manager search path or store ID.
- `ui.backURL`
Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#).

Use the following parameters for debugging purposes only:

- `run.outputFormat`
Specifies the output format. Acceptable values are CSV, HTML, PDF, XHTML, XLWA, and XML.
- `asynch.primaryWaitThreshold`
Specifies the maximum amount of time, in seconds, that the server can use to process the request before sending a response to the client. Acceptable values are any integer. Use a value of 0 to make the client wait indefinitely. The default value is 7.
- `asynch.secondaryWaitThreshold`
Specifies the maximum amount of time, in seconds, that the server can use to process the request before sending a response to the client. Acceptable values are any integer. Use a value of 0 to make the client wait indefinitely. The default value is 30.

For a list of all common optional launch parameters for both methods, see [“Common Optional Parameters” on page 1434](#).

Parameterized URL Examples

This section provides examples for performing specific functions when starting IBM Cognos Query Studio using the parameterized URL method.

- Starting Query Studio to a specific package

```
http://localhost/bi/v1/disp?b_action=xts.run&m=
portal/launch.xts&ui.tool=
QueryStudio&ui.object=/content/folder[@name='Samples']/folder
[@name='Models']/package
[@name='GOData Warehouse (analysis)']&ui.action=new
```

- Starting Query Studio to a specific report

```
http://localhost/bi/v1/disp?b_action=xts.run&m=
portal/launch.xts&ui.tool=
QueryStudio&ui.object=/content/folder[@name='Samples']/folder
[@name='Models']/package[@name='GOData Warehouse (analysis)']/folder
[@name='Query Studio Report Samples']/query
[@name='ReturnQuantity by Product Line']&ui.action=edit
```

cognosLaunch Examples

This section provides examples for performing specific functions when starting IBM Cognos Query Studio using the CognosLaunch method.

- Starting Query Studio to a specific package

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.
tool','QueryStudio','ui.action','new','ui.object','/content/folder
[@name=\'Samples\']/folder
[@name=\'Models\']/package
[@name=\'GO DataWarehouse (analysis)\']')
```

- Starting Query Studio to a specific report

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.
tool','QueryStudio','ui.action','edit','ui.object','/content/folder
[@name=\'Samples\']/folder[@name=\'Models\']/package
[@name=\'GO DataWarehouse (analysis)\']/folder
[@name=\'Query Studio Report Samples\']/query
[@name=\'ReturnQuantity by Product Line\']')
```

Starting Analysis Studio

You can use a URL to quickly open and run a specific report in IBM Cognos Analysis Studio.

Use the following parameters to start Analysis Studio with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=
portal/launch.xts&ui.gateway=
http://localhost/ibmcognos/bi/v1/disp&ui.
tool=AnalysisStudio&ui.action=new
```

When starting Analysis Studio with the parameterized URL method, specify both the gateway (<http://localhost/ibmcognos/bi/v1/disp?>) and the `ui.gateway` parameter.

Use the following parameters to start Analysis Studio with the `cognosLaunch` method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.
tool','AnalysisStudio','ui.action','new')
```

The `ui.object` parameter is mandatory for both methods. However, if it is missing, you are prompted to select a package.

In addition to the mandatory parameters required, you can also use the following optional parameter:

- `ui.backURL`

Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#)

Parameterized URL Examples

This section provides examples for performing specific functions when starting IBM Cognos Analysis Studio using the parameterized URL method.

- Starting Analysis Studio to a specific package

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/launch.xts&ui.gateway=http://localhost/ibmcognos/bi/v1/disp&ui.tool=AnalysisStudio&ui.object=/content/folder[@name='Samples']/folder[@name='Cubes']/package[@name='Salesand Marketing (cube)']/folder[@name='Analysis Studio Report Samples']&ui.action=new
```

- Viewing an analysis report in Analysis Studio

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/launch.xts&ui.gateway=http://localhost/ibmcognos/bi/v1/disp&ui.tool=AnalysisStudio&ui.tool=AnalysisStudio&ui.object=/content/folder[@name='Samples']/folder[@name='Cubes']/package[@name='Salesand Marketing (cube)']/folder[@name='Analysis Studio Report Samples']/analysis[@name='CustomRank Sample']&ui.action=edit
```

cognosLaunch Examples

This section provides examples for performing specific functions when starting IBM Cognos Analysis Studio using the CognosLaunch method.

- Starting Analysis Studio to a specific package

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','AnalysisStudio','ui.action','new','ui.object','/content/folder[@name=\'Samples\']/folder[@name=\'Cubes\']/package[@name=\'Salesand Marketing (cube)\']/folder[@name=\'Analysis Studio Report Samples\']')
```

- Viewing an analysis report in Analysis Studio

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','AnalysisStudio','ui.action','edit','ui.object','/content/folder[@name=\'Samples\']/folder[@name=\'Cubes\']/package[@name=\'Salesand Marketing (cube)\']/folder[@name=\'Analysis Studio Report Samples\']/analysis[@name=\'CustomRank Sample\']')
```

Starting Metric Studio

You can use a URL to quickly open a metric in IBM Cognos Metric Studio.

Use the following parameters to start Metric Studio with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/launch.xts&ui.tool=MetricStudio&ui.action=new
```


Use the following parameters to start Metric Studio with the cognosLaunch method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.  
tool','MetricStudio','ui.action','new')
```

The `ui.object` parameter is mandatory for both methods. However, if it is missing, you are prompted to select a package.

In addition to the mandatory parameters required, you can also use the following optional parameters:

- `ui.folder`

Specifies the target folder. Acceptable values are the Content Manager search path or store ID.

- `ui.backURL`

Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#)

Parameterized URL Examples

This section provides an example when starting IBM Cognos Metric Studio using the parameterized URL method.

- Starting Metric Studio to a specific package

```
http://localhost/bi/v1/disp?b_action=xts.run&m=  
portal/launch.xts&ui.  
tool=MetricStudio&ui.object=/content/package  
[@name='GOMetrics']&ui.action=new
```

cognosLaunch Examples

This section provides an example when starting IBM Cognos Metric Studio using the CognosLaunch method.

- Starting Metric Studio to a specific package

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.  
tool','MetricStudio','ui.action','new','ui.object','/content/package  
[@name='GOMetrics\']')
```

Starting Event Studio

You can use a URL to quickly access and edit an agent in IBM Cognos Event Studio.

Use the following parameters to start Event Studio with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=xts.run&m=  
portal/launch.xts&ui.tool=EventStudio&ui.object=/content/package  
[@name='GOSales (analysis)']&ui.action=new
```

Use the following parameters to start Event Studio with the cognosLaunch method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.  
tool','EventStudio','ui.action','new')
```

The `ui.action` and `ui.object` parameters are mandatory. If `ui.object` is missing, you are prompted to select a package.

In addition to the mandatory parameters required, you can also use the following optional parameters:

- `ui.folder`

Specifies the target folder. Acceptable values are the Content Manager search path or store ID.

- `ui.backURL`

Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter. For more information see, [“URL Validation” on page 1434](#).

Parameterized URL Examples

This section provides an example when starting IBM Cognos Event Studio using the parameterized URL method.

- Opening an agent in Event Studio

```
http://localhost/bi/v1/disp?b_action=xts.run&m=portal/launch.xts&ui.tool=EventStudio&ui.object=/content/folder[@name='Samples']/folder[@name='Models']/package[@name='GOSales (query)']/folder[@name='Event Studio Samples']/agentDefinition[@name='ELMEscalation Agent']&ui.action=run
```

cognosLaunch Examples

This section provides examples for performing specific functions when starting IBM Cognos Event Studio using the CognosLaunch method.

- Starting Event Studio

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','EventStudio','ui.action','new','ui.object','/content/package[@name='GOSales and Retailers']')
```

- Opening an agent in Event Studio

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','EventStudio','ui.action','edit','ui.object','/content/folder[@name='Samples']/folder[@name='Models']/package[@name='GO Sales(query)']/folder[@name='Event Studio Samples']/agentDefinition[@name='ELMEscalation Agent']')
```

Starting IBM Cognos Viewer

You can use a URL to quickly open a specified report in IBM Cognos Viewer.

Use the following parameters to start IBM Cognos Viewer with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=cognosViewer&ui.object=/content/folder[@name='Samples']/folder[@name='Cubes']/package[@name='Salesand Marketing (cube)']/folder[@name='Reporting Report Samples']/report[@name='Actualvs. Planned Revenue']&ui.action=run
```

Use the following parameters to start IBM Cognos Viewer with the cognosLaunch method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','CognosViewer')
```

The following parameters are optional for IBM Cognos Viewer:

- `run.outputFormat`

Specifies the output format. Acceptable values are CSV, HTML, PDF, XHTML, XLWA, and XML. The default value is HTML.

To run reports in the CVS, PDF, or XML output formats, you require the generate output capability for the specific format. URLs that hard code a format in a run request can result in an error if the user is restricted from generating output in that format.

Note: To protect IBM Cognos Analytics and your servers, the IBM Cognos Application Firewall (CAF) rejects URL-based requests for output in XLS format.

- `run.outputLocale`

Specifies the output language. Acceptable values are expressed as a hyphenated language-region pair, in accordance with the RFC3066 standard. The default value is `en-us`.

- `run.prompt`

Specifies whether the report service issues prompts, so you can enter report option values. Acceptable values are `true` and `false`. The default value is `true`.

- `run.xslURL`

Specifies the location of an XSL stylesheet that can be applied to the report, thereby rendering it in the requested format. An acceptable value is a URI.

When referencing the XSL file, ensure that the specified file is valid, and exists on the application server in the `templates/rsvp/xsl` directory. Referencing an invalid stylesheet terminates the process.

- `run.outputEncapsulation`

Specifies how output documents in the response are encapsulated. Acceptable values are `HTML`, `URL`, `URLQueryString`, `none`.

- `asynch.attachmentEncoding`

Specifies how attachments to the response are encoded. Acceptable values are `base64`, `MIME`, `MIMECompressed`. The default value is `base64`.

- `asynch.primaryWaitThreshold`

Specifies the maximum amount of time, in seconds, that the server can use to process the request before sending a response to the client. Acceptable values are any integer. Use a value of 0 to make the client wait indefinitely. The default value is 7.

- `asynch.secondaryWaitThreshold`

Specifies the maximum amount of time, in seconds, that the server can use to process the request before sending a response to the client. Acceptable values are any integer. Use a value of 0 to make the client wait indefinitely. The default value is 30.

- `ui.object`

Specifies the path of the target object. Acceptable values are the Content Manager search path or store ID.

- `ui.folder`

Specifies the target folder. Acceptable values are the Content Manager search path or store ID.

- `ui.backURL`

Specifies the URI to open after you close the selected component.

IBM Cognos Application Firewall validation is enforced on URLs that contain this parameter.

Parameterized URL Examples

This section provides examples for performing specific functions when viewing reports using the parameterized URL method.

- Viewing saved reports

```
http://localhost/bi/v1/disp?b_action=cognosViewer&ui.  
object=/content/folder[@name='Samples']/folder  
[@name='Models']/package[@name='G0Sales (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005SalesSummary']&ui.action=view
```

- Running live reports

```
http://localhost/bi/v1/disp?b_action=cognosViewer&ui.  
object=/content/folder[@name='Samples']/folder[@name='Models']/package  
[@name='G0Sales (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005SalesSummary']&ui.action=run
```

- Viewing reports in different output modes

```
http://localhost/bi/v1/disp?b_action=cognosViewer&ui.  
object=/content/folder[@name='Samples']/folder[@name='Models']/package  
[@name='G0Sales (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005SalesSummary']&ui.  
action=run&run.outputFormat=PDF
```

cognosLaunch Examples

This section provides examples for performing specific functions when viewing reports using the CognosLaunch method.

- Viewing saved reports

```
cognosLaunch  
( 'ui.gateway', 'http://localhost/ibmcognos/bi/v1/disp', 'ui.  
tool', 'CognosViewer', 'ui.action', 'view', 'ui.object', 'defaultOutput  
(/content/folder  
[@name='Samples']/folder[@name='Models']/package  
[@name='G0Sales (analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005Sales Summary'] )')
```

- Running live reports

```
cognosLaunch  
( 'ui.gateway', 'http://localhost/ibmcognos/bi/v1/disp', 'ui.  
tool', 'CognosViewer', 'ui.action', 'run', 'ui.object', '/content/folder  
[@name='Samples']/folder[@name='Models']/package  
[@name='G0 Sales(analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005Sales Summary'] )')
```

- Viewing reports in different output modes

```
cognosLaunch  
( 'ui.gateway', 'http://localhost/ibmcognos/bi/v1/disp', 'ui.  
tool', 'CognosViewer', 'ui.action', 'run', 'ui.object', '/content/folder  
[@name='Samples']/folder  
[@name='Models']/package  
[@name='G0 Sales(analysis)']/folder  
[@name='Reporting Report Samples']/report  
[@name='2005Sales Summary'] , 'run.outputFormat', 'PDF')
```

Starting PowerPlay Studio

You can use a URL to launch a package or report in IBM Cognos PowerPlay Studio.

For more information about other options for working with URLs, see the IBM Cognos *Administration and Security Guide*.

Use the following parameters to start PowerPlay Studio with the parameterized URL method:

```
http://localhost/bi/v1/disp?b_action=powerPlayService
```

Use the following parameters to start PowerPlay Studio with the cognosLaunch method:

```
cognosLaunch('ui.gateway','http://localhost/ibmcognos/bi/v1/disp','ui.tool','powerPlayService')
```

The `b.action`, `TARGET`, and `ui.action` parameters are mandatory.

- The `b_action` parameter directs the incoming requests to the appropriate service.
- The `TARGET` parameter identifies the name and location of the report object. The `TARGET` parameter is usually the search path of the report object being used.
- The `ui.action` parameter specifies the action to take. The acceptable values are `run` and `edit`.

The `FORMAT`, `FILTER`, and `PROMPT` parameters are optional.

- The `FORMAT` parameter specifies the output format. Acceptable values are `PDF` and `HTML`.
- The `FILTER` parameter specifies the context of the package or report. The parameter requires a URL-encoded value of `Dimension Code1<tab>PPDSID Code<tab>Dimension Code2<tab>PPDSID Code`. You can only use one `PPDSID Code` per `Dimension Code`. You cannot pass multiple values to the same dimension.
- The `PROMPT` parameter specifies whether the client report with dimension line filtering displays the dimension line filtering page. This setting only applies to published reports in `PDF` format with dimension line filtering enabled.

Parameterized URL Examples

This section provides an example when starting IBM Cognos PowerPlay Studio using the parameterized URL method.

- Using PowerPlay Studio to open a specific report object

```
http://localhost/bi/v1/disp?b_action=powerPlayService&
TARGET=/content/folder
[@name='Samples']/folder[@name='Cubes']/package
[@name='Salesand Marketing (cube)']/folder
[@name='PowerPlay Studio Report Samples']/powerPlay8Report
[@name='Revenueby Order Method']
```

- Opening a report in PowerPlay Studio using filtering

```
http://localhost/bi/v1/disp?b_action=powerPlayService&
TARGET=/content/folder
[@name='Samples']/folder[@name='Cubes']/package
[@name='Salesand Marketing (cube)']/folder
[@name='PowerPlay Studio Report Samples']/powerPlay8Report
[@name='Revenueby Order Method']&FILTER=
Order%20method%09601%09MEASURES%09Returns
```

Starting IBM Cognos Analytics Components in a Specified Browser Window

This feature allows you to start an IBM Cognos Analytics component in a named browser window.

To do this, you must use the following parameters with this syntax:

```
cognosLaunchInWindow(windowName, windowProperties, "ui.gateway", "
gateway", "ui.tool", "component")
```

The `windowName` and `windowProperties` parameters represent the values specific to starting an IBM Cognos Analytics component in a named browser window.

The `windowName` string is the name of the browser window, frame, or `iframe` in which to start the specified component. If the specified name does not exist, a newly created pop-up browser window appears with the name. To create a new pop-up window, use `"_blank"` as the value.

The `windowProperties` string defines the properties applied to the new pop-up window. This only applies to newly created windows. The available values depend on the type of the Web browser you are using, and correspond to the values supported by the JavaScript `window.open()` function. Some options may not work in all browsers.

The `windowProperties` parameter consists of a comma-separated list. Each item consists of an option and a value, separated by the equals sign (`=`). For example, `"fullscreen=yes, toolbar=yes"`. Some common examples include:

- `channelmode`
Specifies whether to display the window in theater mode, and show the channel band. The default is `no`. Acceptable values are `yes` and `no`.
- `directories`
Specifies whether to add directory buttons. The default is `yes`. Acceptable values are `yes` and `no`.
- `fullscreen`
Specifies whether to display the browser in full-screen mode. This mode hides the browser's title bar and menus. A window in full-screen mode must also be in `channelmode`. The default is `no`. Acceptable values are `yes` and `no`.
- `height`
This integer specifies the height of the window, in pixels. The minimum value is `100`.
- `left`
This integer specifies the left position, in pixels, relative to the upper-left corner of the screen.
- `location`
Specifies whether to display the input field for entering URLs directly into the browser. The default is `yes`. Acceptable values are `yes` and `no`.
- `menubar`
Specifies whether to display the menu bar. The default is `yes`. Acceptable values are `yes` and `no`.
- `resizable`
Specifies whether to display resize handles at the corners of the window. The default is `yes`. Acceptable values are `yes` and `no`.
- `scrollbars`
Specifies whether to display horizontal and vertical scroll bars. The default is `yes`. Acceptable values are `yes` and `no`.
- `status`
Specifies whether to display a status bar at the bottom of the window. The default is `yes`. Acceptable values are `yes` and `no`.
- `titlebar`
Specifies whether to display a title bar for the window. This parameter is only valid if the calling application is an HTML Application, or a trusted dialog box. The default is `yes`. Acceptable values are `yes` and `no`.
- `toolbar`
Specifies whether to display the browser toolbar. The default is `yes`. Acceptable values are `yes` and `no`.
- `top`

This integer specifies the top position, in pixels. This value is relative to the upper-left corner of the screen.

- `width`

This sets the width of the window, in pixels. The minimum value is 100.

The `ui.gateway`, `ui.tool`, along with some additional parameter values are described in [“Start Parameters”](#) on page 1435.

Chapter 21. URL API reference

The URL API provides a quick and efficient way to start IBM Cognos components and open specified content, such as reports, metrics, folders, or pages.

IBM Cognos Viewer

There are two methods available to access IBM Cognos Viewer:

1. Parameterized URL
2. CognosLaunch

To access IBM Cognos Viewer using the parameterized URL method, use the following:

```
http://server/ibmcognos/bi/v1/disp?b_action=cognosViewer
```

To access IBM Cognos Viewer using the CognosLaunch method, use the following:

```
cognosLaunch('ui.gateway', 'http://server/ibmcognos/bi/v1/disp', 'ui.tool', 'CognosViewer')
```

New in Version 10.2.2 – “Documentation Updates” on page 1827

The `executionParameters` parameter was removed as it is not currently used.

asynch.<option>

The following [asynchOptionEnum](#) options are supported:

- [alwaysIncludePrimaryRequest](#)
- [attachmentEncoding](#)
- [primaryWaitThreshold](#)
- [secondaryWaitThreshold](#)

The following [asynchOptionEnum](#) options are not supported:

- [continueOnError](#)
- [historyLocation](#)
- [restartHistoryLocation](#)
- [runContext](#)
- [skipTaskHistoryLocations](#)

Type: [asynchOptionEnum](#)

run.<option>

The following [runOptionEnum](#) options are supported:

- [burst](#)
- [continueConversation](#)
- [credentialParameters](#)
- [cssURL](#)
- [data](#)
- [layoutParameters](#)
- [maximumValueCount](#)
- [metadataModel](#)

- outputEncapsulation

- outputFormat

IBM Cognos Viewer uses the output format for the object in the following order of precedence:

- The output format specified in the object in the content store.
- The specified user preference output format.
- The system default (HTML).

The following output format values are allowed when the report is run:

- CSV
- HTML
- PDF
- singleXLS
- spreadsheetML
- XLS
- xlsxData
- XLWA
- XML

The following output format values are allowed when viewing saved output:

- CSV
- HTML
- MHT
- PDF
- singleXLS
- spreadsheetML
- XHTML
- XLS
- XLWA
- XML

- outputLocale

- outputPageDefinition

- outputPageOrientation

- print

- printer

- printer

- printerAddress

- prompt

- promptFormat

- saveAs

To use the saveAs option, specify the required values using the following three URL parameters:

- `run.saveAsObjectClass=<objectClass>`
- `run.saveAsObjectName=<objectName>`
- `run.saveAsParentSearchPath=<parentSearchPath>`

- selectionBasedFeatures

- [skipValueCount](#)
- [verticalElements](#)
- [xslURL](#)

The following [runOptionEnum](#) options are not supported:

- [advancedOutput](#)
- [archive](#)
- [email](#)
- [emailAsAttachment](#)
- [emailAsURL](#)
- [mobile](#)
- [savePromptCache](#)
- [secondaryWaitThreshold](#)
- [xslParameters](#)

Note: The Microsoft® Excel (.XLS) files are not supported as an output format when using the URL-based API.

Type: [runOptionEnum](#)

cv.contextinfo

Specifies whether context information is generated in the report output.

Type: `xs:boolean`

Default: `true`

cv.contextmenu

Specifies whether the context menu is active.

Type: `xs:boolean`

Default: `true`

cv.debug

Specifies whether IBM Cognos Viewer logging is enabled.

Type: `xs:boolean`

Default: `false`

cv.drill

Specifies whether drill functionality is active.

Type: `xs:boolean`

Default: `true`

cv.header

Specifies whether to display the header.

Type: `xs:boolean`

Default: `true`

cv.navlinks

Specifies whether the navigation links are active.

Type: xs:boolean

Default: true

cv.selection

Specifies whether the selection functionality appears.

Type: xs:boolean

Default: true

cv.toolbar

Specifies whether to display the toolbar.

Type: xs:boolean

Default: true

p_<parameterName>

Specifies values for the parameter named parameterName.

Use an XML structure to specify multiple values for a parameter.

Type: xs:string

parameterValues

Specifies parameter values.

The parameter values are encoded in the format defined by the IBM Cognos Software Development Kit.

Type: xs:string

ui.action

Specifies the action that the server will perform.

Type: xs:string

Default: objectDefault

Values:

- objectDefault
- run
- view

ui.backURL

Specifies the URL to use for the return link in the IBM Cognos Viewer header.

Type: xs:anyURI

ui.conversation

Specifies state information used by the server.

IBM Cognos Viewer provides the value for this parameter in responses.

This value must be used in subsequent requests to take advantage of IBM Cognos query re-use and failover capabilities.

This string is encoded by IBM Cognos Application Firewall.

Type: xs:string

ui.errURL

Specifies the URL to use if an error occurs while attempting to load a report.

Type: xs:anyURI

ui.folder

Specifies the startup folder.

This parameter is used by the Save As dialog.

Type: xs:string

ui.format

See the run.<option> (outputFormat) section.

Type: xs:string

ui.gateway

Specifies the location of the IBM Cognos gateway used by the IBM Cognos component identified by `ui.tool`.

Type: xs:anyURI

ui.name

Specifies the name displayed in the title bar and IBM Cognos Viewer header.

Type: xs:string

ui.object

Specifies the content store location of the object to be run.

This parameter is required if `ui.action` is `run`.

Type: xs:string

ui.primaryAction

Specifies the action that started the current wait loop.

The value for this parameter is provided in responses from IBM Cognos Viewer, and must be used in subsequent requests.

Values:

- `objectDefault`
- `run`
- `view`

Type: xs:string

Default: `objectDefault`

ui.productLocale

Specifies the product locale to use for the report.

Type: `xs:locale`

ui.routingServerGroup

Specifies the name of the server group for the request.

Type: `xs:string`

ui.spec

Specifies a specification to run.

This parameter is required if `ui.action` is `runSpecification`.

Type: `xs:string`

ui.tool

Specifies the IBM Cognos component to be invoked.

To launch IBM Cognos Viewer, use the value `CognosViewer`.

Type: `xs:string`

Chapter 22. Passing environment variables to external applications

You can use a URL to pass environment variables, such as content locale, to external applications. For example, if a user clicks a link that drills through to an external application, the content locale environment variable can be passed through the URL so that the language the user sees in the report is the same as the language the user sees in the application.

The mandatory building blocks for a URL that passes environment variables to an external application are as follows:

- `b_action=xts.run&m=portal/bridge.xts`, parameters required to pass the values of the environment variables
- `&c_env=<variables.xml>`, to specify the location of the XML file that identifies the environmental variables to be passed.

This file must be located under the templates folder in `templates/ps`.

- `&c_mode=<get_or_post>`, to specify whether to use the GET or POST request method when calling the external application. If this parameter is not specified, the default method is GET.

When you use the post request method, the HTML page created is UTF-8 encoded. When you use the GET request method, the values of the variables are UTF-8 encoded.

- the `<c_cmd>` attribute specified at the root level of the `templates/ps/portal/<variables.xml>` file. The URL specified can include application variables or other environment variables, to pass to the external application.

Types of Environment Variables

The following table lists a sample of environment variables that can be passed from IBM Cognos to an external application.

Variable Type	Variable Name	Description
Cookies		IBM Cognos supports the passing of <i>cognos</i> cookies.
User Preferences	<code>productLocale</code>	Use this variable to pass the locale used by the product.
	<code>contentLocale</code>	Use this variable to pass the locale used to view or process reports.
	<code>timeZoneID</code>	Use this variable to pass the time zone used by the selected geographical region.
	<code>format</code>	Use this variable to pass the user's preferred format to view the report.
Portal State	<code>view</code>	Use this variable to pass the view to either list view or details view.
	<code>cols</code>	Use this variable to pass the maximum number of columns per row for the detail view.

Table 235. Sample variables that can be passed to an external application (continued)

Variable Type	Variable Name	Description
	lines	Use this variable to pass the maximum number of rows that appear in a list before scroll bars are required. This only applies to rows where scrolling is allowed.
	home	Use this variable to pass the user's preferred view for the portal home page.
User Properties from Current Account Object	userName	Use this variable to pass the user name, as recorded in the account in the security system. For example, this variable specifies the user name as stored in an NTLM or LDAP namespace.
	surname	Use this variable to pass the surname, as recorded in the account in the security system. For example, this variable specifies the surname as stored in an NTLM or LDAP namespace.
	givenName	Use this variable to pass the given name, as recorded in the account in the security system. For example, this variable specifies the given name as stored in an NTLM or LDAP namespace.
	email	Use this variable to pass the email address, as recorded in the account in the security system. For example, this variable specifies the email address as stored in an NTLM or LDAP namespace.

Example - Passing Environment Variables

Use the following URL if you want to use a POST request to pass the environment variables specified in the XML file `templates/ps/<variables.xml>` and application variables named `var1` and `var2` to the application `http://server_name/myapp.cgi`:

```
http://localhost/ibmcognos/bi/v1/disp?b_action=xts.run&m=portal/bridge.xts&c_env=/portal/<variables.xml>&c_mode=post&var1=z&var2=y
```

Specify the Names of the Environment Variables to be Passed

You specify the names of the environment variables that you want to pass to the external application in an XML file.

The root element in this XML file is `CRNenv`. The `CRNenv` element contains an element for each of the types of environment variables:

- The `cookies` element specifies the cookie variables.
- The `cc_state` element specifies the portal state variables.
- The `user_pref` element specifies the user preferences variables.
- The `cm` element specifies the user property variables.
- The `url` element specifies allowable URLs, with pattern matching support through the "is-regex" attribute.

For more information, see [“Types of Environment Variables”](#) on page 1453.

Sample Environment Variables File

These elements are shown in the following example:

```
<CRNenv c_cmd="http://appServer/myapp.asp">
  <urls>
    <url is-regex="true">http://myServer[A-Za-z0-9]*/myaspx.aspx</url>
    <url>http://my.example.com/cgi-bin/mycgi.cgi</url>
  </urls>
  <cookies>
    <param name="CRN" />
  </cookies>
  <cc_state>
    <param name="home" />
  </cc_state>
  <user_pref>
    <param name="productLocale" />
  </user_pref>
  <cm>
    <param name="userName" />
    <param name="surname" />
  </cm>
</CRNenv>
```

Chapter 23. Using report specifications

A report specification is an XML document that defines the structure and layout of a report. A report specification is stored as a string in the `specification` property of a report object. You can create and format reports programmatically by creating an IBM Cognos Software Development Kit application that edits report specifications.

In order to effectively author a report using a report specification, you must first understand the concepts and functionality of Reporting. For more information about report authoring, see the IBM Cognos Reporting *User Guide*.

All of the sample XML in this section was created using Reporting. If there is functionality that you would like to implement through a report specification, we recommend that you first create your report in Reporting and view the report specification generated by IBM Cognos software. For more information, see [“Use Reporting to View Report Specification XML”](#) on page 1460.

To understand the basic elements in a report specification, see [“Basic Structure of a Report Specification”](#) on page 1461. To see a list report sample and for information about the XML that defines list reports, see [“List Reports”](#) on page 1465. To see a crosstab report sample, see [“Sample of a Crosstab Report”](#) on page 1474.

In addition, a variety of report samples are included with IBM Cognos software. To see how elements or features are used, you can examine the report specifications for these samples. For more information, see the Reporting *User Guide*.

The following sections contain additional information about the report specification XML that defines reports.

- [“Report Formatting”](#) on page 1478
- [“Report Variables”](#) on page 1480
- [“Optimize Query Performance”](#) on page 1482
- [“Summarize Data”](#) on page 1486
- [“Ways of Filtering Data ”](#) on page 1490

Use the SDK to Modify Report Specifications

You can modify the report specification using the SDK to

- programmatically apply the same change to many reports
- automate report creation using an SDK application

For example, you can change a report specification to add a column of data, or to apply conditional formatting based on certain values for multiple reports. You can automate a change to all reports, such as updating the company logo. You can also modify report specifications to change the report query, style, layout, model connection, prompts, distribution, or delivery.

We recommend that you run one of the samples that demonstrates how to modify a report specification. For more information, see *Code Samples and Language-specific Coding Practices* in the IBM Cognos Software Development Kit Developer Guide.

Retrieve a Report Specification

Use the following steps to retrieve a report specification from the content store.

Procedure

1. Use the `query(objectPath, parameterValues, options)` method of the report service.
2. Supply the search path of the desired report in the `objectPath` parameter.

For example, the **Order Year List Report** is located in the SDK Report Samples folder of the Go Data Warehouse (query) package.

To retrieve properties for this report, create an instance of `searchPathSingleObject`, and set its `value` property to the following search path: `/content/package[@name='GO Data Warehouse (query)']/folder[@name='SDK Report Samples']/report[@name='Order Year List Report']`

3. Use the `searchPathSingleObject` instance you created as the `objectPath` parameter in the `query(objectPath, parameterValues, options)` method.
4. You must also add the `specification` property to a `properties` option, and include that option in the `options` parameter. For more information about the `properties` option, see the `ContentManagerQueryOptionEnum` enumeration set.

Modify a Report Specification

Use the following steps to modify a report specification.

Procedure

1. You can modify the report specification in a number of ways, including
 - using an XML parser such as a Document Object Model (DOM) parser with your programming language to modify the report specification, or
 - saving the report specification as an XML file, and using an XML editor to make your changes.

For information about how to use IBM Cognos Reporting to determine what changes you want to make to your report specification, see [“Use Reporting to View Report Specification XML” on page 1460](#).

2. If you want to validate your XML file before you add it to the content store, use the `validateSpecification (specification, parameterValues, options)` method. This method validates the XML against the schema and also validates the internal logic of the XML and the expressions in the report specification. Expressions define filters, calculations and data items.

If you use this method, your report specification must validate against the latest version of the report specification schema. For information about validating report specifications in an XML editor using the IBM Cognos DTD or IBM Cognos schema, see [“Report Specification Validation” on page 1459](#)

Save the Report Specification to the Content Store

Procedure

1. Do one of the following.
 - If you want to overwrite an existing report in the content store with the modified report specification, use the `update (object, options)` method.
 - If you want to add a new report containing the modified report specification to the content store, use the `add (parentPath, object, options)` method. Ensure that the search path you use for the new report is unique, so that you do not overwrite the original report.

Both of these methods automatically validate the report specification against the schema before adding it to the content store.

2. To verify the changes to the report specification in the content store, use the `run (objectPath, parameterValues, options)` method of the report service to run the report. Check that the report results meet your reporting requirement.

Report Specification Validation

The report specifications that define your reports must validate against the DTD, `V5_report.dtd`, or schema, `V5_report_one.xsd`, provided with IBM Cognos software.

The IBM Cognos installation includes the schemas and DTDs for multiple product releases. We recommend that you use the latest version and that you upgrade all of your SDK applications to use this version of the schemas or DTD.

The namespace in the report specification specifies the schema or DTD version. For example:

```
<report xmlns="http://developer.cognos.com/schemas/report/n.0/"  
expressionLocale="en-us">
```

where `n.0` indicates the version.

Product Release	DTD or Schema Version
ReportNet 1.0	1.0
IBM Cognos 8 Version 8.1	2.0
IBM Cognos 8 Version 8.2	3.0
IBM Cognos 8 Version 8.3	4.0
IBM Cognos 8 Version 8.4	6.0
IBM Cognos Version 10.1.0	7.0
IBM Cognos Version 10.1.1	8.0
IBM Cognos Version 10.2.0	9.0
IBM Cognos Version 10.2.1	10.0
IBM Cognos Version 10.2.1.1	11.0
IBM Cognos Version 10.2.2	12.0

Note: If you use an external XML parser to validate your report specifications, the parser cannot validate the expressions. A report specification with valid XML syntax can still cause errors at runtime due to problems with expression syntax and internal logic. If you want to validate everything in a report specification, you must use the `validateSpecification (specification, parameterValues, options)` method.

If the external XML parser	use
supports schemas but does not support multiple schema files	<code>V5_report_one.xsd</code> to validate report specifications
supports multiple schemas	<code>V5_report.xsd</code> to validate report specifications
does not support schemas	<code>V5_report.dtd</code> to validate report specifications

The DTDs and schemas are located in `installation_location/webcontent/schemas/rspec/n.o`, where `n.o` indicates the schema or DTD version.

Recommendation - Avoid Using Older Versions of the DTD or Schemas

If your SDK application creates or modifies report specifications that validate against older versions of the schemas or DTD, you must add the reports to the content store with the `add (parentPath, objects,`

options) method from the Content Manager service. You can not use report service methods with older versions because these methods use the latest version of the DTD to validate specifications.

We do not recommend using older versions of the DTD or schemas because the report server must automatically upgrade your reports every time they run. This decreases performance and can potentially cause problems with the report during the automatic upgrade. The upgraded specification is not saved back to the content store.

Use Reporting to View Report Specification XML

Every IBM Cognos report has a report specification that defines the contents and layout of the report. Most of the elements in the report specification schema map directly to an object or property in the Reporting interface. We recommend that you use Reporting to familiarize yourself with the elements and attributes that you can use to author reports.

If you want to use the SDK to make the same change programmatically across many reports, you can use Reporting to determine which parts of the report specifications must be changed. After you have determined the required changes to the XML, you can use the SDK to apply the changes to many report specifications.

Procedure

1. In Reporting, open the report you want to edit.

Tip: If the report specification XML document that you want to edit is open, you can open the document directly into Reporting. Select the entire contents of the report specification and copy the contents to your clipboard. From the **Tools** menu, click **Open Report from Clipboard** to open the report.

2. Make the desired changes to your report in Reporting.
3. From the **Tools** menu, click **Show Specification** to view the changes in the report specification XML.
4. To copy the entire contents of the specification to your clipboard, from the **Tools** menu, click **Copy Report to Clipboard**.
5. If you want to view the XML for a single object in your report, right-click the object in Reporting and click **Copy**. Paste the clipboard contents into an XML editor.

For example, the following is the report specification XML for a column in a list report. If you want to add this XML to a report specification, you must delete the <RSClipboardFragment> elements.

```
<RSClipboardFragment version="2.0">
  <listColumn>
    <listColumnTitle>
      <contents>
        <textItem>
          <dataSource>
            <dataItemLabel refDataItem="Product name"/>
          </dataSource>
        </textItem>
      </contents>
      <style>
        <defaultStyles>
          <defaultStyle refStyle="lt"/>
        </defaultStyles>
      </style>
    </listColumnTitle>
    <listColumnBody>
      <contents>
        <textItem>
          <dataSource>
            <dataItemValue refDataItem="Product name"/>
          </dataSource>
        </textItem>
      </contents>
      <style>
        <defaultStyles>
          <defaultStyle refStyle="lc"/>
        </defaultStyles>
      </style>
    </listColumnBody>
  </listColumn>
</RSClipboardFragment>
```

```
</listColumnBody>  
</listColumn>  
</RSClipboardFragment>
```

Basic Structure of a Report Specification

The top level element in a report specification is the `report` element. The `report` element contains all of the elements that define the data for a report and how the data is presented. The `report` element has attributes that allow you to define the namespace and locale of the report.

The following is a graphical representation of the content model of a `report` element, from the `v5_report_one.xsd` schema.



Figure 11. `report` object child elements

Tip: Most XML editors allow you to view a graphical representation of a schema. If you are new to authoring report specifications, look at this view of the `v5_report_one.xsd` schema to understand how the elements fit together to create a report.

The two main elements to consider are the `queries` and `layouts` elements. These two elements contain the fundamental parts of the report. The `queries` element contains all of the data items, filters, parameters and aggregate information for data in the report. The `layouts` element defines how the data is presented, including the page layout and the data container. In order to put a data item in a report, the data item must be defined in the `queries` element, and then referenced in the `layouts` element.

For more information, see the Report Specification Reference entries for `queries` and `layouts`. See also “[Queries Element](#)” on page 1462 and “[Layouts Element](#)” on page 1463.

The remaining child elements of the `report` element are described here.

modelPath

Specifies the path to the Framework Manager model of the data in the report.

classStyles

Defines the styles to be used locally in the report

burst

Specifies the recipients and delivery method of a bursted report.

reportVariables

Defines report variables that allow conditional rendering of various parts of the report. For more information, see “[Report Variables](#)” on page 1480.

drillBehavior

Defines the drill-through behavior in a report

XMLAttributes

Reserved. Do not add this element to a report specification.

upgradeInfo

Reserved. Do not add this element to a report specification.

namedConditionalStyles

This element is used to define conditional styles. For more information, see “[Conditional Formatting](#)” on page 1479.

drillSets Behavior

This element is used to link data items across queries for drill up or down behavior.

reportName

Contains the name of the report in the content store.

appProperties

Defines properties specific to an IBM Cognos Active Report. For more information, see the IBM Cognos Analytics - Reporting *User Guide*.

Queries Element

The `queries` element contains one or more `query` elements. There is one `query` element for each query in your report. In IBM Cognos Analytics - Reporting, each `query` element corresponds to a query in the Query Explorer. The following image shows two queries in Reporting.

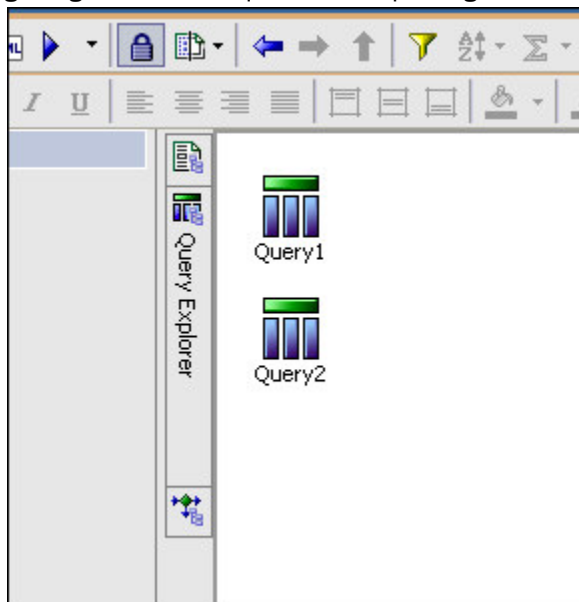


Figure 12. Two queries in Reporting

The queries can be represented in XML as follows.

```
<queries>
  <query name="Query1">
    <source>
      <model/>
    </source>
    <selection>
      <dataItem name="Order number" aggregate="none">
        <expression>[gosales_goretailers].[Orders].[Order number]</expression>
      </dataItem>
      <dataItem name="Order month" aggregate="none">
        <expression>[gosales_goretailers].[Orders].[Order month]</expression>
      </dataItem>
      <dataItem name="Order date" aggregate="none">
        <expression>[gosales_goretailers].[Orders].[Order date]</expression>
      </dataItem>
    </selection>
  </query>
  <query name="Query2">
    <source>
      <model/>
    </source>
    <selection>
      <dataItem name="Order number" aggregate="none">
        <expression>[gosales_goretailers].[Orders].[Order number]</expression>
      </dataItem>
      <dataItem name="Quantity" aggregate="total">
        <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
      </dataItem>
      <dataItem name="Country" aggregate="none">
        <expression>[gosales_goretailers].[Countries].[Country]</expression>
      </dataItem>
    </selection>
  </query>
</queries>
```



```
</query>
</queries>
```

In this example, the `queries` element contains two `query` elements, one for each query in the report: `<query name="Query1">` and `<query name="Query2">`. Each `query` element contains a `source` element that defines the data source for the data items. In this case, the data source is an IBM Cognos Framework Manager model. Each `query` also contains a `selection` element that contains a list of `dataItem` elements. In this simple example, the `dataItem` elements represent a column in the underlying database. If you want to create a more complex query, create the query in Reporting and view the XML. For more information, see [“Use Reporting to View Report Specification XML” on page 1460](#).

The following is a graphical representation of the content model of the `queries` element.

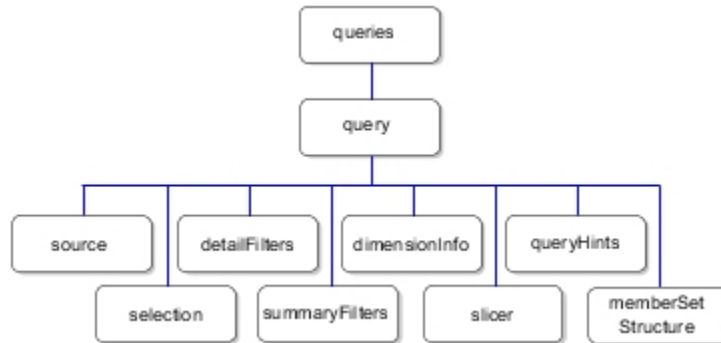


Figure 13. Content model of the `queries` element

To further refine your query, you can use any of the following child elements.

Query Child Element	Purpose
<code>detailFilters</code>	Filters detail rows of data. For more information, see “Ways of Filtering Data” on page 1490 .
<code>summaryFilters</code>	Filters aggregate data. For more information, see “Filter Measure Values in a Crosstab Report Sample” on page 1492 .
<code>dimensionInfo</code>	Models dimensional data.
<code>slicer</code>	Limits data sets. For more information, see “Ways of Filtering Data” on page 1490 .
<code>queryHints</code>	Optimizes query performance. For more information, see “Optimize Query Performance” on page 1482 .
<code>memberSetStructure</code>	Defines a hierarchy of nested member sets.

Layouts Element

The `layouts` element contains a set of pages that defines the appearance and formatting of your report. Although most reports contain only one layout, the `layouts` element can contain one or more `layout` child elements. Use more than one layout for a report when you want to present the report in different formats based on a variable. For example, you can define a different layout for each language in a multilingual report. For more information, see the IBM Cognos Analytics - Reporting User Guide.

Each layout contains a page (page element), and a series of pages, or a page set (pageSet element). The pages can be either report pages (`reportPages` element) or prompt pages (`promptPages` element). Report pages typically contain report data and prompt pages typically contain prompts and prompt

buttons for use when prompting is required. You can see the page structure of your report in Reporting using the Page Explorer, shown here.

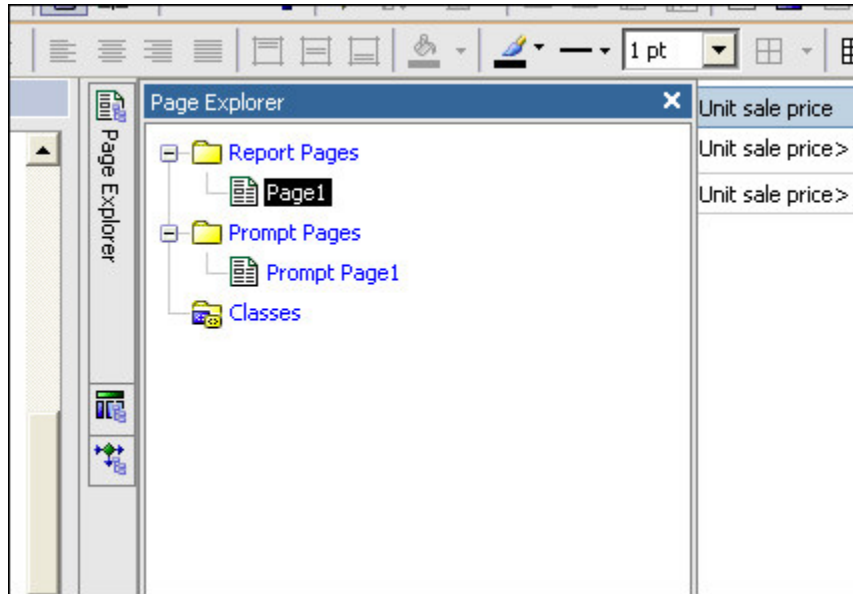


Figure 14. Page structure in Reporting

The XML for this report is as follows.

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
  ...
</queries>
<layouts>
<layout>
  <reportPages>
  <page name="Page1">
  <pageBody>
  <contents>
  <list name="List1" refQuery="Query1">
  ...
  </list>
  </contents>
  <style>
  <defaultStyles>
  <defaultStyle refStyle="pb"/>
  </defaultStyles>
  </style>
  </pageBody>
  <style>
  <defaultStyles>
  <defaultStyle refStyle="pg"/>
  </defaultStyles>
  </style>
  </page>
  </reportPages>
  <promptPages>
  <page name="Prompt Page1">
  ...
  </page>
  </promptPages>
  </layout>
</layouts>
</report>
```

Each page in a report must contain a body (pageBody element) and can also contain a page header (pageHeader element) and a page footer (pageFooter element). The data container is usually located in the pageBody. For a list of available data containers and their related elements, see “Types of Data Containers” on page 1465.

The report specification allows you to format layout objects in a variety of ways. For more information, see [“Report Formatting” on page 1478](#).

Types of Data Containers

The type of data that you want to present in your report determines which type of data container to use. The following table lists the types of reports with their corresponding elements from the report specification schema.

Table 239. Report types and corresponding report specification elements

Report type	Element(s)	Description
List	list	Renders data in a list.
Crosstab	crosstab	Renders data in columns and rows.
Chart	bubbleChart, combinationChart, gaugeChart, paretoChart, pieChart, polarChart, progressiveChart, radarChart, scatterChart, threeDCombinationChart, threeDScatterChart	Renders data in various graphical formats.
Repeater	repeater	Renders data in an unstructured format.
Repeater Table	repeaterTable	Renders repeated data in a table.
Map	mapChart	Renders data in a map image.

IBM Cognos Active Reports

An IBM Cognos Active Report is an extension of the traditional IBM Cognos report. You can leverage existing reports and upgrade them by adding interactive behavior.

You build this type of report with the same elements that you use to build other report types. Then, you include elements that are specific to IBM Cognos Active Reports.

For an IBM Cognos Active Report, the `application` attribute of the `report` element is set to `true`. This indicates that IBM Cognos Active Report controls and variables are valid within the report specification.

The `appProperties` element contains one or more elements to define overall report properties.

Controls specific to this type of report include buttons, button bars, tabs, decks, and cards. For many controls, two different versions are available. For static controls, the data that drives the controls is hard-coded in the rows and columns of an `appStaticDataTable` element. The `appButtonBar`, `appCard`, and `appListBox` elements are examples of static controls. For data driven controls, the data that drives the behavior is the data item specified by the `refDataItem` attribute of the specific control. The `appDataButtonBar`, `appDataCard`, and `appDataListBox` elements are examples of data driven controls.

For more information, see the IBM Cognos Reporting *User Guide*.

List Reports

List reports show data in rows and columns.

The [“Sample of a List Report” on page 1466](#) topic explains how to create a list report with a report specification. The following sections contain steps to modify a list report.

- [“Group Data in a Column” on page 1469](#)
- [“Set the Group Span for a Column” on page 1471](#)
- [“Add List Headers” on page 1472](#)
- [“Sort Data” on page 1473](#)

Sample of a List Report

This sample provides a list report showing the total quantity ordered for each product. To create this report, only two data items are required: Product name and Quantity.

The following is the output for the first part of the sample report.

Product name	Quantity
Aloe Relief	14,786
Bear Edge	28,140
Bear Survival Edge	25,392
Blue Steel Max Putter	3,188
Blue Steel Putter	8,816
BugShield Extreme	69,996
BugShield Lotion	51,126
BugShield Lotion Lite	54,928
BugShield Natural	47,360
BugShield Spray	46,664
Calamine Relief	18,362
Canyon Mule Carryall	19,022

Figure 15. Sample of a list report from Reporting

Required Report Elements

When you create any type of report, you must include the following elements and attributes.

- The `expressionLocale` attribute on the report element.
The default value for this attribute is obtained from your authoring locale.
- The `modelPath` element to identify the package and model to query.
The contents of this element are specified in search path format.
- The `queries` element to specify the data items to include in the report.
- The `layouts` element to specify how you want to render the results of the query.

For more information about the `queries` and `layouts` elements, see [“Basic Structure of a Report Specification” on page 1461](#).

Query Portion of the List Report Sample

The query portion of the report defines the data in the report.

For more information about the `queries` element, see [“Queries Element” on page 1462](#) and `queries`.

The following is the XML for the queries element in this list report sample.

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem aggregate="none" name="Product name">
          <expression>[gosales_goretailers].[Products].[Product name]</expression>
        </dataItem>
        <dataItem aggregate="total" name="Quantity">
          <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
        </dataItem>
      </selection>
    </query>
  </queries>
  <layouts>
    .
    .
  </layouts>
</report>
```

Notes About the Query Portion of the Report

- You must define the name attribute on the query element. The value of the name attribute must be unique. The query name is used to relate the layout portion of the report to the query portion.
- The query must contain a dataItem element for each data item.
- The dataItem elements must be unique within each query.

Use the name attribute to specify a name for each dataItem. This list report queries the data items Product name and Quantity.

- Add an expression element to the dataItem element.

In its simplest form, an expression is a reference to a query item in the model, and must appear in the format [namespace].[query_subject].[query_item].

- Add an aggregate attribute to each data item.

In this sample, Product name is not a column that you want to summarize, and so the aggregate attribute is set to none. Because you want to know the total quantity ordered for each product, the aggregate attribute for Quantity is set to total. For more information about aggregates, see [“Summarize Data” on page 1486](#).

- In this report, the values that appear under the Quantity column represent the total quantity ordered for each product.

This summarization occurs by default because the selection element has an autoSummary attribute that defaults to true.

Layout Portion of the List Report Sample

The layout portion of the report defines how the data appears in the report.

The following is the XML for the layout portion of the list report sample.

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    .
    .
  </queries>
  <layouts>
    <layout>
      <reportPages>
```

```

<page name="Page1">
  <pageBody>
    <contents>
      <list name="List1" refQuery="Query1">
        <style>
          <CSS value="border-collapse:collapse"/>
          <defaultStyles>
            <defaultStyle refStyle="ls"/>
          </defaultStyles>
        </style>
        <listColumns>
          <listColumn>
            <listColumnTitle>
              <contents>
                <textItem>
                  <dataSource>
                    <dataItemLabel refDataItem="Product name"/>
                  </dataSource>
                </textItem>
              </contents>
              <style>
                <defaultStyles>
                  <defaultStyle refStyle="lt"/>
                </defaultStyles>
              </style>
            </listColumnTitle>
            <listColumnBody>
              <contents>
                <textItem>
                  <dataSource>
                    <dataItemValue refDataItem="Product name"/>
                  </dataSource>
                </textItem>
              </contents>
              <style>
                <defaultStyles>
                  <defaultStyle refStyle="lc"/>
                </defaultStyles>
              </style>
            </listColumnBody>
          </listColumn>
          <listColumn>
            <listColumnTitle>
              <contents>
                <textItem>
                  <dataSource>
                    <dataItemLabel refDataItem="Quantity"/>
                  </dataSource>
                </textItem>
              </contents>
              <style>
                <defaultStyles>
                  <defaultStyle refStyle="lt"/>
                </defaultStyles>
              </style>
            </listColumnTitle>
            <listColumnBody>
              <contents>
                <textItem>
                  <dataSource>
                    <dataItemValue refDataItem="Quantity"/>
                  </dataSource>
                </textItem>
              </contents>
              <style>
                <defaultStyles>
                  <defaultStyle refStyle="lm"/>
                </defaultStyles>
              </style>
            </listColumnBody>
          </listColumn>
        </listColumns>
      </list>
    </contents>
    <style>
      <defaultStyles>
        <defaultStyle refStyle="pb"/>
      </defaultStyles>
    </style>
  </pageBody>
  <style>
    <defaultStyles>

```

```

        <defaultStyle refStyle="pg"/>
    </defaultStyles>
</style>
</page>
</reportPages>
</layout>
</layouts>
</report>

```

Notes About the Layout Elements in a List Report

- The `list` element defines the appearance of the list in the report.

Add the `refQuery` attribute to the `list` element to identify the query that contains the data items that you reference in the list. The value of the `refQuery` attribute must match the value of the `name` attribute of the query element.

- The `listColumn` element defines a column in the list. Each column contains a title and a body represented by the `listColumnTitle` and `listColumnBody` elements.
- The `contents` element defines the contents of a number of layout elements.
- The `textItem` element defines text in a report. For example, in this sample, this element is used in the `listColumnBody` and `listColumnTitle` elements.
- The `dataSource` element defines the source for the text in the `textItem` element. The `dataSource` element can have a `dataItemValue` or a `dataItemLabel` child element.

If you use the `dataItemValue` element, the text item contains values from the column of data represented by the data item. If you use the `dataItemLabel` element, the text item contains the label of the data item.

The value of the `refDataItem` attribute must match the value of the `name` attribute of a `dataItem` element from the query.

Group Data in a Column

You can group the data in one or more columns in a list or a repeater to create a hierarchy of levels. Creating more than one level in a hierarchy is known as creating nested groups.

Group a column when you want to group all instances of a value together and render the value only once.

When there is a related data item that you also want to appear only once, you can set the group span for this column. For more information, see [“Set the Group Span for a Column” on page 1471](#).

Steps to Group a Column in a List

Procedure

1. Add a `listGroups` and a `listGroup` element to the `list` element that contains the column you want to group.

Use the `refDataItem` attribute to specify the name of the data item to group.

2. Add a `listColumnRowSpan` element to the `listColumnBody` that you want to group.

Use the `refDataItem` attribute to specify the name of the data item.

Sample XML

```

<list name="List1" refQuery="Query1">
  <style>
    <CSS value="border-collapse:collapse"/>
    <defaultStyles>
      <defaultStyle refStyle="ls"/>
    </defaultStyles>
  </style>
  <listColumns>
    <listColumn>

```

```

<listColumnTitle>
  <contents>
    <textItem>
      <dataSource>
        <dataItemLabel refDataItem="Product type"/>
      </dataSource>
    </textItem>
  </contents>
</listColumnTitle>
<listColumnBody>
  <contents>
    <textItem>
      <dataSource>
        <dataItemValue refDataItem="Product type"/>
      </dataSource>
    </textItem>
  </contents>
  <listColumnRowSpan refDataItem="Product type"/>
</listColumnBody>
</listColumn>
...
</listColumns>
<listGroups>
  <listGroup refDataItem="Product type"/>
</listGroups>
</list>

```

Step to Group a Data Item in a Repeater or Repeater Table

Procedure

Add a `repeaterGroups` and `repeaterGroup` element to the `repeater` or `repeaterTable` element.

Use the `refDataItem` attribute to specify the name of the data item to group.

Sample XML

```

<repeaterTable across="2" down="3" refQuery="Query1"
name="Repeater Table1">
  ...
  <repeaterTableCell>
    <contents>
      ...
    </contents>
  </repeaterTableCell>
  <repeaterGroups>
    <repeaterGroup refDataItem="Product line"/>
  </repeaterGroups>
</repeaterTable>

```

Results

The value for each Product type appears in the first row, but is suppressed in subsequent rows as shown here.

Product type	Product name	Quantity	Unit sale price
Binoculars	Seeker 35	14,278	\$100.27
	Seeker 50	7,436	\$127.75
	Seeker Extreme	13,046	\$160.51
	Seeker Mini	8,570	\$78.86
Climbing Accessories	Firefly Charger	15,598	\$50.43
	Firefly Climbing Lamp	17,152	\$38.59
	Firefly Rechargeable Battery	22,840	\$7.45
	Granite Belay	19,894	\$65.31
	Granite Carabiner	43,130	\$3.72
	Granite Chalk Bag	11,072	\$16.48
	Granite Pulley	12,464	\$35.37

Figure 16. Grouped data item table

Set the Group Span for a Column

If there is a one-to-one relationship between a grouped column and another column, you can associate the two columns. This produces the same report output but requires less sorting of the query results.

For example, the Product type code and Product type data items have a one-to-one relationship. Each product type code corresponds to one product type. You can show these columns in a report so that each product type code and corresponding product type appear once, beside each other, spanning the detail rows. Instead of grouping both columns, you can group one column and then set the group span for the second column with the grouped column to achieve the same visual result while improving efficiency.

Do not set the group span for two or more columns if the data in the columns does not have a one-to-one relationship.

Before you begin

There must be at least one grouped column in the list or repeater before you can set the group span.

Procedure

1. Add a `listColumnRowSpan` element to the `listColumnBody` element that you want to span with a grouped column.
2. Set the `refDataItem` attribute value to the name of the grouped column.

Sample XML

```
<listColumnBody>
  <contents>
    <textItem>
      <dataSource>
        <dataItemValue refDataItem="Product type"/>
      </dataSource>
    </textItem>
  </contents>
  <listColumnRowSpan refDataItem="Product type code"/>
  <style>
    <defaultStyles>
      <defaultStyle refStyle="lc"/>
    </defaultStyles>
  </style>
</listColumnBody>
```

Results

In this sample, the product type code column is grouped and sets the control break. Each unique code appears only once, spanning the corresponding detail rows. Although control breaks also appear for the

Product Type column, the Product Type column is not grouped, it is spanned with the Product Type Code column. The resulting report is as follows.

Product type code	Product type	Product name	Quantity
1	Cooking Gear	TrailChef Deluxe Cook Set	3,744
		TrailChef Utensils	15,928
		TrailChef Cook Set	12,700
		TrailChef Double Flame	14,510
		TrailChef Water Bag	41,770
		TrailChef Kitchen Kit	25,964
		TrailChef Canteen	31,712
		TrailChef Single Flame	10,614
		TrailChef Kettle	9,900
		TrailChef Cup	31,834
2	Tents	Star Gazer 3	22,660
		Star Peg	44,686
		Star Gazer 6	10,504
		Star Dome	25,564
		Star Lite	8,988
		Star Gazer 2	18,262
3	Sleeping Bags	Hibernator Lite	23,868
		Hibernator Camp Cot	3,182
		Hibernator	23,984
		Hibernator Pad	9,460

Figure 17. Sample of a group span in a Reporting report

Add List Headers

You can add headers or footers to a report to show summaries or text.

To create a place for report totals, you can add

- overall header or overall footer.

The information contained in the header or footer appears only once for the entire report.

- list header or list footer.

The information contained in the header or footer appears once on each page, at the top or bottom of the list.

- group header or group footer.

Summary information for a given level appears in the header or footer of any grouped column.

Procedure

Add a `listPageHeader` element and its child elements to the `list` element. Use the `afterOverallHeader` attribute to specify whether you want the list page header to appear before or after the overall header.

Sample XML

```
<listPageHeader afterOverallHeader="true">
  <listRows>
    <listRow>
      <rowCells>
        <rowCell colSpan="4">
          <contents>
            <textItem>
              <dataSource>
                <staticValue>List Page Header</staticValue>
              </dataSource>
            </textItem>
          </contents>
        </rowCell>
      </rowCells>
    </listRow>
  </listRows>
</listPageHeader>
```

Results

The following image contains a list page header, a group footer and an overall header.

Product type	Product name	Quantity	Unit sale price
Summary			
List Page Header			
Binoculars	Seeker 35	14,278	\$100.27
	Seeker 50	7,436	\$127.75
	Seeker Extreme	13,046	\$160.51
	Seeker Mini	8,570	\$78.86
Binoculars			
Climbing Accessories	Firefly Charger	15,598	\$50.43
	Firefly Climbing Lamp	17,152	\$38.59
	Firefly Rechargeable Battery	22,840	\$7.45
	Granite Belay	19,894	\$65.31
	Granite Carabiner	43,130	\$3.72
	Granite Chalk Bag	11,072	\$16.48
	Granite Pulley	12,464	\$35.37
Climbing Accessories			

Figure 18. List headers in a report from Reporting

Sort Data

You can organize the data in a report by specifying a sort order. Data can be organized in ascending or descending sort order based on the values in any data item. You can sort the data in many different report elements. For a complete list of these elements, see `sortList`.

For information on sorting data that comes from a dimensional data source, see the *Reporting User Guide*.

Procedure

1. Add the `sortList` element to the `list` element.

2. Add a `sortItem` element to the `sortList` element. Use the `refDataItem` element to specify the data item to sort. Use the `sortOrder` attribute to specify whether to sort the data in ascending or descending order.

Sample XML

```
<list name="List1" refQuery="Query1">
  ...
  <listColumns>
    ...
  </listColumns>
  <sortList>
    <sortItem refDataItem="Revenue" sortOrder="descending"/>
  </sortList>
</list>
```

Sample of a Crosstab Report

A crosstab report provides a different view of data by showing values at the intersection points of rows and columns. In the **Create a Crosstab Report** sample, the returned items appear by Product line, and by Order method. There is also a hierarchy or level of nesting in the rows, showing the product types that are part of the product line.

The sample also shows overall totals for both the rows and columns of the crosstab.

Return quantity		Special	Web	Telephone	E-mail	Fax	Sales visit	Mail	Total(Order method)
Mountaineering Equipment	Climbing Accessories		114	350	254	72	138	4	932
	Tools	2	98	190	192	22	92		596
	Rope		26		2	6	20	2	56
	Safety		62	40	32		16		150
	Total(Product type)	2	300	580	480	100	266	6	1,734
Personal Accessories	Navigation	2	198	38	110	2	80		430
	Watches	2	122	170	52	14	140	2	502
	Binoculars		12	10	36		38	2	98
	Eyewear	8	138	46	72	14	54		332
	Knives	74	90	280	188	192	240	32	1,096
	Total(Product type)	86	560	544	458	222	552	36	2,458

Figure 19. Sample of a crosstab report from Reporting

Query Portion of the Report

The following is the query portion of the crosstab report sample. The names of the data items are shown in bold.

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
  /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem name="Product line" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product line]</expression>
        </dataItem>
        <dataItem name="Product type" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
        <dataItem name="Return quantity" aggregate="total">
          <expression>[gosales_goretailers].[Orders].[Return quantity]</expression>
        </dataItem>
        <dataItem name="Order method" aggregate="none">
          <expression>[gosales_goretailers].[Orders].[Order method]</expression>
        </dataItem>
      </selection>
    </query>
  </queries>
</report>
```

```

</dataItem>
<dataItem name="Total(Order method)">
  <expression>total(currentMeasure within set [Order method])</expression>
</dataItem>
<dataItem name="Total(Product type)">
  <expression>total(currentMeasure within set [Product type])</expression>
</dataItem>
</selection>
</query>
</queries>
...
</report>

```

The query portion of the report specification must contain all the data items that will appear in the crosstab report.

In this sample, the query contains six data items or dataItem elements. Four of the data items return data from the model. They are Product type, Product line, Order method, and Return quantity. The Product type and Product line appear in the crosstab rows. The Order method appears in the crosstab columns. The Return quantity appears at the intersection points of the rows and columns.

The last two data items are calculations that render totals in the report. For more information, see [“Summarize Data” on page 1486](#).

Layout Portion of the Crosstab Report

In the layout portion of the report specification for a crosstab report, you typically define the crosstab, the rows and columns or crosstab nodes, the corner, fact cell and default measure. The XML for these parts of a crosstab and the corresponding XML are described in the following sections.

Crosstab Element

To create a crosstab report, in the layout portion of the specification, you must add a crosstab element in the layout portion of the report specification. In this sample, the crosstab element is a child of the contents element, which is a child of the pageBody element. The following is the XML that shows where the crosstab element appears in this sample.

```

<layout>
  <reportPages>
    <page name="Page1">
      <pageBody>
        <contents>
          <crosstab name="Crosstab1" refQuery="Query1">

```

The crosstab element defines the grid in which the data is displayed. The grid is also known as a crosstab frame. The refQuery attribute on the crosstab element references the query that contains the data items in the crosstab. In this sample, the query is Query1.

Crosstab Rows and Columns

A crosstab report has two edges: rows and columns. The row edge is defined by the crosstabRows element and the column edge is defined by the crosstabColumns element. Both of these elements are child elements of the crosstab element.

The following is the XML that defines the rows and columns of this crosstab sample. The key elements are shown in bold.

```

<crosstabRows>
  <crosstabNode>
    <crosstabNodeMembers>
      <crosstabNodeMember refDataItem="Product line" edgeLocation="e1">
        <contents>
          <textItem>
            <dataSource>
              <memberCaption/>
            </dataSource>
          </textItem>
        </contents>

```

```

<style>
  <defaultStyles>
    <defaultStyle refStyle="m1"/>
  </defaultStyles>
</style>
</crosstabNodeMember>
</crosstabNodeMembers>
<crosstabNestedNodes>
  <crosstabNode>
    <crosstabNodeMembers>
      <crosstabNodeMember refDataItem="Product type" edgeLocation="e2">
        <contents>
          <textItem>
            <dataSource>
              <memberCaption/>
            </dataSource>
          </textItem>
        </contents>
        <style>
          <defaultStyles>
            <defaultStyle refStyle="m1"/>
          </defaultStyles>
        </style>
      </crosstabNodeMember>
    </crosstabNodeMembers>
  </crosstabNode>
  <crosstabNode>
    <crosstabNodeMembers>
      <crosstabNodeMember refDataItem="Total(Product type)"
        edgeLocation="e3">
        <contents>
          ...
          <factCell>
            ...
          </factCell>
          ...
        </crosstabNodeMember>
      </crosstabNodeMembers>
    </crosstabNode>
  </crosstabNestedNodes>
</crosstabNode>
</crosstabRows>
<crosstabColumns>
  <crosstabNode>
    <crosstabNodeMembers>
      <crosstabNodeMember refDataItem="Order method" edgeLocation="e4">
        <contents>
          ...
        </contents>
      </crosstabNodeMember>
    </crosstabNodeMembers>
  </crosstabNode>
  <crosstabNode>
    <crosstabNodeMembers>
      <crosstabNodeMember refDataItem="Total(Order method)" edgeLocation="e5">
        <contents>
          ...
        </contents>
      <factCell>
        ...
      </factCell>
      ...
    </crosstabNodeMember>
  </crosstabNodeMembers>
</crosstabNode>
</crosstabColumns>

```

The `crosstabRows` and `crosstabColumns` elements contain `crosstabNode` child elements. In this topic, the `crosstabNode` element is also referred to simply as a node.

A `crosstabNode` must contain a `crosstabNodeMembers` child element and at least one `crosstabNodeMember` grandchild element. Each `crosstabNode` represents a node or a level in the hierarchy. If you want to union two sets of data at the same level, add two `crosstabNode` elements or `crosstabNodeMember` grandchild elements in the same `crosstabNode` element. Multiple node members in the same node allow you to nest another level of data beneath the two members. In this sample, there are no `crosstabNode` elements with more than one `crosstabNodeMember` grandchild element.

In this sample, there are two nodes at the top level. The first node contains the Product line and the second node contains the total for the Product type. The first node also has a nested node, or a second level in the hierarchy, for the Product type.

Crosstab Corner, Fact Cell, and Default Measure

The `crosstab` element typically contains `crosstabCorner`, `crosstabFactCell` and `defaultMeasure` child elements. All `crosstab` elements require these three child elements. The following is the portion of the XML for these three elements. The key parts are shown in bold.

```
<crosstab refQuery="Query1">
  <style>
    ...
  </style>
  <crosstabColumns>
    ...
  </crosstabColumns>
  <crosstabRows>
    ...
  </crosstabRows>
```

```
<crosstabCorner>
  <contents>
    <textItem>
      <dataSource>
        <dataItemLabel refDataItem="Return quantity"/>
      </dataSource>
    </textItem>
  </contents>
  <style>
    <defaultStyles>
      <defaultStyle refStyle="xm"/>
    </defaultStyles>
  </style>
</crosstabCorner>
<defaultMeasure refDataItem="Return quantity"/>
<crosstabFactCell applySingleDefaultStyle="true">
  <contents>
    <textItem>
      <dataSource>
        <cellValue/>
      </dataSource>
    </textItem>
  </contents>
  <style>
    <defaultStyles>
      <defaultStyle refStyle="mv"/>
    </defaultStyles>
  </style>
</crosstabFactCell>
```

The `crosstabCorner` element defines the contents of the cell at the upper left corner of the crosstab. In this sample, the `crosstabCorner` contains the label for the Return quantity data item. The `refDataItem` attribute specifies the data item to reference.

The `crosstabFactCell` element defines the contents of the cells in the crosstab grid. These cells display measures at the intersection points of the rows and columns. In this sample, the child element of `dataSource` is `cellValue`. The `cellValue` element renders data in the report according to information on the edges and in the `defaultMeasure` element as follows.

- If one of the edges contains a data item that is a measure, the report grid contains the measure.
- If neither edge contains a data item that is a measure, the report grid contains the data item referenced in the `defaultMeasure` element.

The `defaultMeasure` child element uses the `refDataItem` attribute to specify data item to use if the data item on the edge is not a measure. In this sample, the default measure is Return quantity.

Report Formatting

The report specification allows you to format your report in a number of different ways. You can format report output using a style sheet, class styles, and conditional styles. Styles are defined in CSS format.

Default Style Sheet

IBM Cognos software uses a default style sheet called `GlobalReportStyles.css` for all reports. You can modify this file to change the default styles that apply to all reports. For more information about modifying this file, see the IBM Cognos Administration and Security Guide.

Procedure

1. Add the following elements as children of the element to inherit the style:

```
<style>
  <defaultStyles>
    <defaultStyle refStyle="m1"/>
  </defaultStyles>
</style>
```

2. Use the `refStyle` attribute to specify the name of the style in the `GlobalReportStyles.css` style sheet.

Use Local Class Styles to Define and Reuse Styles

The report specification allows you to create styles that only apply to the report in which they are defined. You can define styles within a report in the two following ways:

- To define the styling for a single object in a report, add a `style` element as a child of the element to inherit the style. Add a CSS element as a child of the `style` element and define the style in CSS format in the `value` attribute. For example:

```
<style><CSS value="border:1pt solid
black"/></style>
```

- To define a style and re-use it throughout the report, use the `classStyles` element.

The `classStyles` element is a child element of the `report` element. Use this element to define any styles that you want to use locally in a report. To define styles, you must add a `classStyle` child element for each style that you want to use.

Procedure

1. Add a `classStyles` element to the `report` element.
2. Add a `classStyle` child element for each of the styles that you want to reuse in the report.
Specify a unique value for the `name` attribute of each `classStyle` element.
3. Add a CSS child element to each `classStyle` element. Define the style in CSS format in the CSS element.
4. To reference a class style anywhere in the report, add the following elements as children of the element to inherit the style:

```
<style>
  <defaultStyles>
    <defaultStyle refStyle="m1"/>
  </defaultStyles>
</style>
```

Use the `refStyle` attribute to specify the name of the style you want to use. The value of the `refStyle` attribute must be the same as the `name` attribute of the `classStyle` element that defines the style.

Conditional Formatting

Conditional formatting allows you to highlight exceptional data in a report.

You can format parts of your report based on one or more string values, a report condition, or data values. To define conditional formatting, add a `namedConditionalStyles` element to your report specification.

To format based on	use this <code>namedConditionalStyles</code> child element
a string	<code>stringsConditionalStyle</code>
report expression	<code>advancedConditionalStyle</code>
a data item value or range of values	<code>rangeConditionalStyle</code>

For more report formatting information, see the IBM Cognos Analytics - Reporting *User Guide*.

Format Cells Based on Data

You can use conditional formatting to highlight ranges of numeric values in a list report.

Procedure

1. Add a `namedConditionalStyles` element to the report element.
2. Add a `rangeConditionalStyle` element to the `namedConditionalStyles` element.

Specify the type of data in the `type` attribute.

The type must match the data type in the data item used to determine the conditional styling.

The following is a graphical representation of the content model of the `rangeConditionalStyle` element.

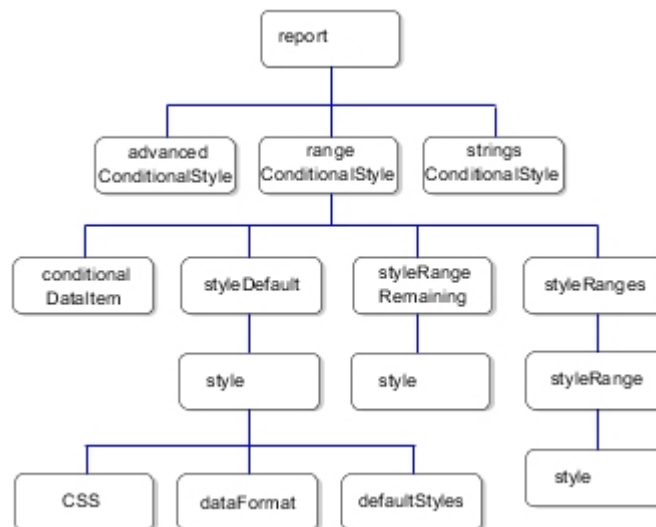


Figure 20. `rangeConditionalStyle` graphical representation

3. Add a `conditionalDataItem` element to the `rangeConditionalStyle` element. Use the `refQuery` and `refDataItem` attributes to specify the query and data item used to determine the conditional styling.
4. Add a `styleRanges` element to the `rangeConditionalStyle` element.
5. Add as many `styleRange` elements as required to define the ranges of data values for which to apply the conditional styling. Use the `value` attribute to specify the lower boundary of each range of values. You must order the `styleRange` elements from highest to lowest.
6. Add a `style` element to each `styleRange` element to define the styling to apply for that range.

7. Add a `styleRangeRemaining` element to define the styling to apply to all data values that fall outside of the defined ranges.

In this example, data values that are 12000 and higher are rendered with lime as the background color. All remaining values are rendered with a red background color.

```
<namedConditionalStyles>
  <rangeConditionalStyle name="Conditional Style 1" type="number">
    <conditionalDataItem refQuery="Query1" refDataItem="Revenue"/>
    <styleRanges>
      <styleRange value="12000">
        <style>
          <CSS value="background-color:lime"/>
        </style>
      </styleRange>
    </styleRanges>
    <styleRangeRemaining>
      <style>
        <CSS value="background-color:red"/>
      </style>
    </styleRangeRemaining>
  </rangeConditionalStyle>
</namedConditionalStyles>
```

Formatting Data

The children of the `dataFormat` element allow you to format the data in your report.

For example, you can set the date format so that the month, day, and year are separated by forward slashes, or so that the month appears in character format rather than numeric.

To format data items in a report, use the format element that corresponds to the data type of the data item, as listed in the following table:

Data type	Format element
currency	currencyFormat
date	dateFormat
date time	dateTimeFormat
interval	intervalFormat
number	numberFormat
percentage	percentFormat
string	stringFormat
time	timeFormat

For example, to set the number of digits that appear after the decimal point in numeric data, use the `numberFormat` element. To specify the format for dates, use the `dateFormat` element.

Each of the format elements has attributes that specify formatting options applicable to the corresponding data type.

By default, IBM Cognos software sets the format of locale-specific items, such as date format or currency format, based on the content locale specified in IBM Cognos Connection. Set the format of locale-specific items only if you want to override the content locale settings.

Report Variables

Most conditional processing is based on report variables. There are three types of report variables: string, boolean and locale.

The variable to test is specified by the `refVariable` attribute. The test condition is specified by the `refVariableValue` attribute. These attributes are available on most of the conditional elements.

Some typical tasks you may want to do using report variables are as follows.

- Support multiple locales by using different report layouts or displays for each language.
- Decide whether to render or not render report results depending on the sensitivity of the data.
- Add conditional formatting to highlight specific report results.

Note: Although you can use report variables to apply conditional formatting, we recommend using the `namedConditionalStyles` element and its child elements. For more information, see [“Conditional Formatting”](#) on page 1479.

The main report specification objects that are used to define report variables or use report variables are as follows.

Report Specification Objects	Used to
<code>reportVariable</code> element	Define a string, Boolean or locale report variable.
<code>refVariable</code> attribute	Specify the report variable to test for conditional processing.
<code>refVariableValue</code> attribute	Specify the test condition.
<code>conditionalDataSource</code> element	Contain the data source to return based on a report variable.
<code>conditionalChartColor</code> element	Specify a color, color gradient, or pattern to apply to a portion of the chart if a certain condition is true.
<code>conditionalBlock</code> element	Contain a set of layout elements to render based on a report variable.
<code>conditionalDisplayValue</code> element	Specify the display value to use, conditionally based on the <code>refVariable</code> attribute of the <code>selectOptions</code> element.
<code>conditionalRender</code> element	Render the parent layout element based on the evaluation of a report variable.
<code>conditionalLayout</code> element	Render a report with different layouts based on the evaluation of a report variable.

For more information about the use of report variables and conditional processing, see the IBM Cognos Analytics - Reporting *User Guide*.

Support Multiple Locales

You can create a report that supports multiple locales.

When you run a report that retrieves data from a multilingual model, it generates data in the appropriate language for your locale. However, IBM Cognos software does not translate text items that the report author created, like a title on a cover page. Such text items appear in the language that the report author used.

To support users in multiple locales, create a report variable based on locale, and define conditional data sources to use in the report output, based on the user's locale. When a user runs the report, it renders appropriate text strings for his or her locale.

The following XML sample shows how the value of a table cell reflects the language of the report depending on the value of the locale variable.

Procedure

1. Use the `reportVariables` element and its children to create the variable and specify supported locales.

The report variable named "Report Language1" obtains its value from the `ReportLocale` function.

2. Use the `conditionalDataSources` element and its children to specify a list of translated text strings, one for each supported locale. Use the `dataSource` element to specify a default text string to use for unsupported locales.

For information about creating a multilingual report using IBM Cognos Reporting, see the *Reporting User Guide*.

Results

Sample XML

In this sample, the translated text strings are provided in the report, using a `staticValue` element.

```
...
<layouts>
  <layout>
    <reportPages>
      ...
        <dataSource>
          <dataItemLabel refDataItem="Revenue"/>
        </dataSource>
        <conditionalDataSources refVariable="Report Language1">
          <conditionalDataSource refVariableValue="fr">
            <staticValue>Revenus</staticValue>
          </conditionalDataSource>
          <conditionalDataSource refVariableValue="de">
            <staticValue>Einnahmen</staticValue>
          </conditionalDataSource>
          <conditionalDataSource refVariableValue="ja">
            <staticValue>??</staticValue>
          </conditionalDataSource>
        </conditionalDataSources>
      </textItem>
    </contents>
  </listColumnTitle>
  ...
</list>
</contents>
</pageBody>
</page>
</reportPages>
</layout>
</layouts>
<reportVariables>
  <reportVariable name="Report Language1" type="locale">
    <reportExpression>ReportLocale()</reportExpression>
    <variableValues>
      <variableValue value="fr"/>
      <variableValue value="de"/>
      <variableValue value="ja"/>
    </variableValues>
  </reportVariable>
</reportVariables>
</report>
```

Optimize Query Performance

If your reports are taking too long to run, you may be able to improve performance by optimizing the queries in your reports. The tasks described in this section will help you to optimize your queries.

IBM Cognos Framework Manager models may include governors that optimize query performance. A governor is a set of rules to limit user activities, such as the execution of reports, that either take too long or consume too many resources. By default, at run time IBM Cognos software applies governors that are set in the model to all queries that query metadata from that model. For information about setting governors in a model, see the *Framework Manager User Guide*.

Similarly, you can optimize or otherwise control how IBM Cognos software executes a specific query by adding hints to a specific query in a report specification. Like governors, query hints can be applied to sorting, joins, row retrieval, and other aspects of the query. The governors can also affect how IBM Cognos software builds the SQL for the query. In general, hints do not return errors whereas governors do return errors when limits set in the governor are exceeded.

Use Query Hints

Query hints are specified on the query in the report specification. These hints allow you to control how the query is executed. If setting a query hint on the query does not produce the results that you intended, check the governors specified in the IBM Cognos Framework Manager model.

Procedure

1. Add a `queryHints` element to the query that you want to optimize.
2. Add the appropriate query hint child element to the `queryHints` element.

Choose from any of the query hints in the following table:

<i>Table 243. Use of query hints in report specifications</i>		
Goal	Child Element to Add to the queryHints Element	Overrides this Governor, if Set in the Model
Warn about, prevent, or allow cross-product joins	<code>crossProductAllowed</code>	Cross-Product joins
Warn about, prevent, or allow outer joins	<code>outerJoinAllowed</code>	Outer joins
Specify whether and how to automatically sort level keys	<code>autoSort</code>	
Specify that a request continues or terminates when a calculation or expression encounters a divide by zero error	<code>avoidZeroDiv</code>	
Retrieve either all rows or the first row when rendering the output	<code>executionOptimization</code>	
Specify whether any of the query processing is performed by the database server	<code>queryProcessing</code>	
Specify whether rollup aggregates are computed by the database server, and if not, whether extended or running aggregates are computed	<code>rollupProcessing</code>	
Specify whether to allow the query to run concurrently or to run the query sequentially.	<code>executionMethod</code>	
Specify whether this query is given priority in determining parameter information.	<code>useForParameterInfo</code>	
Allow or prevent local caching	<code>localCache</code>	Allow usage of local cache

Table 243. Use of query hints in report specifications (continued)		
Goal	Child Element to Add to the queryHints Element	Overrides this Governor, if Set in the Model
Generate WITH clauses in the SQL, or generate derived tables instead	useSQLWithClause	Use WITH clause when generating SQL
Stop a query from continuing if the length of time it runs exceeds the number of seconds that you specify	maxQueryExecutionTime	Query execution time limits
Stop a query from continuing if the number of rows retrieved exceeds the value you specify	maxRowsRetrieved	Data retrieval limits
Stop a query from continuing if the number of tables queried exceeds the value you specify	maxTablesPerQuery	Report table limits
Stop a query from continuing if the size of any text BLOB exceeds the number of characters that you specify	maxTextBlobCharacters	Large text items limit
Specify whether to suppress null values	suppress	
Specifies whether to use explicit or implicit join syntax in IBM Cognos SQL.	useSQLJoinSyntax	SQL Join Syntax
Specifies whether to use the complete SAP MUN as the business key.	useSAPMUNAsBusinessKey	
Specifies whether to use the cache server for dimensions specified in the query.	cacheServerQuery	
Specifies whether generated SQL uses parameter markers or literal values.	useSQLParameters	SQL Parameter Syntax

Sample XML to add query hints:

```

<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
        expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        ...
      </selection>
      <queryHints>
        <outerJoinAllowed value="warn"/>
        <crossProductAllowed value="deny"/>
        <autoSort value="none"/>
        <executionOptimization value="firstRow"/>
        <queryProcessing value="databaseOnly"/>
        <rollupProcessing value="database"/>
        <avoidZeroDiv value="true"/>
      </queryHints>
    </query>
  </queries>
</report>

```

```

    <maxRowsRetrieved value="1500"/>
    <maxTablesPerQuery value="20"/>
    <maxQueryExecutionTime value="1800"/>
    <maxTextBlobCharacters value="1500"/>
    <useSQLWithClause value="false"/>
    <localCache value="true"/>
  </queryHints>
</query>
</queries>
<layouts>
  ..
</layouts>
</report>

```

Results

Setting the `value` attribute to `warn` pauses the query when the outer join is detected. The report user may then either cancel or continue the query.

If you want to prevent queries from executing when outer joins are detected, set the `value` attribute to `deny` instead. For more information about these options, see `outerJoinAllowed`.

Reuse Queries

When you run a report, the query request is sent to the database and the result set is returned. After the initial report execution, you may decide to make changes to the report. Often, the report can be changed without querying the database again. To take advantage of this, you use the `queryHints` element to re-use the query.

When query reuse is turned on and you run a report for the first time, the query is stored in the cache of your current session and reused the next time you run the report. The queries are kept in the cache for each user. The cache is cleared when the report consumer exits the reporting tool and returns to the portal or when the report server times out the session.

The first time the report is run and the cache is created, the response time may be slow. A performance improvement is realized by the report consumer on each subsequent report execution. This performance improvement occurs because the report does not have to re-query the database. In addition to this, a reduction in queries to the database improves overall system performance for all users.

Query reuse can be set on the model or on individual reports. To specify that all reports using a particular model use cached data, set the appropriate governor on the model in IBM Cognos Framework Manager and republish the model. By default, this setting affects all reports that use this model.

Procedure

1. Add a `queryHints` element to the query that you want to reuse.
2. Add a `localCache` element to the `queryHints` element.

Sample XML

```

<query name="Query1">
  <source>
    <model/>
  </source>
  <selection>
    <dataItem aggregate="none" name="Product type">
      <expression>[gosales_goretailers].[Products].[Product type]</expression>
    </dataItem>
  </selection>
  <queryHints>
    <localCache value="true"/>
  </queryHints>
</query>

```

Summarize Data

In the report specification, you can control how to summarize numeric data, just as you can by using IBM Cognos Analytics - Reporting.

When you query a relational data source, you can specify when and how to calculate regular aggregates and rollup aggregates. However, when you query an OLAP data source, you cannot override the type of regular aggregate because the data sources contain rolled up values.

A regular aggregate is the type of aggregate that is calculated when a data item is summarized. A rollup aggregate is the type of aggregate to apply when calculating the rollup values. A rollup involves summarizing values for a particular level in a dimension.

In list reports, summaries are rendered by header and footer elements. In crosstabs, summaries are rendered by expressions.

Some typical tasks that you may want to do in the report specification are as follows.

- Set the type of regular or rollup aggregates instead of using the types specified in the model.
- Show detail rows and summaries for grouping levels that you specify.
- Create summaries using expressions.
- Filter summarized data so you can view a subset of the report results.

The main report specification objects you may use in summary tasks are listed in the following table.

Report Specification Objects	Description
aggregate attribute of the dataItem element	Specifies the type of aggregate to calculate for this data item when the data item is summarized.
rollupaggregate attribute of the dataItem element	Specifies the type of aggregate to apply when calculating a rollup value.
expression element	Specifies a summary for a crosstab.
autosummary attribute of selection element	Specifies whether to group and summarize the results of the query.
listGroup element	Defines one level of grouping in a list report. The grouping defines a rollup aggregate level.
summaryFilter element	Specifies a filter that is applied after aggregates are calculated.

We recommend that you first use IBM Cognos Analytics - Reporting and the Reporting *User Guide* to familiarize yourself with the properties and settings that affect summaries. By viewing the resulting report specifications, you can see what parts of the report specification are affected.

Set the Type of Aggregate

When a data item is summarized in a report, IBM Cognos software must identify the type of aggregate to calculate. If no aggregates are specified in the report specification, the defaults from the model are used.

If an aggregate is specified in both the report specification and in the model, the type of aggregate in the report specification takes precedence.

Note: when you query an OLAP data source, you cannot override the type of regular aggregate because the data sources contain rolled up values.

To override the aggregates that are set on the query item in the model, specify the aggregate attribute and/or the rollupAggregate attribute on the dataItem element in the selection element. The

advantage of this approach is that if a modeler makes changes to query item aggregates in the model, those changes do not affect the type of aggregates that you already specified in your reports. This ensures that you get the results that you want.

To use the aggregate that is specified in the model, either omit or remove the aggregate attribute and/or the rollupAggregate attribute from the dataItem element in the selection element. The advantage of this approach is that subsequent changes to query item aggregates in the model are applied to new reports. This is useful when you want the type of aggregate in the report to always reflect the aggregate that the modeler specifies.

If you do not specify either attribute, IBM Cognos software uses the aggregate set on the query item in the model and the report runs as if the aggregate attribute had a default value of automatic. If the Regular Aggregate property is not specified on the query item in the model, IBM Cognos software uses the data type of the expression result to determine the type of aggregate function to apply. For more information, see the Framework Manager *User Guide*.

Procedure

1. For regular aggregates, add an aggregate attribute to the dataItem for which you want to calculate an aggregate.

For a complete list of values, see the description of the aggregate attribute on the dataItem element.

In this sample, Quantity and Unit sale price data items are the numeric data items whose values you want summarized.

The aggregate attributes are also set for the Product type and Product name data items. The value of the attribute is none because you do not want to compute aggregates for these data items.

2. For rollup aggregates, add a rollupaggregate attribute to the dataItem for which you want to calculate an aggregate.

By default, the rollupaggregate value is the same as the aggregate. You only need to add the rollupaggregate attribute if you want to use a different value. For a complete list of values, see the description of the rollupaggregate attribute on the dataItem element.

If no groups have been specified, a single rollup summary is calculated and rendered at the bottom of the report.

3. Ensure that the autoSummary attribute of the selection element is set to true or is omitted. This enables automatic grouping and summarization. The aggregate attribute of each dataItem element contained in the selection is recognized.

Results

Sample XML

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem name="Product type" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
        <dataItem name="Product name" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product name]</expression>
        </dataItem>
        <dataItem name="Order number" aggregate="none">
          <expression>[gosales_goretailers].[Orders].[Order number]</expression>
        </dataItem>
        <dataItem name="Unit sale price" aggregate="average">
          <expression>[gosales_goretailers].[Orders].[Unit sale price]</expression>
        </dataItem>
      </selection>
    </query>
  </queries>
</report>
```

```

<dataItem name="Quantity" aggregate="total">
  <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
</dataItem>
</selection>
</query>
</queries>

```

Show Detail Rows and Rollup Summaries

You can create a report that shows a combination of detail rows with transactional, unsummarized data as well as summaries for grouping levels that you specify.

Procedure

1. Add an `rollupAggregate` attribute to the `dataItem` for which you want to calculate an aggregate.

To include multiple summaries for the same data item, you must add one `dataItem` element for each summary type. Then, set the `rollupAggregate` attribute on each `dataItem` to the required value.

For a complete list of values, see the description of the `rollupAggregate` attribute on the `dataItem` element.

2. Ensure that the `autoSummary` attribute of the `selection` element is set to `false`. This disables automatic grouping and summarization. The `aggregate` attribute of each `dataItem` element contained in the `selection` is ignored so no regular aggregates are calculated.
3. If no groups have been specified, a single rollup summary will be calculated and rendered at the bottom of the report. To show other rollup summaries, you must identify groups where these rollups will occur. Groupings are identified using `listGroup` elements in the layout section.

Rollup summaries are calculated and displayed for Revenue and Quantity for Product Type and Product Name, based on the information in the layout section.

Results

Sample XML

```

<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
        expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection autoSummary="false">
        <dataItem aggregate="none" name="Product type">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
        <dataItem aggregate="none" name="Product name">
          <expression>[gosales_goretailers].[Products].[Product name]</expression>
        </dataItem>
        <dataItem aggregate="none" name="Order number">
          <expression>[gosales_goretailers].[Orders].[Order number]</expression>
        </dataItem>
        <dataItem aggregate="average" name="Unit sale price">
          <expression>[gosales_goretailers].[Orders].[Unit sale price]</expression>
        </dataItem>
        <dataItem aggregate="total" name="Quantity" rollupAggregate="maximum">
          <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
        </dataItem>
        <dataItem name="Revenue" rollupAggregate="total">
          <expression>[Unit sale price]*[Quantity]</expression>
        </dataItem>
      </selection>
    </query>
  </queries>

```

...

```

<layouts>
  <layout>

```

```

<listGroups>
  <listGroup refDataItem="Product type">
    ...

  <listGroup refDataItem="Product name">
    ...

```

Create a Calculated Column

You can create calculated columns using more complex expressions than are available using the predefined aggregate and rollup aggregate functions. To create a calculated column, you create a dataItem that is not available from the model. Then use the expression associated with the dataItem to define the value of the column.

You can create complex expressions using functions, operators, summaries, and constants that produce a single value. For information about the components you can use in a report expression, see the IBM Cognos Reporting *User Guide*. The functions available when creating expressions are restricted to the functions available in your data source.

Procedure

1. Create a dataItem element whose expression child element contains a calculation.

Tip: If the calculation you need is a common requirement for report users, we recommend that you create the calculation in the model instead of in the report. For more information, see the Framework Manager User Guide.

2. Add the dataItem to the appropriate place in the report layout.

For more information about where to place the dataItem in a list report, see [“Layout Portion of the List Report Sample” on page 1467](#).

Results

In the following sample, the Revenue column does not exist in the model. Its value is calculated from the Unit Sale Price and Orders dataItems from the model.

```

<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem name="Product type" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
        <dataItem name="Product name" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product name]</expression>
        </dataItem>
        <dataItem name="Order number" aggregate="none">
          <expression>[gosales_goretailers].[Orders].[Order number]</expression>
        </dataItem>
        <dataItem name="Unit sale price" aggregate="average">
          <expression>[gosales_goretailers].[Orders].[Unit sale price]</expression>
        </dataItem>
        <dataItem name="Quantity" aggregate="total">
          <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
        </dataItem>
        <dataItem name="Revenue">
          <expression>[Unit sale price]*[Quantity]</expression>
        </dataItem>
      </selection>

```

Ways of Filtering Data

Filters, prompts and slicers provide ways of selecting or restricting the data that appears in your report.

You can apply filters to narrow the results returned by a query, based on criteria that you supply in a condition. Filters can be applied to detail rows before any aggregation occurs, which is equivalent to adding a WHERE clause to an SQL SELECT statement. Filters can also be applied to summary rows, which is equivalent to using a HAVING clause.

A prompt, or prompt control, provides a way to obtain user input for a filter or calculation. A parameter is a placeholder for a value that the user is asked to provide. A parameter value is the answer that the user provides when asked for input. The parameter value populates the parameter and supplies the criteria for the filter or calculation.

A slicer is a dimensional filter that is applied to the cells but not the rows or columns in a crosstab. It is defined as part of the query.

Some typical tasks you may want to do in the report specification are:

- Add filters to both queries and subqueries
- Add a summary or detail filter
- Add a prompt to a separate prompt page
- Filter the cells in a crosstab using a slicer

The main report specification objects you may want to use in filtering data are shown in the following table.

Report Specification Objects	Used to
promptPages element	Contain all the prompt pages for the report.
promptButton element	Represent an HTML-style navigation button used in prompt pages.
detailFilter element	Specify a filter that is applied to the input data stream before aggregations or calculations are performed.
summaryFilter element	Specify a filter that is applied after aggregates are calculated.
summaryFilterLevel element	Specify a level to which the summary filter is applied.
selectDate, selectDateTime, selectInterval, selectOption, selectTime, selectValue, selectWithSearch, selectWithTree, selectWithTreeItem elements	Specify prompt controls with characteristics based on the type of control.
defaultSelections, defaultTreeSelection, defaultRangeSelection, defaultSimpleSelection elements	Specify the default values of prompt controls based on the type of control.
generatedPrompt element	Specify an advanced control that acts as a placeholder for a prompt.
slicer element	Create dimensional filters that reduce the data included in measure rollups.

Add a Filter and a Prompt Sample

In this sample, the first query contains a filter that allows the report to return a subset of data based on a prompt selection made by the user. The second query contains the data item that provides values for the prompt. This sample requires a second query to ensure that the prompt has the correct Product type values. If you used the Product type data item from the first query to populate the prompt, you could get duplicate values and other errors.

If there are very few conditions, and you can easily manage them in one expression, add more conditions to the contents of the `filterExpression` element, and then separate them using an and operator. If there are many conditions that you want to manage separately, or if you use prompts to obtain filter values and you want some filters to be required, and others optional, add a `detailFilter` element to the `detailFilters` parent element for each additional filter condition.

The prompt is the drop-down list of values, the parameter is a placeholder for the selected product type, and the parameter value is the specific product type that the user selects.

The prompt is represented by the `selectValue` element in the layout section, and the parameter is located in the `filterExpression` element. The parameter name for Product type is `?Parameter1?`. The Product type values appear at run time. The parameter value is the value that the user specifies at run time when presented with the drop-down list.

The name you assign to the parameter in the layout section must match the parameter specified in the `filterExpression` element of the `filter` element.

In this sample, the prompt is defined under the `promptPages` element. When the report is run, the prompt is shown separately from the report results. The prompt page usually includes a prompt and prompt buttons. You can also put the prompt control directly on the report page.

The prompt page must contain a way to proceed to the next step of the prompt report. In this sample, buttons are provided for this purpose. When the user clicks **Finish**, the next step is for the report to run because this sample includes only one parameter and one prompt page. If you build reports that include multiple parameters or multiple prompt pages, the next step prompts the user to supply values for other parameters. If you do not want to use prompt buttons, you must set the `autoSubmit` attribute on the `selectValue` element to `true`.

The XML that shows the filter and the elements required to produce the prompt is highlighted in bold:

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        ...
      </selection><detailFilters>
      <detailFilter>
      <filterExpression>[gosales_goretailers].[Products].[Product
type] = ?Parameter1?</filterExpression>
      </detailFilter>
      </detailFilters>
    </query>
    <query name="Query2">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem aggregate="none" name="Product type">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
      </selection>
    </query>
  </queries>
  <layouts>
    <layout>
      <reportPages>
        <page name="Page1">
```

```

...
</page>
</reportPages>
<b>promptPages</b>
  <page name="Prompt Page1">
    <pageBody>
      <contents>
        <selectValue multiSelect="false" parameter="Parameter1" refQuery="Query2"
required="true" selectValueUI="dropdown">
<useItem refDataItem="Product type"/>
</selectValue>
      </contents>
      <style>
        <defaultStyles>
          <defaultStyle refStyle="py"/>
        </defaultStyles>
      </style>
    </pageBody>
    <pageFooter>
      <contents>
        <b>promptButton type="cancel"</b>
        <contents/>
        <style>
          <defaultStyles>
            <defaultStyle refStyle="bp"/>
          </defaultStyles>
        </style>
        </promptButton>
        <b>promptButton type="back"</b>
        <contents/>
        <style>
          <defaultStyles>
            <defaultStyle refStyle="bp"/>
          </defaultStyles>
        </style>
        </promptButton>
        <b>promptButton type="next"</b>
        <contents/>
        <style>
          <defaultStyles>
            <defaultStyle refStyle="bp"/>
          </defaultStyles>
        </style>
        </promptButton>
        <b>promptButton type="finish"</b>
        <contents/>
        <style>
          <defaultStyles>
            <defaultStyle refStyle="bp"/>
          </defaultStyles>
        </style>
        </promptButton>
      </contents>
      <style>
        <defaultStyles>
          <defaultStyle refStyle="fp"/>
        </defaultStyles>
      </style>
    </pageFooter>
  </style>
  <defaultStyles>
    <defaultStyle refStyle="pp"/>
  </defaultStyles>
</style>
</page>
</promptPages>
</layout>
</layouts>
...
</report>

```

Filter Measure Values in a Crosstab Report Sample

To filter measure values when data is summarized for more than one dimension, such as in a crosstab or chart report, you must specify more than one level in a summary filter.

In this sample, the `summaryFilter` includes two `summaryFilterLevel` elements because the measure values that are filtered are summary values at intersections of two level groups from different

edges. The return quantity filter is applied for two levels of grouping, Product type and Order method. So there is one `summaryFilterLevel` element for each level.

Sample XML

The XML required to create this filter is highlighted in bold.

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
  <modelPath>/content/package[@name='GO Sales and Retailers']
    /model[@name='model']</modelPath>
  <queries>
    <query name="Query1">
      <source>
        <model/>
      </source>
      <selection>
        <dataItem name="Product line" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product line]</expression>
        </dataItem>
        <dataItem name="Product type" aggregate="none">
          <expression>[gosales_goretailers].[Products].[Product type]</expression>
        </dataItem>
        <dataItem name="Order method" aggregate="none">
          <expression>[gosales_goretailers].[Orders].[Order method]</expression>
        </dataItem>
        <dataItem name="Return quantity" aggregate="total">
          <expression>[gosales_goretailers].[Orders].[Return quantity]</expression>
        </dataItem>
        <dataItem name="Total(Product type)">
          <expression>total(currentMeasure within set [Product type])</expression>
        </dataItem>
        <dataItem name="Total(Order method)">
          <expression>total(currentMeasure within set [Order method])</expression>
        </dataItem>
      </selection>
      <summaryFilters>
        <summaryFilter use="required">
          <filterExpression>[Return quantity]&gt;200</filterExpression>
          <summaryFilterLevels>
            <summaryFilterLevel refDataItem="Product type"/>
            <summaryFilterLevel refDataItem="Order method"/>
          </summaryFilterLevels>
        </summaryFilter>
      </summaryFilters>
    </query>
  </queries>
  <layouts>
    ..
  </layouts>
</report>
```

Filter Data With a Subquery

You can create a report that uses one or more subqueries to perform operations that would otherwise require complex joins and unions. For example, you can create a query that contains a detail filter that references the results of a subquery.

The filter on the subquery references the Minimum life to date product revenue parameter, which obtains its value from a numeric text box prompt on the prompt page. The filter excludes data rows for which the life to date revenue for a product is less than the amount specified.

Procedure

1. Define the main query and the subquery in the `queries` element as two separate query elements.
2. To the main query, add a `detailFilter` that references the subquery.

Sample XML

```
<report xmlns="http://developer.cognos.com/schemas/report/7.0/"
  expressionLocale="en-us">
```

```

<modelPath>/content/package[@name='GO Sales and Retailers']
  /model[@name='model']</modelPath>
<queries>
  <query name="Query1">
    <source>
      <model/>
    </source>
    <selection>
      <dataItem aggregate="none" name="Product name">
        <expression>[gosales_goretailers].[Products].[Product name]</expression>
      </dataItem>
      <dataItem aggregate="none" name="Order number">
        <expression>[gosales_goretailers].[Orders].[Order number]</expression>
      </dataItem>
      <dataItem aggregate="none" name="Order date">
        <expression>[gosales_goretailers].[Orders].[Order date]</expression>
      </dataItem>
      <dataItem aggregate="total" name="Revenue">
        <expression>[gosales_goretailers].[Orders].[Revenue]</expression>
      </dataItem>
      <dataItem aggregate="total" name="Quantity">
        <expression>[gosales_goretailers].[Orders].[Quantity]</expression>
      </dataItem>
    </selection>
    <detailFilters>
      <detailFilter>
<filterExpression>[Query1].[Order date]&gt;= ?Date?</filterExpression>
      </detailFilter>
      <detailFilter>
<filterExpression>[Query1].[Product name]in ([Query2].[Product name])</filterExpression>
      </detailFilter>
    </detailFilters>
  </query>
  <query name="Query2">
    <source>
      <model/>
    </source>
    <selection>
      <dataItem aggregate="none" name="Product name">
        <expression>[gosales_goretailers].[Orders].[Product name]</expression>
      </dataItem>
      <dataItem aggregate="total" name="Revenue">
        <expression>[gosales_goretailers].[Orders].[Revenue]</expression>
      </dataItem>
    </selection>
    <detailFilters>
      <detailFilter postAutoAggregation="true">
<filterExpression>[Query2].[Revenue]&gt;=?Minimum life to date product revenue?</filterExpression>
      </detailFilter>
    </detailFilters>
  </query>
</queries>
<layouts>
  <layout>
    <reportPages>
      ...
    </reportPages>
    <promptPages>
      <page name="Prompt Page1">
        <pageBody>
          <contents>
            <block>
              <contents>
<selectDate multiSelect="false" parameter="Date" range="false" required="true"/>
              </contents>
              ...
              <contents>
<textBox numbersOnly="true" parameter="Minimum life to date product revenue"/>
              </contents>
            </block>
          </contents>
        </pageBody>
      </page>
    </promptPages>
  </layout>
  ...
</report>

```


Define a Slicer

Use slicers to create dimensional filters that reduce the data included in measure rollups. A slicer is a filter that is applied to the cells but not the rows or columns in a crosstab.

Sample XML

The query defines all the data items included in the report. The slicer is defined in the query. The `slicerMemberSet` reduces the data to that related only to Camping Equipment, Golf Equipment and Outdoor Protection.

The XML required to define the slicer is highlighted in bold.

```
<query name="Query1">
  <source>
    <model/>
  </source>
  <selection>
    <dataItem name="Revenue">
      <expression>[great_outdoors_company].[Measures].[Revenue]</expression>
    </dataItem>
    <dataItem aggregate="none" name="Product line">
      <expression>[great_outdoors_company].[Products].[Products]
        .[Product line]</expression>
    </dataItem>
    <dataItem name="Fax">
      <expression>[great_outdoors_company].[Order Method].[Order
Method].[Order Method1]-&gt;;[PC].[Order Method (Root)].[601]</expression>
    </dataItem>
    <dataItem name="Web">
      <expression>[great_outdoors_company].[Order Method].[Order
Method].[Order Method1]-&gt;;[PC].[Order Method (Root)].[605]</expression>
    </dataItem>
    <dataItem name="Camping Equipment">
      <expression>[great_outdoors_company].[Products].[Products]
        .[Product line]-&gt;;[PC]
        .[Products (Root)].[1]</expression>
    </dataItem>
    <dataItem name="Mountaineering Equipment">
      <expression>[great_outdoors_company].[Products].[Products]
        .[Product line]-&gt;;[PC]
        .[Products (Root)].[2~239]</expression>
    </dataItem>
    <dataItem name="Outdoor Protection">
      <expression>[great_outdoors_company].[Products].[Products]
        .[Product line]-&gt;;[PC]
        .[Products (Root)].[4~237]</expression>
    </dataItem>
    <dataItem name="Golf Equipment">
      <expression>[great_outdoors_company].[Products].[Products]
        .[Product line]-&gt;;[PC]
        .[Products (Root)].[5~236]</expression>
    </dataItem>
    <dataItem aggregate="none" name="Retailer type">
      <expression>[great_outdoors_company].[Retailer].[Retailer]
        .[Retailer type]</expression>
    </dataItem>
    <dataItem aggregate="none" name="Year">
      <expression>[great_outdoors_company].[Years].[Years].[Year]</expression>
    </dataItem>
  </selection>
  <bold><slicer><slicerMemberSet>set([Fax],[Web])</slicerMemberSet>
<slicerMemberSet>set([Camping Equipment],[Golf Equipment],[Outdoor Protection])
</slicerMemberSet></slicer></bold></query>
</queries>
```

Chapter 24. Creating custom report functions and function sets

This chapter describes how to create custom report functions and custom function sets for report authors to use in IBM Cognos Analytics.

Creating custom report functions

Report authors create report expressions using the expression editor. The expression editor provides a list of functions that can be used in expressions. In addition to the functions that are available by default, such as `Today()`, `ReportDate()` or `ReportName()`, you can create custom functions and make them available to report authors by defining report function libraries.

All functions available through the expression editor must be identified in the function definition service, a series of files that tells the expression editor what functions are defined. Report function collections are provided to report authors through dynamic load libraries on Windows operating systems, in shareable libraries on UNIX operating system, or in shared object files on the Linux operating system.

Many types of function metadata definitions are shared among IBM Cognos applications and their components. Only the requirements for defining custom report functions are outlined in this chapter.

To make custom report functions available to report authors, you:

- Build a custom report functions library [“Building a custom report functions library” on page 1497](#)
- Register the report functions library [“Registering custom report functions” on page 1500](#)
- Install the custom report functions [“Installing a custom report functions library” on page 1502](#)

For an example, see [“Example of a custom report functions implementation” on page 1503](#).

Building a custom report functions library

Custom report functions can be built in any programming language that allows creation of the appropriate file type - dynamic load libraries on Windows operating systems, shareable libraries on UNIX operating system, or shared object files on the Linux operating system.

The report function declaration must follow a specific format, as defined in the `crxSDK.h` file. In compiling your `.dll` files, this header file is always included via an include statement.

Example of a report function prototype

Report functions may have any number of arguments, ranging from none to 15. The report function prototype, consisting of its name and arguments, is defined as follows in the `crxSDK.h` file.

```
typedef CCLDBColumnState (*PF_CallFunction)
(
    void* result,
    uint resultsize,
    const crxDataI* context,
    void* arg1,
    void* arg2,
```

```

void* arg3,
void* arg4,
void* arg5,
void* arg6,
void* arg7,
void* arg8,
void* arg9,
void* arg10,
void* arg11,
void* arg12,
void* arg13,
void* arg14,
void* arg15,
void* arg16
);

```

where:

- The first argument receives the result of the function execution.
- The second argument, resultsize, sets the size of the results buffer in bytes. The buffer is pre-allocated by the expression engine.
- The third argument is the context, and is ignored.
- All other arguments are the function input arguments as specified in the function definition file. [“Function definition file” on page 1500.](#)

Result and function arguments

The result and the function input arguments are pointers to any of the types shown in this table.

Supported Types	Comments
CCL_int8	
CCL_uint8	
CCL_int16	
CCL_uint16	
CCL_int32	

Table 246. Result and Function Argument types (continued)

Supported Types	Comments
CCL_uint32	
CCL_int64	
CCL_uint64	
CCL_float32	
CCL_float64	
CCL_char[CRX_MAXIMUM_STR_CODEPOINTS]	for strings
CCL_uint8[CRX_MAX_DECIMAL_BYTES]	for decimals
CCLDate2	
CCLTime2	
CCLTimeTZ	
CCLDateTime	
CCLTimeStamp2	
CCLTimeStampTZ	
CCLIntervalYM	
CCLInterval2	

Note: For the definition of these types, see the `crxSDK.h` file located in the *installation_location/webcontent/samples/sdk/crx/crxSDKsample* directory.

Context argument

The `context` argument points to a helper object that the expression engine and its client application, IBM Cognos Analytics, use to handle the variables that can be specified in report expressions. It is used:

- At compilation time, to resolve variables by name and retrieve their properties (type, size, precision, scale).
- At execution time, to retrieve the variables' values.

Custom functions do not use the `context` argument. For these functions, this argument is always null.

Report function return value

The value returned by the report function call, of type `CCLDBColumnState`, tells the expression engine the status of the function execution. `CCLDBColumnState` may take any of the following values.

CCL_DB_COLSTATE_OK

Function call was successful.

CCL_DB_COLSTATE_NULL

One of the function arguments was missing (NULL).

CCL_DB_COLSTATE_NA

One of the function arguments was unavailable.

CCL_DB_COLSTATE_DIVBYZERO

A divide-by-zero error occurred.

CCL_DB_COLSTATE_OVERFLOW

For numerics, an overflow or underflow occurred. For strings, truncation of the string occurred.

CCL_DB_COLSTATE_SECURITY

Access to one of the function arguments was prohibited for security reasons.

CCL_DB_COLSTATE_UNKNOWN

Status is reserved for cases where the status is not truly known.

CCL_DB_COLSTATE_ERROR

A generic error indicating all other cases.

CCL_DB_CASTING_ERROR

Invalid data was passed to a data type casting function.

CCL_DB_COLSTATE_SAMPLE

Temporary status returned by the engine while processing is not yet complete. Internal use only.

Registering custom report functions

After building the dynamic load libraries, you must register the functions so the expression engine can recognize the custom functions.

To register the functions, you must perform the following tasks:

- Create a custom file list “[File list](#)” on page 1500.
- Create a function definition file “[Function definition file](#)” on page 1500.
- Create one or more function description files “[Function description files](#)” on page 1502.

The default files used by the Function Definition Service are located in the *installation_location/configuration/functions* directory. You can use these as models to create your custom files.

For an example, see “[Example of a custom report functions implementation](#)” on page 1503.

To register your custom functions after an upgrade, the custom function files can simply be copied back into the functions directory.

As in previous releases, you can still add custom functions to a default group. However, changes to the Function Definition Service are not retained after you upgrade to another version of IBM Cognos Analytics. If you modify a default group and then upgrade IBM Cognos Analytics, you will have to recreate your custom functions.

File list

FileList.xml is the default file where all function definition files provided by IBM Cognos are listed. For custom functions, you must create a similar file with the name pattern of `filelistn.xml`, where *n* is any name that you assign. For example,

```
filelist_custom.xml
```

This file will contain a list of your custom function definition files and related function description files.

Function definition file

For each entry in your custom filelist, you must create a function definition file.

Every report function and its `function` element must be unique in the entire series of function definition files. To avoid duplication of functionality, check that the functions you require don't already exist in the default definition files. The default files are `cogRSReportFunctions.xml` and `cogCRXReportFunctions.xml`.

The schema file for the function definition file is FuncTree.xsd. It is located in the *installation_location/configuration/functions* directory.

Every group of functions is described by a `group` element uniquely identified by the value of its `id` element. Every report function in the group must be described within a `function` element. The function element contains these elements described here.

id

A unique string identifier designated by the developer and used internally by the expression engine. The function description files also use this identifier. The `id` element must be unique across the entire set of IBM Cognos functions.

name

The name that shows up in the expression editor tree controls. It can be overridden by an entry in the language file.

canonical

The name of the report function as defined in the dynamic load library.

dll

The library name. The extension (.dll, .so) is not included if it matches the default for the platform. The default path for the library is the bin directory. You can also specify a path relative to the default path.

context

This element is used internally by the definition service, and must always be set to CRX.

returnType

The type for the value returned by the function to the user. It must map to the `result` argument type as specified in the report function declaration. For more information, see [“Type mapping” on page 1501](#).

parameter

Optional. Used to describe function input arguments. Every `parameter` must contain a type element.

type

Sub-element of the `parameter` element. Must map to the report function argument type. For more information, see [“Type mapping” on page 1501](#).

Type mapping

There is a direct one to one mapping between the `returnType` and `type` elements in the function definition file and the argument type as defined in the .dll file.

Possible types for `returnType` and `type` elements and their corresponding mappings are shown in the following table.

returnType or type element	Report function argument type from crxSDK.h file
crxDTypeInt8	CCL_int8
crxDTypeUInt8	CCL_uint8
crxDTypeInt16	CCL_int16
crxDTypeUInt16	CCL_uint16
crxDTypeInt32	CCL_int32
crxDTypeUInt32	CCL_uint32
crxDTypeInt64	CCL_int64
crxDTypeUInt64	CCL_uint64
crxDTypeFloat	CCL_float32

Table 247. *returnType* and *type* mappings (continued)

returnType or type element	Report function argument type from crxSDK.h file
crxDTypeDouble	CCL_float64
crxDTypeString	CCL_char[CRX_MAXIMUM_STR_CODEPOINTS]
crxDTypeDecimal	CCL_uint8[CRX_MAX_DECIMAL_BYTES]
crxDTypeDate	CCLDate2
crxDTypeTime	CCLTime2
crxDTypeTimeTZ	CCLTimeTZ
crxDTypeDatetime	CCLTimeStamp2
crxDDatetimeTZ	CCLTimeStampTZ
crxDTypeYMInterval	CCLIntervalYM
crxDTypeDTInterval	CCLInterval2

For the definition of these types, see the `crxSDK.h` file located in the *installation_location/webcontent/samples/sdk/crx/crxSDKsample* directory.

Function description files

For each function definition file, you need to create at least one function description file. There must be one description file for each supported language. Each file contains the function name, syntax, and tip for a particular language. The functions described in these files are cross-referenced by the `id` attribute of the function element.

Each function description file is named by combining the file name and a locale identifier, separated by an underscore. If only one description file is provided, the locale identifier must be `en`. The contents of the file, however, can be in any language.

For example, if the function definition file name is `crxSDKSampleTree.xml`, then the function description files could be named `crxSDKSampleStrings_xx.xml` where `xx` stands for any locale identifier, such as `en` for English or `ja` for Japanese.

The `i18n_res.xml` file, located in the `bin` directory, contains the list of locales and their identifiers. If this file is missing, IBM Cognos Analytics substitutes a standard list of default locales: `en` (English), `fr` (French), `de` (German), and `ja` (Japanese).

The English function description file is the default. If the requested localized file does not exist, the English file will be used.

The content of the function description files is used in the expression editor. Each function description has three parts that are described here.

function name

Identifies the function in the functions list in the expression editor.

syntax

Describes the exact format and required parameters that must be entered by the report author.

tip

Describes what the function does.

Installing a custom report functions library

IBM Cognos Analytics recognizes your custom functions automatically, once you ensure that the three function files you create are located in the *installation_location/configuration/functions* directory.

To install your custom functions after an upgrade, the custom function files can simply be copied back into the functions directory.

The location that you specify in the `dll` element of a function definition file tells the expression engine where to find the corresponding dynamic load library or libraries. You must ensure that this specification matches the location of the `.dll` files. If you specify the file name only, ensure that the `.dll` files are located in the `installation_location/bin` directory.

Example of a custom report functions implementation

This topic illustrates the implementation of a set of custom report functions.

The Sample files are located in the `installation_location/webcontent/samples/sdk/crx/crxSDKsample` directory. They are described in the following table.

File Name	Purpose	Description
<code>crxSDKSample.cpp</code>	sample C++ code	Is required to add the custom report functions. It will be compiled into a dynamic load library and referenced in the function definition file
<code>crxSDK.h</code>	header file	Contains the report function prototype, and the type definitions used by CRX report functions. It must be referenced by an include statement in each dynamic load library you create for custom report functions.
<code>FileList_custom.xml</code>	custom file list file	Identifies the custom function definition and description files.
<code>crxSDKSampleTree.xml</code>	function definition file	Represents a custom group of functions. The file identifies the functions and their location to FDS.
<code>crxSDKSampleStrings_en.xml</code>	function description file	Provides the strings that will appear to the report author in the expression editor. In this example, only an English description file is included.

The functions are:

- A random integer generator
- A surface area calculator
- A date to a string converter

The following table shows the report function declaration and the function name and syntax that the report author sees in the expression editor.

Function Declaration in C++	Function Name and Syntax
<code>CCLDBColumnState SDKRandomInt (void* result, const crxDataI* context);</code>	<code>SDKRANDOMINT()</code>

Table 249. Report function declarations (continued)

Function Declaration in C++	Function Name and Syntax
CCLDBColumnState SDKArea (void* result, const crxDataI* context, const CCL_float64 arg1 const CCL_float64 arg2 const CCL_char arg3);	SDKAREA ([base],[height],[shape])
CCLDBColumnState SDKDateToString (void* result, const crxDataI* context const QDate* date	SDKDATETOSTRING(date)

Example of a custom file list file

The following is a segment of the custom file list.

```
<filelist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileList.xsd">
  <definition>crxSDKSampleTree.xml</definition>
  <description>
    <language>en-us</language>
    <file>crxSDKSampleStrings_en.xml</file>
  </description>
</filelist>
```

Example: crxSDKSampleTree.xml

The following is a segment of the function definition file showing the definition of the Area function.

```
...
<functionsRoot xsi:schemaLocation=
"http://www.developer.cognos.com/schemas/commonfunctionservice/1/0 FuncTree.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<group>
<id>SDKTEST</id>
<!--CRX SDK Test Functions-->
...

<function>
  <id>crx_sdk_area</id>
  <canonical>SDKArea</canonical>
  <name>SDKArea</name>
  <dll>CRXSampleSDKFunctions</dll>
  <context>CRX</context>
  <returnType>crxDTypeDouble</returnType>
  <parameter>
    <type>crxDTypeDouble</type>
  </parameter>
  <parameter>
    <type>crxDTypeDouble</type>
  </parameter>
  <parameter>
    <type>crxDTypeString</type>
  </parameter>
</function>
...
</group>
</functionsRoot>
```

Example: crxSDKSampleStrings_en.xml

The following is a segment of the English function description file showing the Area function.

```
...
<stringTable xsi:noNamespaceSchemaLocation="CCLMessageFile.xsd"
usage="String"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<component name="CRN">
```

```

<section name="TST">
...

<string id="crx_sdk_area">SDKAREA</string>
<string id="crx_sdk_area.syntax">SDKAREA(numeric_exp, numeric_exp, shape)</string>
<string id="crx_sdk_area.tip">Returns value of the area of a
Parallelepiped (P) or a Triangle (T).</string>
...
</section>
</component>
</stringTable>

```

Creating custom report functions

You can create custom report functions with the following steps.

Procedure

1. Define the functions (`crxSample.cpp`). The header file (`crxSDK.h`) is included in the program.
2. Compile the program into a dll, shared library or shared object file (`CRXSampleSDKFunctions.dll`).
3. Create a custom filelist (`filelist_custom.xml`).
4. Create a new function definition file (`crxSDKSAMPLETree.xml`).
5. Create a new function description file (`crxSDKSAMPLEStrings_en.xml`).

Custom function sets

A function set is a collection of database functions that are vendor-specific. You can customize the Function Description Service to expose or define functions that your data source supports. Defining additional vendor-specific functions is similar to defining custom functions. The following elements are used to specify the vendor group.

id

For a custom function set, the value of the `id` is `vendor_group`. Groups with the same identifier in the same position of the tree are merged.

context

Defining a context restricts the visibility of this group when the application specifies a particular context.

vendor

Identifier for the vendor. This restricts the visibility of this group when the application specifies a particular vendor.

datasourceQueryType

Underlying database query technology.

Creating a custom function set

You can create a custom function set with the following steps.

Procedure

1. Create a custom filelist named `filelist_customerExtensions.xml` for the vendor extensions.
2. Create a functions definition file named `customerExtensionsTree.xml` in the `installation_location/configuration/functions` directory. The contents of `customerExtensionsTree.xml` are validated against the `FuncTree.xsd` file.
3. Create a function description file named `customerExtensionsString_en.xml`.

Example of creating a custom function set

The following example shows how to add the datepart SQL Server function to the list of functions displayed in the expression editor. This function is not delivered as part of the default FDS content because there is an equivalent extract function in the SQL99 folder.

Create a filelist_customerExtensions.xml file

For adding custom function sets, a custom filelist is created the same way as for custom functions. A single custom filelist could be used for all your custom function definition and description files.

```
...
<filelist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileList.xsd">
  <definition>customerExtensionsTree.xml</definition>
  <description>
    <language>en-us</language>
    <file>customerExtensionsString_en.xml</file>
  </description>
</filelist>
```

Create a CustomerExtensionsTree.xml file

Create a customerExtensionsTree.xml file that contains the following content. Note that the file structure mimics the SQLServerTree.xml file, so that the content of folders can come from multiple files.

```
<?xml version="1.0" encoding="UTF-8"?>
<functionsRoot xsi:schemaLocation=
"http://www.developer.cognos.com/schemas/commonfunctionservice/1/0FuncTree.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<group>
<id>vendor_group</id>
<group><id>SQLServer</id>
  <context>tabular</context>
  <vendor>V_SQLServer</vendor>
  <datasourceQueryType>relational</datasourceQueryType>
</group>
```

```
<id>ext_sql</id>
<!--SQL Server Extensions. There are probably equivalent SQL99
functions available for these-->
<function>
  <id>ext_datepart</id>
  <canonical>datepart</canonical>
  <name>datepart</name>
  <returnType>numeric</returnType>
  <parameter>
    <type>string</type>
  </parameter>
  <parameter>
    <type>dateTime</type>
  </parameter>
</function>
</group>
</group>
</group>
</functionsRoot>
```

Create a CustomerExtensionsString_en.xml file

Create a customerExtensionsString_en.xml file that contains the following content. You can create additional description files for each language you want to support.

```
<?xml version="1.0" encoding="UTF-8"?>
<stringTable xsi:noNamespaceSchemaLocation="CCLMessageFile.xsd"
usage="String"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<component name="EXT">
<section name="EXT" type="UI">
<string id="ext_sql">Additional SQL Server Functions</string>
  <string id="ext_datepart">datepart</string>
  <string id="ext_datepart.syntax">
    datepart ( { year | month | day }, datetime_exp )
  </string>
  <string id="ext_datepart.tip">
    Returns the numeric value of part of a date.
    Note the { } braces around the first parameter.
  </string>
</section>
</component>
</stringTable>
```

Chapter 25. Using lineage specifications

Lineage is the process of determining how objects in a report, package, or model are derived through transformations from other objects, back to source data.

A lineage request specifies a report, package, or model for which lineage is desired. You must send lineage requests for report data to `reportService` (or `batchReportService`), followed by a subsequent request to `metadataService`; this process is described in [“Creating lineage requests for reports”](#) on page 1509. Lineage requests for a package or model are issued against `metadataService` (`reportService` and `batchReportService` are not involved); this process is described in [“Creating lineage requests for packages or models”](#) on page 1513.

Lineage responses from `metadataService` consist of a number of objects that contain metadata about the request subject. These objects can have parent-child references to other objects so the object history can be traced. In addition, the response contains details of any transformations that data objects have undergone. For information on `metadataService` lineage responses, see [“Understanding lineage responses from metadataService”](#) on page 1514.

The lineage API specification file, `lineage.xsd`, is located in `<install_location>/templates/mdsrv`. It is a generic specification for sending lineage requests to `metadataService`.

Creating lineage requests for reports

Beginning in IBM Cognos Analytics V10.1.0, you must issue lineage requests for report data against `reportService` or `batchReportService` to obtain `querySet` data, which you must then deliver to `metadataService` in a subsequent lineage request. Lineage information can be derived for both saved reports and for reports that you have not saved.

For saved report data, you can obtain lineage for data items and layout expressions that contain data items by issuing a `runSpecification(specification, parameterValues, options)` lineage request against `reportService` or `batchReportService`. The `runSpecification(specification, parameterValues, options)` method must include `metadataServiceLineageSpecification` as a specification and contain the search path. See for an example. Reports saved in earlier versions of IBM Cognos Analytics will only support lineage for data items; lineage for layout expressions is not supported.

For unsaved report data, such as during an interactive session, you can obtain lineage by issuing a secondary `lineage(conversation, parameterValues, options)` SOAP request against `reportService` or `batchReportService`. See the [Secondary Requests](#) chapter for more information.

For both saved reports and interactive sessions, the lineage `querySet` response from `reportService` (or `batchReportService`) must be encoded within a `V5QuerySet` object and forward it to `metadataService`, along with the `objectQuery` objects for which lineage information is desired. Requests to `metadataService` are issued using a `runSpecification(specification, parameterValues, options)` method running a `metadataServiceLineageSpecification` specification.

Sample request XML for a saved report

This sample XML contains a lineage request using a `runSpecification(specification, parameterValues, options)` method issued against `reportService` for data items in a saved **Order Invoices - Donald Chow, Sales Person** report. This report is contained in the **Reporting Report Samples** folder of the **Go Sales (query)** package included with the IBM Cognos Analytics samples.

```
<lineageRequest version="0.1">
  <connection>
    <param name="searchPath">
      defaultOutput(/content/folder[@name='Samples']/folder[@name='Models'])
    </param>
  </connection>
</lineageRequest>
```

```

        /package[@name='GO Sales (query)']/folder[@name='Reporting Report Samples']
        /report[@name='Order Invoices - Donald Chow, Sales Person'])
    </param>
</connection>
</lineageRequest>

```

The following sample XML represents the response received from reportService, including the XML querySet information contained in the asynchDetailMIMEAttachment property. Although the querySet response is XML-encoded, it is shown decoded here for illustration purposes.

```

<reportLineageResponse>
  <lineageResponse>
    <object id="[Report]">
      <name>Order Invoices - Donald Chow, Sales Person</name>
      <type>baseReport</type>
      <property name="objectType" displayName="Type">Report</property>
      <childRef>
        /content/folder[@name='Samples']/folder[@name='Models']
        /package[@name='GO Sales (query)']
      </childRef>
      <property name="description" displayName="Description">Generates
invoices of sales by Donald Chow.</property>
      <property name="owner" displayName="Owner">Anonymous</property>
    </object>
  </lineageResponse>
  <querySet expressionLocale="en-ca">
    <modelPath>
      /content/folder[@name='Samples']/folder[@name='Models']/package[@name='GO Sales
(query)']/model[@name='model']
    </modelPath>
    <queries xmlns="http://developer.cognos.com/schemas/report/7.0/">
      <query name="Query - Order Invoice">
        <source>
          <model/>
        </source>
        <selection>
          <dataItem aggregate="none" name="Order number" rollupAggregate="none"
sort="descending">
            <expression>[Sales (query)].[Order].[Order number]</expression>
          </dataItem>
          <dataItem aggregate="none" name="Order method"
rollupAggregate="automatic">
            <expression>[Sales (query)].[Order method].[Order method type]
</expression>
          </dataItem>
          <dataItem aggregate="none" name="Order date" rollupAggregate="none">
            <expression>[Sales (query)].[Time].[Date]</expression>
          </dataItem>
          <dataItem aggregate="none" name="Product name" rollupAggregate="none">
            <expression>[Sales (query)].[Products].[Product]</expression>
          </dataItem>
          <dataItem aggregate="none" name="Product number" rollupAggregate="none">
            <expression>[Sales (query)].[Products].[Product number]</expression>
          </dataItem>
          <dataItem aggregate="none" name="Description" rollupAggregate="none">
            <expression>[Sales (query)].[Products].[Product description]
</expression>
          </dataItem>
          <dataItem aggregate="none" name="Quantity" rollupAggregate="none">
            <expression>[Sales (query)].[Sales].[Quantity]</expression>
          </dataItem>
          <dataItem aggregate="none" name="Unit price" rollupAggregate="none">
            <expression>[Sales (query)].[Sales].[Unit price]</expression>
          </dataItem>
          <dataItem aggregate="total" name="Price" rollupAggregate="total">
            <expression>[Sales (query)].[Sales].[Quantity]*[Sales (query)].
[Sales].[Unit price]</expression>
          </dataItem>
          <dataItem name="Tax" rollupAggregate="total">
            <expression>([Sales (query)].[Sales].[Quantity]*[Sales (query)].
[Sales].[Unit price])*0.07</expression>
          </dataItem>
          <dataItem name="Shipping" rollupAggregate="total">
            <expression>([Sales (query)].[Sales].[Quantity]*[Sales (query)].
[Sales].[Unit price])*0.01</expression>
          </dataItem>
          <dataItem name="Total" rollupAggregate="total">
            <expression>([Sales (query)].[Sales].[Quantity]*[Sales (query)].
[Sales].[Unit price])*1.08</expression>
          </dataItem>
        </selection>
      </query>
    </queries>
  </querySet>

```



```

    <dataItem name="Staff name" rollupAggregate="none">
      <expression>[Sales (query)].[Sales staff].[Staff name]</expression>
    </dataItem>
  </selection>
  <detailFilters>
    <detailFilter use="required">
      <filterExpression>[Staff name] = 'Donald Chow'</filterExpression>
    </detailFilter>
  </detailFilters>
</query>
</queries>
<queryResultDefinitions>
  <queryResultDefinition name="Query - Order Invoice.0"
    refQuery="Query - Order Invoice">
    <edges>
      <edge name="3">
        <edgeGroups>
          <edgeGroup>
            <valueSets>
              <valueSet name="7" refDataItem="Order number">
                <groupHeader name="7_groupHeader"/>
                <groupFooter name="7_groupFooter"/>
                <propertyExpressions>
                  <propertyExpression>RoleValue('_memberUniqueName')</propertyExpression>
                </propertyExpressions>
              </valueSet>
            </valueSets>
          </edgeGroup>
          <edgeGroup>
            <valueSets>
              <valueSet name="11">
                <groupHeader name="11_groupHeader">
                  <dataItemRef refDataItem="Staff name"/>
                  <dataItemRef refDataItem="Order method"/>
                  <dataItemRef refDataItem="Order date"/>
                </groupHeader>
                <groupBody name="11_groupBody">
                  <dataItemRef refDataItem="Product name"/>
                  <dataItemRef refDataItem="Product number"/>
                  <dataItemRef refDataItem="Description"/>
                  <dataItemRef refDataItem="Quantity"/>
                  <dataItemRef refDataItem="Unit price"/>
                  <dataItemRef refDataItem="Price"/>
                  <propertyExpressions>
                    <propertyExpression>RoleValue('_memberUniqueName')
                  </propertyExpression>
                </propertyExpressions>
                </groupBody>
                <groupFooter name="11_groupFooter">
                  <dataItemRef refDataItem="Price"/>
                  <dataItemRef refDataItem="Tax"/>
                  <dataItemRef refDataItem="Shipping"/>
                  <dataItemRef refDataItem="Total"/>
                </groupFooter>
              </valueSet>
            </valueSets>
          </edgeGroup>
        </edgeGroups>
      </edge>
    </edges>
  </queryResultDefinition>
</queryResultDefinitions>
</querySet>
</reportLineageResponse>

```

The following sample XML contains a lineage request using a `runSpecification` (specification, parameterValues, options) method issued against `metadataService`, consisting of a `V5QuerySet` object and a set of `objectQuery` objects. The `V5QuerySet` contains the XML-encoded `querySet` response from `reportService`. Each `objectQuery` contains a set of `param` elements that leads to a given object in the specification.

You can specify the following types of `objectQuery` objects:

- A single `param` named `modelItemRef` whose value is a reference to an object or member in the package model.

- Two `param` objects that lead to an item in the query section of an authoredReport object. The first `param` is named `queryName` and its value is the name of the query object containing the item, and the second `param` is named `dataItemName` and its value is the name of the data item.

Note: The `V5QuerySet` object was added in IBM Cognos Analytics V10.1.0 to enhance lineage reporting capabilities. In previous versions of IBM Cognos Analytics, the `connection` object was used. For report data lineage requests, the `connection` object was deprecated in IBM Cognos Analytics V10.1.0; backwards compatibility is maintained.

```
<lineageRequest>
  <V5QuerySet>&lt;querySet expressionLocale=&quot;en-ca&quot;&gt;&lt;modelPath&gt;
    /content/folder[@name=&apos;Samples&apos;]/folder[@name=&apos;Models&apos;]
    /package[@name=&apos;GO Sales (query)&apos;]/model[@name=&apos;model&apos;]&lt;
    /modelPath&gt;&lt;queries
    xmlns=&quot;http://developer.cognos.com/schemas/report/7.0/&quot;
    &gt;&lt;query name=&quot;Query - Order Invoice&quot;&gt;&lt;source&gt;&lt;
    model/&gt;&lt;/source&gt;&lt;selection&gt;&lt;dataItem aggregate=&quot;none&quot;
    name=&quot;Order number&quot; rollupAggregate=&quot;none&quot;
    sort=&quot;descending&quot;&gt;&lt;expression&gt;
    [Sales (query)].[Order].[Order number]&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem aggregate=&quot;none&quot;
    name=&quot;Order method&quot; rollupAggregate=&quot;automatic&quot;&gt;&lt;
    expression&gt;[Sales (query)].[Order method].[Order method]&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem aggregate=&quot;none&
    quot; name=&quot;Order date&quot; rollupAggregate=&quot;none&quot;&gt;&lt;
    expression&gt;[Sales (query)].[Time].[Date]&lt; /expression&gt;&lt;/dataItem&gt;
    &lt;dataItem aggregate=&quot;none&quot; name=&quot;Product name&quot;
    rollupAggregate=&quot;none&quot;&gt;&lt;expression&gt;
    [Sales (query)].[Products].[Product]&lt;/expression&gt;&lt;/dataItem&gt;&lt;
    dataItem aggregate=&quot;none&quot; name=&quot;Product number&quot;
    rollupAggregate=&quot;none&quot;&gt;&lt;expression&gt;
    [Sales (query)].[Products].[Product number]&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem aggregate=&quot;none&quot;
    name=&quot;Description&quot; rollupAggregate=&quot;none&quot;&gt;&lt;
    expression&gt;[Sales (query)].[Products].[Product description]&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem aggregate=&quot;none&quot;
    name=&quot;Quantity&quot; rollupAggregate=&quot;none&quot;&gt;&lt;
    expression&gt;[Sales (query)].[Sales].[Quantity]&lt;/expression&gt;&lt;
    /dataItem&gt;&lt;dataItem aggregate=&quot;none&quot; name=&quot;
    Unit price&quot; rollupAggregate=&quot;none&quot;&gt;&lt;expression&gt;
    [Sales (query)].[Sales].[Unit price]&lt;/expression&gt;&lt;/dataItem&gt;&lt;
    dataItem aggregate=&quot;total&quot; name=&quot;Price&quot;
    rollupAggregate=&quot;total&quot;&gt;&lt;expression&gt;
    [Sales (query)].[Sales].[Quantity]*[Sales (query)].[Sales].[Unit price]&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem name=&quot;Tax&quot;
    rollupAggregate=&quot;total&quot;&gt;&lt;expression&gt;
    ([Sales (query)].[Sales].[Quantity]*[Sales (query)].[Sales].[Unit price])*0.07&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem name=&quot;Shipping&quot;
    rollupAggregate=&quot;total&quot;&gt;&lt;expression&gt;
    ([Sales (query)].[Sales].[Quantity]*[Sales (query)].[Sales].[Unit price])*0.01&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem name=&quot;Total&quot;
    rollupAggregate=&quot;total&quot;&gt;&lt;expression&gt;
    ([Sales (query)].[Sales].[Quantity]*[Sales (query)].[Sales].[Unit price])*1.08&lt;
    /expression&gt;&lt;/dataItem&gt;&lt;dataItem
    name=&quot;Staff name&quot; rollupAggregate=&quot;none&quot;&gt;&lt;
    expression&gt;[Sales (query)].[Sales staff].[Staff name]&lt;/expression&gt;&lt;
    /dataItem&gt;&lt;/selection&gt;&lt;detailFilters&gt;&lt;detailFilter
    use=&quot;required&quot;&gt;&lt;filterExpression&gt;[Staff name] = &apos;
    Donald Chow&apos;&lt; /filterExpression&gt;&lt;/detailFilter&gt;&lt;
    /detailFilters&gt;&lt;/query&gt;&lt;/queries&gt;&lt;queryResultDefinitions&gt;&lt;
    queryResultDefinition name=&quot;Query - Order Invoice.0&quot; refQuery=&quot;
    Query - Order Invoice&quot;&gt;&lt;edges&gt;&lt;edge name=&quot;3&quot;&gt;&lt;
    edgeGroups&gt;&lt;edgeGroup&gt;&lt;valueSets&gt;&lt;valueSet name=&quot;7&quot;
    refDataItem=&quot;Order number&quot;&gt;&lt;groupHeader name=&quot;
    7_groupHeader&quot;/&gt;&lt;groupFooter name=&quot;
    7_groupFooter&quot;/&gt;&lt;propertyExpressions&gt;&lt;propertyExpression&gt;
    RoleValue (&apos;_memberUniqueName&apos;)&lt;/propertyExpression&gt;&lt;
    /propertyExpressions&gt;&lt;/valueSet&gt;&lt;/valueSets&gt;&lt;
    edgeGroups&gt;&lt;edgeGroup&gt;&lt;valueSets&gt;&lt;valueSet
    name=&quot;11&quot;&gt;&lt;groupHeader name=&quot;
    11_groupHeader&quot;&gt;&lt;dataItemRef
    refDataItem=&quot;Staff name&quot;/&gt;&lt;dataItemRef
    refDataItem=&quot;Order method&quot;/&gt;&lt;dataItemRef
    refDataItem=&quot;Order date&quot;/&gt;&lt;/groupHeader&gt;&lt;groupBy
    name=&quot;11_groupBody&quot;&gt;&lt;dataItemRef
    refDataItem=&quot;Product name&quot;/&gt;&lt;dataItemRef
    refDataItem=&quot;Product number&quot;/&gt;&lt;dataItemRef
    refDataItem=&quot;Description&quot;/&gt;&lt;dataItemRef
```

```

refDataItem=&quot;Quantity&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Unit price&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Price&quot;/&gt;&lt;/propertyExpressions&gt;&lt;/
propertyExpression&gt;RoleValue(&apos;_memberUniqueName&apos;)&lt;/
/propertyExpression&gt;&lt;/propertyExpressions&gt;&lt;/groupBody&gt;&lt;/
groupFooter name=&quot;11_groupFooter&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Price&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Tax&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Shipping&quot;/&gt;&lt;/dataItemRef
refDataItem=&quot;Total&quot;/&gt;&lt;/groupFooter&gt;&lt;/
/valueSet&gt;&lt;/valueSets&gt;&lt;/edgeGroup&gt;&lt;/
/edgeGroups&gt;&lt;/edgeGroup&gt;&lt;/edgeGroups&gt;&lt;/
/edge&gt;&lt;/edges&gt;&lt;/queryResultDefinition&gt;&lt;/
/queryResultDefinitions&gt;&lt;/querySet&gt;</V5QuerySet>
<objectQuery id="1">
  <param name="queryName">Query - Order Invoice</param>
  <param name="dataItemName">Product name</param>
</objectQuery>
<objectQuery id="2">
  <param name="queryName">Query - Order Invoice</param>
  <param name="dataItemName">Quantity</param>
</objectQuery>
</lineageRequest>

```

Creating lineage requests for packages or models

Lineage requests for packages and models are issued against `metadataService` using a `runSpecification(specification, parameterValues, options)` method running a `metadataServiceLineageSpecification` specification. The lineage request portion consists of a connection object and a set of `objectQuery` objects. The connection contains a set of name-value pairs that point to a specification that the service operates on, and each `objectQuery` contains a set of param elements that leads to a given object (such as a `dataItem`) in the specification. The service provider defines the domain of valid parameter values.

The `connection` object of the request must contain a single `param` object whose name is `searchPath` and whose value is a search path that refers to one of the following types of objects:

- `baseReport`
- `reportVersion`
- `output`
- `package`
- `model`

If a package is specified that has model versioning enabled, the model object that represents the last model version is used.

You can specify the following types of `objectQuery` objects.

- A single `param` named `modelItemRef` whose value is a reference to an object or member in the package model.
- Two `param` objects that lead to an item in the query section of an `authoredReport` object. The first `param` is named `queryName` and its value is the name of the query object containing the item, and the second `param` is named `dataItemName` and its value is the name of the data item.

Sample request XML for package data

This sample XML contains a lineage request to `metadataService`, using the connection object, for the **Product number** and **Quantity** items in the **Order Invoices - Donald Chow, Sales Person** report. This report is contained in the **Reporting Report Samples** folder of the **Go Sales (query)** package included with the IBM Cognos Analytics samples. The search path to the report is specified in the connection object. Two data items **Product number** and **Quantity**, both of which belong to the **Query – Order Invoice** query in the report specification, are specified as separate `objectQuery` objects.

```

<lineageRequest version="0.1">
  <connection>

```

```

    <param name="searchPath">/content/package[@name='GO Sales (query)']
    /folder[@name='Reporting Report Samples']
    /report[@name='Order Invoices - Donald Chow, Sales Person']</param>
</connection>
<objectQuery id="1">
  <param name="queryName">Query - Order Invoice</param>
  <param name="dataItemName">Product number</param>
</objectQuery>
<objectQuery id="2">
  <param name="queryName">Query - Order Invoice</param>
  <param name="dataItemName">Quantity</param>
</objectQuery>
</lineageRequest>

```

Understanding lineage responses from metadataService

The metadataService lineage response consists of a `LineageResponse` root element that contains a collection of `queryResult` elements and a set of objects. Each `queryResult` refers back to an `objectQuery` in the originating request and contains references to objects that serve as starting points for the lineage traversal.

The metadataService will return a single `queryResult` instance for each `objectQuery` specified in the request. If an expression is specified in the `objectQuery`, then an object will exist in the response for each object referenced in the expression, and the `queryResult` item will contain an `objectRef` reference to each of these objects. If an authoredReport data item is specified then a single object will appear in the response that corresponds to that item, and this object will be referenced in `queryResult`.

The response can contain objects of various types, including those from the Framework Manager model schema and from content store objects. A listing of source types, object types, property types, and source mappings is shown in [“Source mappings” on page 1520](#).

Each object has an `id` attribute that serves as a logical key for the object and must be unique. An object has `name`, `type`, and `property` elements that describe the object. The object can have a `transformation` that has a `type` and can have additional `property` elements that describe the transformation. The `transformation` can also contain transformation sources, which can be references to other objects (that may or may not have their own transformation) in the response. A transformation source can also be a new lineage request that the caller can invoke to get further lineage information. Finally, an object can have references to a parent and/or multiple child objects in the response, which in turn can have their own parents/children. This helps describe the objects by giving them additional context.

Portions of the corresponding metadata service response (to [“Sample request XML for package data” on page 1513](#)) are shown here. The metadata service response for report data is similar in structure and content, but is excluded for brevity. A separate `queryResult` appears for each `objectQuery` in the request. The first `queryResult` (in Query Results) begins with a reference to an object in the response whose id is `[Report].[Query - Order Invoice].[Product number]` (Object A).

Query Results

```

<queryResult>
  <objectRef>[Report].[Query - Order Invoice].[Product number]</objectRef>
  <objectQueryRef>1</objectQueryRef>
</queryResult>
<queryResult>
  <objectRef>[Report].[Query - Order Invoice].[Quantity]</objectRef>
  <objectQueryRef>2</objectQueryRef>
</queryResult>

```

Object A

```

<object id="[Report].[Query - Order Invoice].[Product number]">
  <name>Product number</name>

```

```

<type>dataItem</type>
<property name="objectId" displayName="ID">
  [Query - Order Invoice].[Product number]
</property>
<property name="objectName" displayName="Name">Product number</property>
<property name="objectType" displayName="Type" displayValue="Data Item">query
</property>
<property name="expression" displayName="Expression">
  [Sales (query)].[Product].[Product number]
</property>
<transformation>
  <type>objectReference</type>
  <property name="expression" displayName="Expression">
    [Sales (query)].[Product].[Product number]
  </property>
  <objectRef>[Sales (query)].[Product].[Product number]</objectRef>
</transformation>
<parentRef>[Report].[Query - Order Invoice]</parentRef>
</object>

```

Object A has a transformation based on an expression with a transformation source reference to the object whose id is [Sales (query)].[Product].[Product number] (Object B).

Object B

```

<object id="[Sales (query)].[Product].[Product number]">
  <name>Product number</name>
  <type>queryItem</type>
  <property name="objectId" displayName="ID">
    [Sales (query)].[Product].[Product number]
  </property>
  <property name="objectName" displayName="Name">Product number</property>
  <property name="objectType" displayName="Type">Query Item</property>
  <property name="expression" displayName="Expression">
    [gosales].[PRODUCT].[PRODUCT_NUMBER]
  </property>
  <property name="datatype" displayName="Datatype">Int 32</property>
  <property name="precision" displayName="Precision">0</property>
  <property name="scale" displayName="Scale">0</property>
  <property name="size" displayName="Size">4</property>
  <property name="regularAggregate" displayName="Regular Aggregate">Count</property>
  <property name="semiAggregate" displayName="Semi Aggregate">Unsupported Value
  </property>
  <transformation>
    <type>objectReference</type>
    <property name="expression" displayName="Expression">
      [gosales].[PRODUCT].[PRODUCT_NUMBER]
    </property>
    <objectRef>[gosales].[PRODUCT].[PRODUCT_NUMBER]</objectRef>
  </transformation>
  <parentRef>[Sales (query)].[Product]</parentRef>
</object>

```

Object B in turn has a transformation that is based on an expression whose transformation source is a lineage object whose id is [gosales].[PRODUCT].[PRODUCT_NUMBER], which represents another queryItem in the Framework Manager package model (Object C).

Object C

```

<object id="[gosales].[PRODUCT].[PRODUCT_NUMBER]">
  <name>PRODUCT_NUMBER</name>
  <type>queryItem</type>
  <property name="objectId" displayName="ID">[gosales].[PRODUCT].[PRODUCT_NUMBER]
  </property>
  <property name="objectName" displayName="Name">PRODUCT_NUMBER</property>
  <property name="objectType" displayName="Type">Query Item</property>
  <property name="externalName" displayName="External Name">PRODUCT_NUMBER</property>
  <property name="datatype" displayName="Datatype">Int 32</property>
  <property name="precision" displayName="Precision">0</property>
  <property name="scale" displayName="Scale">0</property>
  <property name="size" displayName="Size">4</property>
  <property name="regularAggregate" displayName="Regular Aggregate">Count
  </property>

```

```

<property name="semiAggregate" displayName="Semi Aggregate">Unsupported Value
</property>
<transformation>
  <type>transformation</type>
  <property name="externalName" displayName="External Name">PRODUCT_NUMBER
  </property>
  <objectRef>[].[dataSources].[gosales]</objectRef>
</transformation>
<parentRef>[gosales].[PRODUCT]</parentRef>
</object>

```

As this queryItem represents a column in a data source, Object C has a transformation whose transformation source is a lineage object with id [] . [dataSources] . [gosales] that represents the underlying data source object in the Framework Manager model (Object D).

Object D

```

<object id="[].[dataSources].[gosales]">
  <name>gosales</name>
  <type>dataSource</type>
  <property name="objectId" displayName="ID">[].[dataSources].[gosales]</property>
  <property name="objectName" displayName="Name">gosales</property>
  <property name="objectType" displayName="Type">Data Source</property>
  <property name="cmDataSource" displayName="Cognos Connection Data Source">great_outdoors_sales</property>
  <property name="catalog" displayName="Catalog"/>
  <property name="schema" displayName="Schema">GOSALES</property>
  <parentRef>/content/package[@name='GO Sales (query)']</parentRef>
</object>

```

Traversing the parentRef of leads to [Report] . [Query - Order Invoice] (Object E), a lineage object pertaining to the query in the report, and then to a lineage object that pertains to the report itself (Object F).

Object E

```

<object id="[Report].[Query - Order Invoice]">
  <name>Query - Order Invoice</name>
  <type>query</type>
  <property name="objectId" displayName="ID">[Query - Order Invoice]</property>
  <property name="objectName" displayName="Name">Query - Order Invoice</property>
  <property name="objectType" displayName="Query" displayValue="Query">dataItem
  </property>
  <parentRef>/content/package[@name='GO Sales (query)']/folder[@name='Reporting Report Samples']/report[@name='Order Invoices - Donald Chow, Sales Person']</parentRef>
  <childRef>[Report].[Query - Order Invoice].[Product number] </childRef>
  <childRef>[Report].[Query - Order Invoice].[Quantity]</childRef>
</object>

```

Object F

```

<object id="/content/package[@name='GO Sales (query)']/folder[@name='Reporting Report Samples']/report[@name='Order Invoices - Donald Chow, Sales Person']">
  <name>Order Invoices - Donald Chow, Sales Person</name>
  <type>auhoredReport</type>
  <property name="objectId" displayName="ID">/content/package[@name='GO Sales (query)']/folder[@name='Reporting Report Samples']/report[@name='Order Invoices - Donald Chow, Sales Person']</property>
  <property name="objectName" displayName="Name">GO Sales (query)</property>
  <property name="objectType" displayName="Type" displayValue="Report">report
  </property>
  <property name="owner" displayName="Owner">Anonymous</property>
  <childRef>/content/package[@name='GO Sales (query)']</childRef>
  <childRef>[Report].[Query - Order Invoice]</childRef>
</object>

```

Traversing the `parentRef` of “Object B” on page 1515 leads to a lineage object that represents the `[Sales (query)]. [Product]` query subject in the Framework Manager model (Object G).

Object G

```
<object id="[Sales (query)]. [Product]">
  <name>Product</name>
  <type>querySubject</type>
  <property name="objectId" displayName="ID">[Sales (query)]. [Product]</property>
  <property name="objectName" displayName="Name">Product</property>
  <property name="objectType" displayName="Type">Query Subject</property>
  <property name="definitionType" displayName="Definition Type">Model</property>
  <property name="sql" displayName="SQL">with...</property>
  <parentRef>[Sales (query)]</parentRef>
  <childRef>[Sales (query)]. [Product]. [Product number]</childRef>
</object>
```

The `parentRef` of Object G is an object representing the "[Sales (query)]" namespace in the Framework Manager model (Object H). Object H has a `parentRef` that leads to a lineage object (Object I) corresponding to the package. Object I in turn has a `parentRef` whose value is the lineage object pertaining to the report (“Object F” on page 1516).

Object H

```
<object id="[Sales (query)]">
  <name>Sales (query)</name>
  <type>namespace</type>
  <property name="objectId" displayName="ID">[Sales (query)]</property>
  <property name="objectName" displayName="Name">Sales (query)</property>
  <property name="objectType" displayName="Type">Namespace</property>
  <parentRef>/content/package[@name='GO Sales (query)']</parentRef>
  <childRef>[Sales (query)]. [Sales]</childRef>
  <childRef>[Sales (query)]. [Product]</childRef>
</object>
```

Object I

```
<object id="/content/package[@name='GO Sales (query)']">
  <name>GO Sales (query)</name>
  <type>package</type>
  <property name="objectId" displayName="ID">/content/package[@name='GO
Sales (query)']</property>
  <property name="objectName" displayName="Name">GO Sales (query)</property>
  <property name="objectType" displayName="Type" displayValue="Package">package
</property>
  <property name="description" displayName="Description">Package
based on relational view.</property>
  <property name="owner" displayName="Owner">Anonymous</property>
  <property name="version" displayName="Model version">1</property>
  <parentRef>/content/package[@name='GO Sales (query)']/folder[@name='Reporting Report
Samples']/report[@name='Order Invoices - Donald Chow,
Sales Person']</parentRef>
  <childRef>[Sales (query)]. [gosales]</childRef>
  <childRef>[gosales]</childRef>
  <childRef>[Sales (query)]</childRef>
</object>
```

Integrating a custom lineage solution

To replace the default IBM Cognos lineage with a different lineage solution, you must provide a Web interface reachable by a URL that can receive the lineage parameters from IBM Cognos and translate the parameters to invoke the custom lineage solution. You can use a technology such as Python, JSP, or CGI to build the Web interface.

The IBM Cognos lineage request parameters are:

- `selectionContext`

An XML value that conforms to the selection schema of the object model (see for more information). This parameter is used when lineage is accessed in IBM Cognos Viewer.

Here is an example:

```
<s:selection
rModel="MP_0" xmlns:s="http://developer.cognos.com/schemas/selection/1/"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
rSelectedCells="C_0">
  <s:metadataCells>
    <s:metadataCell xml:id="M_0" rQueryName="QN_0" rName="N_0" rUsage="U_0"/>
  </s:metadataCells>
  <s:cells>
    <s:cell xml:id="C_0" rMetadataCell="M_0" rValue="N_0" display="Sales">
    </s:cell>
  </s:cells>
  <s:strings>
    <s:s xml:id="QN_0">Query1</s:s>
    <s:s xml:id="N_0">Sales</s:s>
    <s:s xml:id="MP_0">/content/package[@name='Widgets &apos;']/
report[@name='WidgetSales&apos;]</s:s>
  </s:strings>
</s:selection
```

In this example, the `rSelectedCells` attribute indicates the cell selected in a report, which is `C_0`. Finding the cell with this id yields the cell whose metadata cell is `M_0`. The metadata cell `M_0` has a query name of `QN_0`, which is an entry into the string pool yielding `Query1`. The name of the cell is `N_0`, which is also an entry into the string table yielding `Sales`. The report upon which the selection was based can be retrieved by unwinding the `rModel` attribute of the selection in the select `[MP_0]` to its entry in the string table, `/content/package[@name='Widgets']/report[@name='WidgetSales']`.

- `lineageRequest`

An XML value that is used when lineage is accessed from the studio's metadata tree.

Here is an example:

```
<lineageInfo>
  <packagePath>/content/packagePath</packagePath>
  <item queryRef="[optional]">[an].[fm].[id]</item>
</lineageInfo>
```

This parameter consists of the `packagePath` element containing the package path of the model, and one or more `item` elements with a `queryRef` attribute whose value is a Framework Manager ID.

- `executionParms`

An xml value that captures any resolved prompts or data source connections. When viewed in IBM Cognos Viewer, this value may be signed by IBM Cognos Application Firewall (CAF), and it is not decodable for the purposes of custom lineage.

After the Web interface is created, use its URL to configure lineage in IBM Cognos. See the *Administration and Security Guide* for information on how to configure lineage.

Example: Creating a Web interface to customize the lineage request parameters

A company named Widget Co has purchased a lineage solution from a provider named LineageRUS to provide lineage capabilities for all company reports and packages, including the IBM Cognos reports. The LineageRUS server is installed on the Widget Co network, and it is accessible at the following URL:

<http://widget-co/lineagerus/>

LineageRUS activates its lineage by passing a number of URL parameters. For example, in the report named WidgetSales, referenced in the following sample, the URL passed to LineageRUS for the selection of Sales would be:

```
http://widget-co/lineagerus?reportSearchPath=/content/public/Widgets/WidgetSales&reportQuery='Query1'&reportColumn='Sales'
```

Assuming that the Sales report is based on the package named Widgets, and that the Framework Manager model contains a query item with the ID [Widgets].[Sales].[Sales], the LineageRUS URL in IBM Cognos would become:

```
http://widget-co/lineagerus?packageSearchPath=/content/public/Widgets/WidgetSales&fmID=' [Widgets].[Sales].[Sales]'
```

To invoke LineageRUS from IBM Cognos, Widget Co creates a Web interface that translates IBM Cognos lineage request parameters into the LineageRUS request parameters. This Web interface is accessible from the Widget Co network, and is invoked by IBM Cognos Viewer and the studios using the IBM Cognos lineage request parameters. The following URL is chosen for this interface:

```
http://widget-co/urlTranslation
```

This URL is used in IBM Cognos to configure the lineage solution.

Following is a sample of what this Web interface may look like if implemented as a Java EE Servlet:

```
public class IBMCognosToLineageRUS
extends HttpServlet
{
    protected void doPost(HttpServletRequest req, HttpServletResponse
resp){
        Enumeration enum = req.getParameterNames();
        String lineageRUSURL=null;
        while (enum.hasMoreElements()){
            String pName = (String)enum.nextElement();
            if ("selectioncontext".equals(pName)){
                lineageRUSURL=extractFromSelectionContext(req.getParameter(pName));
                break;
            }
            else if ("lineagerequest".equals(pName)){
                lineageRUSURL=extractFromLineageRequest(req.getParameter(pName));
                break;
            }
        }
        if (lineageRUSURL != null){
            try{
                //send back a redirect page to the application to redirect to the
                LineageRUS application with the required parameters
                resp.getWriter().write("<html><head>");
                resp.getWriter().write("<meta http-equiv=\"Refresh\" content=\"0;
url=\"");
                resp.getWriter().write(lineageRUSURL);
                resp.getWriter().write("\");
                resp.getWriter().write("</head><body></body></html>");
            }
            catch (IOException io){
            }
        }
    }
    /** Parses an IBM Cognos 8 selection context,
    and extracts the report path, the query, and query item name.
    */
    private String extractFromSelectionContext(String selectionContext){
        StringBuffer url = new StringBuffer("http://widget-co/lineagerus?");
        try{
            Document d = DocumentHelper.parseText(selectionContext);
            Element rt = d.getRootElement();
            String tblId = rt.attributeValue("rModel");
            String report = d.selectSingleNode("//*[ @id='"+tblId+"' ]").getText();
            String selectedCell = rt.attributeValue("rSelectedCells");
            Element cell = (Element)d.selectSingleNode("//*[ @id='"+selectedCell+"' ]");
            String metadataCellId = cell.attributeValue("rMetadataCell");
            Element metdataCell =
            (Element)d.selectSingleNode("//*[ @id='"+metadataCellId+"' ]");
            String queryNameId = metdataCell.attributeValue("rQueryName");
            String rNameId = metdataCell.attributeValue("rName");
            String query = d.selectSingleNode("//*[ @id='"+queryNameId+"' ]").getText();
        }
    }
}
```

```

        String queryItem = d.selectSingleNode("//*[id='"+rNameId+"']").getText();
        url.append("reportSearchPath=");
        url.append(report);
        url.append("&reportQuery=");
        url.append(query);
        url.append("&reportColumn=");
        url.append(queryItem);
    }
    catch (DocumentException de){
    }
    return url.toString();
}
/** Extracts the package name, and Framework Manager IDs from
the lineage request xml.
*/
private String extractFromLineageRequest(String selectionContext){
    StringBuffer url = new StringBuffer("http://widget-co/lineagerus?");
    try{
        Document d = DocumentHelper.parseText(selectionContext);
        Element rt = d.getRootElement();
        String pkgPath = d.selectSingleNode("//reportPath").getText();
        String fmId = d.selectSingleNode("//item").getText();
        url.append("packageSearchPath=");
        url.append(pkgPath);
        url.append("&fmID=");
        url.append(fmId);
    }
    catch (DocumentException de){
    }
    return url.toString();
}
}
}
}

```

Security considerations

Note the following security considerations.

- The `canUseLineage` capability must be granted for the caller and for the package.
- The caller must have read access to the `authoredReport` or package.
- If object security has been specified in the published Framework Manager model specification, then users will not see objects in the response for which they are not granted access.

Also note the following limitations before using lineage.

- Evaluation of macros contained within expressions is not currently supported.
- The `metadataService` implementation reports lineage back to the data source in the Framework Manager model but no further.
- The caller must resolve any required model prompts, and data source connection and signon parameters that may be encountered beforehand and specify them in the parameters section of the request.
- The response is returned by default in base64 encoding, although support for MIME and MIME Compressed encodings are available.
- The response does not preserve the namespace hierarchy from the originating Framework Manager model in the context hierarchy of the lineage objects. Essentially, the namespaces are flattened.

Source mappings

The response can contain objects of various types, including those from the Framework Manager model schema and some of those from the content store. A listing of source types, object types, property types, and source mappings is shown here.

Source Type - Content Store Objects

- Property types common to all object types.

<i>Table 250. Source mappings for property types common to all object types</i>	
Property Type	Source Mapping
objectId	The searchPath of the object in the content store.
objectName	The name of the object.
description	The value of the defaultDescription property of the authoredReport object.
screenTip	The value of the defaultScreenTip property of the authoredReport object.
contact	The value of the defaultName property of the object referenced by the contact property.
contactEMail	The value of the contactEMail property.
owner	The value of the defaultName property of the account object referenced by the owner property.
version	The version of the report that pertains to the request. If a report or reportView was specified in the request, then it will be that of the last report version.

- Object Type - baseReport

<i>Table 251. Source mappings for objectType baseReport</i>	
Property Type	Source Mapping
objectType	The concrete type of the baseReport object. If the object is a reportView, then it is the type of the authoredReport object (report, query, or analysis) being referenced.

- Object Type - package

<i>Table 252. Source mappings for objectType package</i>	
Property Type	Source Mapping
objectType	Same as intrinsic type, but includes a localized value.
version	The version number of the particular model object that pertains to the request. If a package was specified in the request, then it will be that of the last model version.

Source Type - authoredReport Query Specification

- Property types common to all object types.

Table 253. Source mappings for property types common to all object types

Property Type	Source Mapping
objectId	A reference to the object in the query specification of the containing authoredReport object.
objectName	The name of the object.
objectType	Same as intrinsic type, but includes a localized value.
description	The description of the object.
screenTip	The screenTip of the object.

- Object Type - query

Table 254. Source mappings for objectType query

Property Type	Source Mapping
definitionType	An enumeration that contains the type of child element contained in the definition property of the source Framework Manager model object.
sql	The value of the sql enumeration property of the source Framework Manager model object.
tableType	The value of the tableType enumeration property of the source Framework Manager model object.
canonicalName	The value of the canonicalName property of the source Framework Manager model object.

- Object Type - dataItem

Table 255. Source mappings for objectType dataItem

Property Type	Source Mapping
expression	The value of the expression property of the source Framework Manager model object.
externalName	The value of the externalName property of the source Framework Manager model object.

- Object Type - detailFilter

Table 256. Source mappings for objectType detailFilter

Property Type	Source Mapping
expression	The value of the expression property of the source Framework Manager model object.

- Object Type - summaryFilter

<i>Table 257. Source mappings for objectType summaryFilter</i>	
Property Type	Source Mapping
filterExpression	The value of the expression property of the source Framework Manager model object.

- Object Type - slicerMemberSet

<i>Table 258. Source mappings for objectType slicerMemberSet</i>	
Property Type	Source Mapping
expression	The value of the slicerMemberSet property of the source Framework Manager model object.

Source Type - Framework Manager Model Schema Specification

- Property types common to all object types.

<i>Table 259. Source mappings for property types common to all object types</i>	
Property Type	Source Mapping
objectId	A reference to the object in the query specification of the containing authoredReport object.
objectName	The name of the object in the caller's content locale.
objectType	Same as intrinsic type, but includes a localized value.

- Object Types:
 - querySubject
 - updateSubject

<i>Table 260. Source mappings for objectTypes querySubject and updateSubject</i>	
Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
definitionType	An enumeration that contains the type of child element contained in the definition property of the source Framework Manager model object.
sql	The value of the sql property of the source Framework Manager model object.
tableType	The value of the tableType enumeration property of the source Framework Manager model object.
canonicalName	The value of the canonicalName property of the source Framework Manager model object.

- Object Type - dimension

<i>Table 261. Source mappings for objectType dimension</i>	
Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
externalName	The value of the externalName property of the source Framework Manager model object.
dimensionType	The value of the type enumeration property of the Framework Manager model object.
definitionType	An enumeration that contains the type of child element contained in the definition property of the source Framework Manager model object.
sql	The value of the sql property of the source Framework Manager model object.
tableType	The value of the tableType enumeration property of the source Framework Manager model object.

- Object Types:

- queryItem
- measure

<i>Table 262. Source mappings for objectTypes queryItem and measure</i>	
Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
expression	The value of the expression property of the source Framework Manager model object.
externalName	The value of the externalName property of the source Framework Manager model object.
datatype	The value of the datatype enumeration property of the source Framework Manager model object.
precision	The value of the precision property of the source Framework Manager model object.
scale	The value of the scale property of the source Framework Manager model object.

Table 262. Source mappings for objectTypes queryItem and measure (continued)

Property Type	Source Mapping
size	The value of the size property of the source Framework Manager model object.
regularAggregate	The value of the regularAggregate enumeration property of the source Framework Manager model object.
semiAggregate	The value of the semiAggregate enumeration property of the source Framework Manager model object.

- Object Types:
 - calculation

Table 263. Source mappings for objectType calculation

Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
expression	The value of the expression property of the source Framework Manager model object.
externalName	The value of the externalName property of the source Framework Manager model object.
datatype	The value of the datatype enumeration property of the source Framework Manager model object.
precision	The value of the precision property of the source Framework Manager model object.
scale	The value of the scale property of the source Framework Manager model object.
size	The value of the size property of the source Framework Manager model object.
regularAggregate	The value of the regularAggregate enumeration property of the source Framework Manager model object.
semiAggregate	The value of the semiAggregate enumeration property of the source Framework Manager model object.
calcType	The value of the source Framework Manager model object's calcType property.

- Object Types:
 - queryItemFolder
 - folder

- namespace
- level
- measureFolder
- hierarchyFolder

Table 264. Source mappings for objectTypes queryItemFolder, folder, namespace, level, measureFolder, hierarchyFolder

Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.

- Object Type - hierarchy

Table 265. Source mappings for objectType hierarchy

Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
externalName	The value of the externalName property of the source Framework Manager model object.

- Object Type - function

Table 266. Source mappings for objectType function

Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
canonicalName	The value of the canonicalName property of the source Framework Manager model object.
syntaxTip	The value of the syntaxTip property in the caller's content locale.

- Object Type - dataSource

Table 267. Source mappings for objectType dataSource

Property Type	Source Mapping
cmDataSource	The value of the cmDataSource property of the source Framework Manager model object.

<i>Table 267. Source mappings for objectType dataSource (continued)</i>	
Property Type	Source Mapping
catalog	The value of the catalog property of the source Framework Manager model object.
schema	The value of the schema property of the source Framework Manager model object.
cube	The value of the cube property of the source Framework Manager model object.

- Object Type - filter

<i>Table 268. Source mappings for objectType filter</i>	
Property Type	Source Mapping
description	The value of the description property of the source Framework Manager model object.
screenTip	The value of the screenTip property of the source Framework Manager model object.
expression	The value of the expression property of the source Framework Manager model object.

- Object Type - filterDefinition

<i>Table 269. Source mappings for objectType filterDefinition</i>	
Property Type	Source Mapping
apply	The value of the apply property of the source Framework Manager model object.
expression	The value of the expression property of the source Framework Manager model object.
refobj	The value of the refobj property of the source Framework Manager model object.

Chapter 26. Lineage specification reference

For each layout data specification element, this section provides

- the name and description of the element
- information about attributes that apply to the element, including each attribute's name, description, optionality, legal values, and default value, if applicable
- content model information, consisting of a list of valid child elements presented as an element model group
- a list of valid parent elements

childRef

Specifies an object in the main object set that represents the contextual parent of the current object. Its value must match the `id` of another object in the main object set.

Content model

Content type is string.

Parent elements

object

connection

Specifies a source specification against which the objectQuery objects are processed.

Content model

param (*one or more*)

Parent elements

lineageRequest

lineageRequest

The root element of a lineage request.

Attributes

version

Specifies the version of the lineage API being employed.

Usage: required Type: string

Content model

service (*optional*) then (connection or V5QuerySet) then objectQuery (*any number*)

Parent elements

[transformation](#)

lineageResponse

The root element of a lineage response.

Content model

([queryResult](#) (*one or more*) then [object](#) (*one or more*)) (*optional*)

name

Specifies the name of an object. It does not have to be unique.

Content model

Empty element.

Parent elements

[object](#)

object

Specifies an object in the source specification against which the lineage request is being processed. It can either be an object involved in the requested lineage, with the [transformation](#) specified accordingly. It can also provide a context for other objects, or as a transformation source with no [transformation](#) of its own. In that case no [transformation](#) will exist, and only the [parentRef](#) and/or the [childRef](#) objects may exist.

Attributes

id

Specifies a unique identifier for the object. Its value can be referenced in the [childRef](#) of another object, the [parentRef](#) of another object or the [objectRef](#) of a [queryResult](#) or [transformation](#).

Usage: required Type: string

Content model

name then [type](#) then [property](#) (*any number*) then [transformation](#) (*optional*) then [parentRef](#) (*optional*) then [childRef](#) (*any number*)

Parent elements

[lineageResponse](#)

objectQuery

Specifies the request for the lineage of a single object in the specification pertaining to the [connection](#) or [V5QuerySet](#). The domain of values are service-specific.

Attributes

id

Specifies a value that uniquely identifies the `objectQuery` in the request. This value is specified as the `objectQueryRef` value of a `queryResult` in the response. This allows the caller to match the various `queryResult` elements in the response with corresponding `objectQuery` elements in the originating request.

Usage: optional Type: string

Content model

`param` (*one or more*)

Parent elements

`lineageRequest`

objectQueryRef

Specifies the binding of the `queryResult` to the `objectQuery` in the originating request. Its value must match that of the `id` of an `objectQuery` in the originating request.

Content model

Content type is string.

Parent elements

`queryResult`

objectRef

Specifies a reference from an object in the main object set of the response. Its value must match that of the `id` of another object in the main object set.

Content model

Content type is string.

Parent elements

`queryResult` , `transformation`

param

Specifies a generic name-value pair, which pertains to both `param` and `property` objects.

Attributes

name

Specifies the name of the element.

Usage: required Type: string

Content model

Content type is string.

Parent elements

[connection](#) , [objectQuery](#)

parentRef

Specifies a reference to an object in the main object set that represents the contextual parent of the current object. Its value must match the `id` of another object in the main object set.

Content model

Empty element.

Parent elements

[object](#)

property

Specifies a generic name-value pair, that contains source mappings for objects in the lineage response. Its `name` must be unique across sibling instances. The domain of name and value is service-specific.

Attributes

displayName

Specifies the displayed value for the name of a property.

Usage: optional Type: string

displayValue

Specifies the displayed value for the value of a property.

Usage: optional Type: string

name

Specifies the name of the element.

Usage: required Type: string

Content model

Content type is string.

Parent elements

[object](#) , [transformation](#)

queryResult

Specifies the result of an `objectQuery` in the originating request. It contains references to the objects in the main object set that represent the starting points for traversing the transformations.

Content model

objectRef then objectQueryRef (*optional*)

Parent elements

lineageResponse

service

Reserved.

Specifies a lineageRequest element that appears in the transformation for an object in the response, to inform the caller of the service that can handle this request.

Content model

Content type is string.

Parent elements

lineageRequest

transformation

Specifies the transformation data for a particular object in the source specification on which the parent object element is based. Its transformation sources are determined by its objectRef elements and/or a separate lineage request instance. The value of each objectRef element should match that the id of another object in the main object set. The lineageRequest element represents a subsequent lineage request that needs to be processed by some other service that can semantically process the transformation source.

Content model

type then property (*any number*) then (objectRef or lineageRequest)

Parent elements

object

type

Specifies a semantic type of the object.

Content model

Content type is string.

Parent elements

object

type

Specifies the semantic type of a transformation, whose domain of values are service-specific.

Content model

Empty element.

Parent elements

[transformation](#)

V5QuerySet

Specifies the set of data items returned from `reportService` or `batchReportService`, as a result of a lineage request on report data. The source specification is represented as escaped (encoded) XML and adheres to the report specification schema.

Content model

Content type is string.

Parent elements

[lineageRequest](#)

What's new

[Schema version 10.1.0](#)

This element was added.

Chapter 27. Understanding dynamic drill-through

Drill-through is the process of navigating from a source resource, such as a report or a PowerCube, to a target resource. To preserve context during navigation, the values selected in the source resource are usually mapped to parameters defined in the target resource.

Dynamic drill-through extends the current drill-through functionality by providing another mechanism to communicate the context for the drill-through operation.

Dynamic drill-through uses the values selected in source resource to dynamically create filters that filter the information that appears in the target resource. Dynamic drill-through allows applications to be more fluid and simpler to maintain because parameters do not need to be defined in the target resource.

This part contains reference information for three specifications used during a dynamic drill-through operation: the selection context, defined by the `SelectionContext.xsd` schema, the drill-through specification defined by the `DrillThroughSpecification.xsd` schema, and the edit specification, defined by the `EditSpecification.xsd` schema.

During a dynamic drill-through operation, the values selected in the source resource are represented in the selection context. The selection context is a dynamically created XML document that represents the values selected. The selection context also contains information about the values that are dependent on the values selected to provide additional opportunities to filter the values in the target resource. For selection context XML samples, see [Using Selection Context](#). For the `SelectionContext.xsd` schema reference, see [Selection Context Reference](#).

The drill-through specification identifies the target resource and the action to be performed on that resource, as well as the rules that define how the selection context should be converted to a form that can be consumed by the target resource. The drill-through specification is defined in the `specification` parameter of the `drillPath` object. For drill-through specification XML samples, see [Drill-through specification examples](#). For the `DrillThroughSpecification.xsd` schema reference, see [Drill-through specification reference](#).

The edit specification defines actions, such as filters, to perform on the target resource. For edit specification XML samples, see [Edit specification examples](#). For the `EditSpecification.xsd` schema reference, see [Edit specification reference](#).

Chapter 28. Drill-through specification examples

A drill-through specification contains the information necessary to perform a drill-through operation between two resources. This specification identifies the target resource and the action to be performed on that resource by the target provider agent. This specification also contains rules that define how the context from the source should be converted to a form that can be consumed by the target resource.

A drill-through specification is defined in the `specification` parameter of a `drillPath` object. The XML schema that defines a drill-through specification is `DrillThroughSpecification.xsd`.

For the `DrillThroughSpecification.xsd` schema reference, see [Drill-through specification reference](#).

Example - Drill-through to an IBM Cognos Analytics report

This sample XML contains a drill-through specification to navigate from a source resource in the **Sales** package to the IBM Cognos Analytics report named **Products in Inventory by Location** in the **Inventory** package. Mappings are defined for eight of the target report's parameters. The target report may define additional parameters. Some of the mappings are defined so that key values from multiple query subjects in the source model may be used to provide values for the target report parameters.

This drill-through specification specifies that the drill-through context must specify at least one country (items `[NS].[Countries].[ID]`, `[NS].[Orders].[CountryID]` or `[NS].[Countries].[Name]`) and at least one product type (items `[NS].[ProductType].[ID]`, `[NS].[Orders].[ProductTypeID]` or `[NS].[ProductType].[Label]`) before the drill-through specification can be used to perform a drill-through operation.

This drill-through specification specifies that the action to be performed on the target report is `view`.

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<dts:drillThroughSpecification xmlns:dts=
"http://developer.cognos.com/schemas/drillThroughSpecification/1/">
  <dts:modelSearchPaths>
    <dts:searchPath>
/content/package[@name='Sales']/model[@name='2007-05-30T18:49:25.035Z']
</dts:searchPath>
</dts:modelSearchPaths>
  <dts:parameterAssignments>
    <dts:parameterAssignment>
      <dts:parameterName>Countries
</dts:parameterName>
      <dts:alternateItems>
        <dts:item>[NS].[Countries].[ID]</dts:item>
        <dts:item>[NS].[Orders].[CountryID]</dts:item>
      </dts:alternateItems>
    </dts:parameterAssignment>
    <dts:parameterAssignment>
      <dts:parameterName>Country Names</dts:parameterName>
      <dts:item>[NS].[Countries].[Name]</dts:item>
    </dts:parameterAssignment>
    <dts:parameterAssignment>
      <dts:parameterName>States</dts:parameterName>
      <dts:alternateItems>
        <dts:item>[NS].[States].[ID]</dts:item>
        <dts:item>[NS].[Orders].[StateID]</dts:item>
      </dts:alternateItems>
    </dts:parameterAssignment>
    <dts:parameterAssignment>
      <dts:parameterName>State Names</dts:parameterName>
      <dts:item>[NS].[States].[Name]</dts:item>
    </dts:parameterAssignment>
    <dts:parameterAssignment>
      <dts:parameterName>Product Types</dts:parameterName>
      <dts:alternateItems>
```

```

    <dts:item>[NS].[ProductType].[ID]</dts:item>
    <dts:item>[NS].[Orders].[ProductTypeID]</dts:item>
  </dts:alternateItems>
</dts:parameterAssignment>
<dts:parameterAssignment>
  <dts:parameterName>Product Labels</dts:parameterName>
  <dts:item>[NS].[ProductType].[Label]</dts:item>
</dts:parameterAssignment>
<dts:parameterAssignment>
  <dts:parameterName>Years</dts:parameterName>
  <dts:item>[NS].[Orders].[Year]</dts:item>
</dts:parameterAssignment>
<dts:parameterAssignment>
  <dts:parameterName>Months</dts:parameterName>
  <dts:item>[NS].[Orders].[Month]</dts:item>
</dts:parameterAssignment>
</dts:parameterAssignments>
<dts:scope>
  <dts:alternateItems>
    <dts:item>[NS].[Countries].[ID]</dts:item>
    <dts:item>[NS].[Orders].[CountryID]</dts:item>
    <dts:item>[NS].[Countries].[Name]</dts:item>
  </dts:alternateItems>
  <dts:alternateItems>
    <dts:item>[NS].[ProductType].[ID]</dts:item>
    <dts:item>[NS].[Orders].[ProductTypeID]</dts:item>
    <dts:item>[NS].[ProductType].[Label]</dts:item>
  </dts:alternateItems>
</dts:scope>
<dts:reportTarget>
  <dts:action>
http://?developer.cognos.com/?schemas/?bibus/?3/?constants/
?drillThroughActionEnum#?view
  </dts:action>
  <dts:targetSearchPath>
    /content/package[@name='Inventory']/
    report[@name='Products in Inventory by Location']
  </dts:targetSearchPath>
</dts:reportTarget>
</dts:drillThroughSpecification>

```

Example - Drill-through to a package

This sample XML contains a drill-through specification to navigate from a source resource in the **Sales** package to the **Inventory** package. In this case, drilling through to a package causes a new resource to be created using the drill-through context in either IBM Cognos Analysis Studio or IBM Cognos Query Studio.

Mappings are not defined in this case since parameters are not defined in the target. No scoping restrictions are defined in this drill-through specification.

Sample XML

```

<?xml version="1.0" encoding="UTF-8"?>
<dts:drillThroughSpecification xmlns:dts=
"http://developer.cognos.com/schemas/drillThroughSpecification/1/">
  <dts:modelSearchPaths>
    <dts:searchPath>
/content/package[@name='Sales']/model[@name='2007-05-30T18:49:25.035Z']
    </dts:searchPath>
  </dts:modelSearchPaths>
  <dts:packageTarget>
    <dts:action>
http://?developer.cognos.com/?schemas/?bibus/?3/?constants/
?drillThroughActionEnum#?edit
    </dts:action>
    <dts:studios>
      <dts:uri>
http://?developer.cognos.com/?schemas/?bibus/?3/?constants/
?drillThroughRecipientEnum#?analysisStudio
      </dts:uri>
      <dts:uri>
http://?developer.cognos.com/?schemas/?bibus/?3/?constants/
?drillThroughRecipientEnum#?queryStudio
      </dts:uri>
    </dts:studios>
  </dts:packageTarget>
</dts:drillThroughSpecification>

```

```
</dts:studios>  
<dts:targetSearchPath>  
  /content/package[@name='Inventory']  
</dts:targetSearchPath>  
</dts:packageTarget>  
</dts:drillThroughSpecification>
```

Chapter 29. Drill-through specification reference

This section provides reference information for each drill-through specification element.

For each drill-through specification element, this section provides

- the name and description of the element
- sample code that demonstrates how to use the element, or a cross-reference to a topic that contains sample code
- information about attributes that apply to the element, including each attribute's name, description, optionality, legal values, and default value, if applicable
- content model information, consisting of a list of valid child elements presented as an element model group
- a list of valid parent elements

action

Specifies the action to be performed on the target when performing the drill-through. The `drillThroughActionEnum` enumeration set lists the defined actions.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

Content type is `anyURI`.

Parent elements

`iwrTarget` , `packageTarget` , `powerPlay7ReportTarget` , `powerPlay8ReportTarget` , `reportTarget` , `uriTarget`

alternateItems

specified as a child of the `bookmark` element

Identifies the items that may be used to provide values for the bookmark for the drill-through operation.

Use this element to specify a bookmark based on an item value.

If the context contains values for multiple items referenced by this list, the values corresponding to the item with the name that appears earliest in this list must be used to provide values for the bookmark. Other matching context values must not be used to provide values for the bookmark.

If this element is specified as the child of a `bookmark` element, the following elements must not be specified: `item` and `text` .

specified as a child of the `scope` element

Specifies the names of drill-through context items. At least one item must be present in the drill-through context before the drill-through specification can be used to perform a drill-through operation.

specified as a child of the parameterAssignment element

Identifies the items that may be used to provide values for the parameter for the drill-through operation.

If the context contains values for multiple items referenced by this list, the values corresponding to the item with the name that appears earliest in this list must be used to provide values for the target parameter. Other matching context values must not be used to provide values for the target parameter.

If this element is specified as a child of the parameterAssignment element, the item element must not be specified.

Attributes

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

(item or extension or any) (*one or more*)

Parent elements

bookmark , parameterAssignment , scope

bookmark

Specifies the information items used to reference a bookmark in the target resource.

Attributes

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

(item or alternateItems or text) then extension (*optional*) then any (*any number*)

Parent elements

reportTarget , uriTarget

characterEncoding

Specifies the character encoding to be used in the request to launch the target URI, target IBM Cognos Series 7 PowerPlay report or target IWR report .

The IANA (Internet Assigned Numbers Authority) maintains a registry of character sets that can be used in the Internet. Values in this element should match an alias from the registry. Whenever possible, the preferred MIME name should be used.

IBM Cognos Series 7 supports a limited number of the character sets registered with IANA.

If not specified, the utf-8 character encoding must be used.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is string.

Parent elements

[iwrTarget](#) , [powerPlay7ReportTarget](#) , [uriTarget](#)

drillThroughSpecification

The root element of a drill-through specification XML document.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

`modelSearchPaths` then `parameterAssignments` (*optional*) then `scope` (*optional*) then (`iwrTarget` or `packageTarget` or `powerPlay7ReportTarget` or `powerPlay8ReportTarget` or `reportTarget` or `uriTarget`) (*optional*) then `extension` (*optional*) then any (*any number*)

extension

Allows elements from future versions of the target namespace to be added to the element while maintaining compatibility with the current version of the target namespace schema.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

any (*any number*)

Parent elements

[alternateItems](#) , [bookmark](#) , [drillThroughSpecification](#) , [iwrTarget](#) , [modelSearchPaths](#) , [packageTarget](#) , [parameterAssignment](#) , [parameterAssignments](#) , [powerPlay7ReportTarget](#) , [powerPlay8ReportTarget](#) , [reportTarget](#) , [scope](#) , [studios](#) , [uriTarget](#)

item

Specifies the metadata model item in the drill-through context.

specified as a child of the parameterAssignment element

Identifies the item that may be used to provide values for the parameter for the drill-through operation.

If this element is specified, the alternateItems element must not be specified.

specified as a child of the alternateItems element

specified as a child of the scope element

Specifies the metadata model item that must be present in the drill-through context before the drill-through specification can be used to perform a drill-through operation.

specified as a child of the bookmark element

Identifies the item that may be used to provide values for the bookmark for the drill-through operation. Use this element to specify a bookmark based on an item value.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

Content type is string.

Parent elements

alternateItems , bookmark , parameterAssignment , scope

iwrTarget

Specifies the information items related to performing a drill-through operation to a target IWR report.

If this element is specified, the following elements must not be specified: packageTarget , powerPlay7ReportTarget , powerPlay8ReportTarget , reportTarget , and uriTarget .

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

action then characterEncoding then targetURI then extension (*optional*) then any (*any number*)

Parent elements

[drillThroughSpecification](#)

modelSearchPaths

Specifies the location(s) of the model(s) to be used with the drill-through specification. A model search path is an alternate search path that is required due to model versioning, object renaming, and deployment, all of which can invalidate a search path.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

([searchPath](#) or [extension](#) or any) (*one or more*)

Parent elements

[drillThroughSpecification](#)

packageTarget

Specifies the information items related to performing a drill-through operation to a target package.

If this element is specified, the following elements must not be specified: [iwrTarget](#) , [powerPlay7ReportTarget](#) , [powerPlay8ReportTarget](#) , [reportTarget](#) , and [uriTarget](#) .

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

[action](#) then [studios](#) then [targetSearchPath](#) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[drillThroughSpecification](#)

parameterAssignment

Assigns values from the selection context to parameters defined by the drill-through target during the drill-through operation. The metadata model items in the selection context are compared with the item names in the parameter assignments to determine which values are assigned to the target's parameters

For more information about the selection context, see *Using selection context*.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

`parameterName` then (`item` or `alternateItems`) then `extension` (*optional*) then any (*any number*)

Parent elements

`parameterAssignments`

parameterAssignments

Specifies mappings between source model items and parameters defined in the drill-through target.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

(`parameterAssignment` or `extension` or any) (*one or more*)

Parent elements

`drillThroughSpecification`

parameterName

Specifies the name of a parameter defined by the drill-through target.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is string.

Parent elements

`parameterAssignment`

powerPlay7ReportTarget

Specifies the information items related to performing a drill-through operation to a target IBM Cognos Series 7 PowerPlay report.

If this element is specified, the following elements must not be specified: [iwrTarget](#) , [packageTarget](#) , [powerPlay8ReportTarget](#) , [reportTarget](#) , and [uriTarget](#) [uriTarget](#) .

Attributes

Adding Other Attributes

[anyAttribute](#) indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and [processContents](#) parameters, respectively.

Content model

[action](#) then [characterEncoding](#) then [targetURI](#) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[drillThroughSpecification](#)

powerPlay8ReportTarget

Specifies the information items related to performing a drill-through operation to a target basePowerPlay8Report.

If this element is specified, the following elements must not be specified: [iwrTarget](#) , [packageTarget](#) , [powerPlay7ReportTarget](#) , [reportTarget](#) , and [uriTarget](#) [uriTarget](#) .

Attributes

Adding Other Attributes

[anyAttribute](#) indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and [processContents](#) parameters, respectively.

Content model

[action](#) then [targetSearchPath](#) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[drillThroughSpecification](#)

reportTarget

Specifies the information items related to performing a drill-through operation to a target baseReport.

If this element is specified, the following elements must not be specified: [iwrTarget](#) , [packageTarget](#) , [powerPlay7ReportTarget](#) , [reportTarget](#) , and [uriTarget](#) [uriTarget](#) .

Attributes

Adding Other Attributes

[anyAttribute](#) indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and [processContents](#) parameters, respectively.

Content model

action then targetSearchPath then bookmark (*optional*) then extension (*optional*) then any (*any number*)

Parent elements

drillThroughSpecification

scope

Identifies the model items that must be present in the drill-through context before the specification can be used to perform a drill-through from source resource to target resource.

Attributes

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

(item or alternateItems or extension or any) (*one or more*)

Parent elements

drillThroughSpecification

searchPath

Identifies an object in the content store.

For more information about search path syntax, see in the appendix section.

Attributes

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

Content type is string.

Parent elements

modelSearchPaths

studios

Specifies the list of studios that may perform the action on the target when performing a drill-through. The drillThroughRecipientEnum drillThroughRecipientEnum enumeration set lists the defined studios.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

(uri or extension or any) (*one or more*)

Parent elements

packageTarget

targetSearchPath

Specifies the location of the target package, the PowerPlay 8 Report or the target report.

If this element is a child of the powerPlay8ReportTarget element, or the reportTarget element, this element must be included in the target base-Power-Play8-Report or target base-Report specification.

For more information about search path syntax, see in the appendix section.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is string.

Parent elements

packageTarget , powerPlay8ReportTarget , reportTarget

targetURI

specified as a child of the iwrTarget element

Specifies the location of the target IWR report.

specified as a child of the powerPlay7ReportTarget element

Specifies the location of the target IBM Cognos Series 7 PowerPlay report.

specified as a child of the uriTarget element

Specifies the location of the target resource.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is `anyURI`.

Parent elements

[iwrTarget](#) , [powerPlay7ReportTarget](#) , [uriTarget](#)

text

Identifies the bookmark for the drill-through operation.

Use this element to specify a static bookmark.

If this element is specified then the following elements must not be specified: [item](#) and [alternateItems](#) .

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is `string`.

Parent elements

[bookmark](#)

uri

Specifies the URI.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

Content type is `anyURI`.

Parent elements

[studios](#)

uriTarget

Specifies the information items related to performing a drill-through operation to a target URI.

If this element is specified, the following elements must not be specified: iwTarget , packageTarget , powerPlay7ReportTarget , powerPlay8ReportTarget and reportTarget must not be specified.

Attributes

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

action then targetURI then bookmark (*optional*) then characterEncoding (*optional*) then extension (*optional*) then any (*any number*)

Parent elements

drillThroughSpecification

Chapter 30. Edit specification examples

An edit specification defines the actions, such as filters, to apply to a target resource during a drill-through operation.

The XML schema that defines an edit specification is `EditSpecification.xsd`. For the `EditSpecification.xsd` reference, see [Edit specification reference](#).

The following examples demonstrate how an edit specification is encoded. The examples are based on the relational selection context example in `Using Selection Context`.

Example - Edit specification for entire selection context

The following sample XML is an edit specification constructed from the relational selection context example. Note that the selected cell does not appear. This edit specification contains only the values from the defining cells and their dependent cells.

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<es:editSpecification
  xmlns:es="http://developer.cognos.com/schemas/editSpecification/1/">
  <es:actions>
    <es:filter>
      <es:expression>
        <es:operator>
          <es:name>
            http://developer.cognos.com/schemas/editSpecification/1/operators#and
          </es:name>
          <es:arguments>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Order Type</es:name>
                </es:dataItem>
                <es:value>
                  <es:simple>Internet</es:simple>
                </es:value>
              </es:value>
            </es:argument>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Shipment Type</es:name>
                </es:dataItem>
                <es:value>
                  <es:simple>Express</es:simple>
                </es:value>
              </es:value>
            </es:argument>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Payment Type</es:name>
                </es:dataItem>
                <es:value>
                  <es:simple>Visa</es:simple>
                </es:value>
              </es:value>
            </es:argument>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Country Name</es:name>
                </es:dataItem>
                <es:value>
                  <es:simple>Canada</es:simple>
                </es:value>
              </es:value>
            </es:argument>
          </es:arguments>
        </es:operator>
      </es:expression>
    </es:filter>
  </es:actions>
</es:editSpecification>
```

```

</es:argument>
<es:argument>
  <es:value>
    <es:dataItem>
      <es:name>State Name</es:name>
    </es:dataItem>
    <es:value>
      <es:simple>Ontario</es:simple>
    </es:value>
  </es:argument>
<es:argument>
  <es:value>
    <es:dataItem>
      <es:name>State Population</es:name>
    </es:dataItem>
    <es:value>
      <es:simple>12721776</es:simple>
    </es:value>
  </es:argument>
<es:argument>
  <es:value>
    <es:dataItem>
      <es:name>Product Type</es:name>
    </es:dataItem>
    <es:value>
      <es:simple>CD</es:simple>
    </es:value>
  </es:argument>
<es:argument>
  <es:value>
    <es:dataItem>
      <es:name>Year</es:name>
    </es:dataItem>
    <es:value>
      <es:simple>2006</es:simple>
    </es:value>
  </es:argument>
</es:arguments>
</es:operator>
</es:expression>
</es:filter>
</es:actions>
</es:editSpecification>

```

Example - Edit specification demonstrating various data value alternatives

The previous example showed how a selection context can be used to construct an edit specification. However, all of the values in that example were simple, non-null values. This sample XML changes some of the values to show how NULL values, NULL-or-blank values, and range values can be specified.

Note that the entire edit specification is not repeated in this sample XML.

Sample XML

```

<?xml version="1.0" encoding="UTF-8"?>
<es:editSpecification
  xmlns:es="http://developer.cognos.com/schemas/editSpecification/1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <es:actions>
    <es:filter>
      <es:expression>
        <es:operator>
          <es:name>
            http://developer.cognos.com/schemas/editSpecification/1/operators#and
          </es:name>
          <es:arguments>
            <es:argument>
              <es:value>

```

```

<es:dataItem>
  <es:name>Order Type</es:name>
</es:dataItem>
<es:value>
  <es:simple xsi:nil="true"/>
</es:value>
</es:argument>
<es:argument>
<es:value>
  <!-- This data item does not appear in the selection context -->
  <es:dataItem>
    <es:name>City Name</es:name>
  </es:dataItem>
  <es:value>
    <es:nullOrBlank/>
  </es:value>
</es:argument>
<es:argument>
<es:value>
  <es:dataItem>
    <es:name>Year</es:name>
  </es:dataItem>
  <es:value>
    <es:range>
      <es:maxValue>2006-12-31</es:maxValue>
      <es:maxValueInclusive>true</es:maxValueInclusive>
      <es:minValue>2006-01-01</es:minValue>
      <es:minValueInclusive>true</es:minValueInclusive>
    </es:range>
  </es:value>
</es:argument>
</es:arguments>
</es:operator>
</es:expression>
</es:filter>
</es:actions>
</es:editSpecification>

```

Sample XML - Edit specification demonstrating various data item operations

The edit specification allows filter expressions that define operations to be performed on the data item values before they are compared to the values provided in the edit specification. This example demonstrates how the edit specification declares that the trim operation is performed on the **Order Type** data item values.

Note that the entire edit specification is not repeated in this example.

```

<?xml version="1.0" encoding="UTF-8"?>
<es:editSpecification
  xmlns:es="http://developer.cognos.com/schemas/editSpecification/1/">
  <es:actions>
    <es:filter>
      <es:expression>
        <es:operator>
          <es:name>
            http://developer.cognos.com/schemas/editSpecification/1/operators#and
          </es:name>
          <es:arguments>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Order Type</es:name>
                  <es:operator>
                    http://developer.cognos.com/schemas/editSpecification/1/operators#trim
                  </es:operator>
                </es:dataItem>
                <es:value>
                  <es:simple>Internet</es:simple>
                </es:value>
              </es:value>
            </es:argument>
          </es:arguments>

```

```
</es:operator>
</es:expression>
</es:filter>
</es:actions>
</es:editSpecification>
```

Example - Edit specification demonstrating various data item operations with xml:base

The attribute `xml:base` [XMLBase], may be used to define a base URI for the document. The base URI may be defined in a way that allows relative URIs to be used instead of absolute URIs to identify operators used in the edit specification.

This example builds on the previous example by specifying an `xml:base` attribute value on the root element. Relative URIs identify the operators in the specification.

Sample XML

```
<?xml version="1.0" encoding="UTF-8"?>
<es:editSpecification
  xml:base="http://developer.cognos.com/schemas/editSpecification/1/"
  xmlns:es="http://developer.cognos.com/schemas/editSpecification/1/">
  <es:actions>
    <es:filter>
      <es:expression>
        <es:operator>
          <es:name>operators#and</es:name>
          <es:arguments>
            <es:argument>
              <es:value>
                <es:dataItem>
                  <es:name>Order Type</es:name>
                  <es:operator>operators#trim</es:operator>
                </es:dataItem>
              <es:value>
                <es:simple>Internet</es:simple>
              </es:value>
            </es:argument>
          </es:arguments>
        </es:operator>
      </es:expression>
    </es:filter>
  </es:actions>
</es:editSpecification>
```

Chapter 31. Edit specification reference

This section provides reference information for each edit specification element.

For each edit specification element, this section provides

- the name and description of the element
- sample code that demonstrates how to use the element, or a cross-reference to a topic that contains sample code
- information about attributes that apply to the element, including each attribute's name, description, optionality, legal values, and default value, if applicable
- content model information, consisting of a list of valid child elements presented as an element model group
- a list of valid parent elements

actions

Specifies the actions to apply to the target specification.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([filter](#) or [extension](#)) (*one or more*)

Parent elements

[editSpecification](#)

argument

Specifies an expression that can be evaluated to determine the value of an operator argument.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([operator](#) or [value](#) or [extension](#)) (*optional*)

Parent elements

[arguments](#)

arguments

Specifies the arguments for the operator.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([argument](#) or [extension](#)) (*one or more*)

Parent elements

[operator](#)

dataItem

Identifies the data item used in the comparison operation. The comparison operation compares the value of the data item with the construct in the [value](#) element.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

[name](#) then [operator](#) (*optional*) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[value](#)

editSpecification

Root element of an edit specification.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

[actions](#) (*optional*) then [extension](#) (*optional*) then any (*any number*)

end

Defines the maximum value of the range.

Content model

Content type is string.

Parent elements

range

expression

Specifies the expression for the filter.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

(operator or value or extension) (*optional*)

Parent elements

filter

extension

This element is an extensibility mechanism to allow elements from future versions of the target namespace to be added to the element while maintaining compatibility with the current version of the target namespace schema.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

any (*one or more*)

Parent elements

actions , argument , arguments , dataItem , editSpecification , expression , filter , operator , range , value , value

filter

Specifies a filter to add to the target resource specification.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

expression (*optional*) then extension (*optional*) then any (*any number*)

Parent elements

actions

name

Identifies the data item used in the comparison operation. The comparison operation compares the value of the data item with the construct in the value element.

Content model

Content type is anyURI.

Parent elements

dataItem , operator

null

Defines a value that matches a NULL value.

Content model

Empty element.

Parent elements

value

nullOrBlank

Defines a value that matches a NULL value or a value consisting of an arbitrary number of blanks (x20).

If this element is specified, the following elements must not be specified: range and simple .

If this element is not specified, either the range element or the simple element must be specified.

Content model

Empty element.

Parent elements

[value](#)

operator

Specifies an operator node in the expression tree.

If this element is specified, the [value](#) element must not be specified.

If this element is not specified, the [value](#) element must be specified.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

[name](#) then [arguments](#) (*optional*) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[argument](#) , [expression](#)

operator

Identifies the operation performed on data item values before being compared to the values in the edit specification.

Data item operators are specific to the data items and do not include operators like `and` and `or`.

Content model

Content type is `anyURI`.

Parent elements

[dataItem](#)

range

Defines a value range.

If this element is specified, the following elements must not be specified: [nullOrBlank](#) and [simple](#) .

If this element is not specified, either of the following elements must be specified: [nullOrBlank](#) or [simple](#) .

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

start then end then extension (*optional*) then any (*any number*)

Parent elements

value

simple

Defines a single value which may be NULL.

If this element is specified, the following elements must not be specified: nullOrBlank and range .

If this element is not specified, either of the following elements must be specified: nullOrBlank or range .

Content model

Content type is string.

Parent elements

value

start

Defines the minimum value of the range.

Content model

Content type is string.

Parent elements

range

value

Specifies a value node in the expression tree.

If this element is specified, the operator element must not be specified.

If this element is not specified, the operator element must be specified.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

dataItem then value then extension (*optional*) then any (*any number*)

Parent elements

argument , expression

value

Specifies the values used in the comparison operation.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

null or nullOrBlank or range or simple or extension (*optional*)

Parent elements

value

Chapter 32. Search path syntax

A search path uses expressions to specify a path through the content store hierarchy to find objects. The search path is similar to a path in an operating system such as DOS or UNIX®. The search path is constructed through parent-child relationships from the root object.

When you use an IBM Cognos Software Development Kit method that performs an operation on a set of objects in the content store, one of the parameters you must specify is a search path. IBM Cognos can then select the objects you want to perform the operation on.

For example, to select the sample report object named Order Product List Report, use the following path:

```
/content/package[@name='GO Sales and Retailers']/folder[@name='Documentation Reports']/report[@name='Order Product List Report']
```

You can make the search path specific to target one particular object, or more generic to widen the search. For example, to select all objects in the Documentation Reports folder, replace the specific information about the report in the previous example with the wildcard character (*), as follows:

```
/content/package[@name='GO Sales and Retailers']/folder[@name='Documentation Reports']/*
```

However, the wider the search, the more likely that multiple objects match the search criteria and the longer the search takes.

Find the Search Path for an Object

In IBM Cognos Connection, you can find the search path for any object visible in the user interface, then use that search path in a BI Bus API request.

Steps

1. In the **Actions** column, click **More** for the object.
2. In the **Perform an action** dialog box, click **Set properties**.
3. Click **View the search path**.

A message box appears that contains the search path of the object.

4. Copy the search path.
5. Click **Close**.
6. Paste the search path where it is required.

Search Path Examples

The following table contains examples of search paths and their descriptions.

Search path	Description
~	Selects the account object associated with the current user.
~~	Selects the session object associated with the current user.
/	Selects the root.

Table 270. Search path descriptions (continued)

Search path	Description
/*	<p>Selects the child objects of the root.</p> <p>The asterisk (*) is a wildcard character.</p>
//folder	<p>Selects all <code>folder</code> objects in the content store.</p> <p>When a path starts with two slashes (//), all objects in the content store that fulfill the specified criteria are selected. In this case, the selected objects must be instances of the class <code>folder</code>.</p>
//folder //report	<p>Selects all <code>folder</code> objects and all <code>report</code> objects in the content store. The vertical bar () specifies that the results of two search paths will be combined.</p>
//*	<p>Selects all objects in the content store.</p> <p>The asterisk (*) is a wildcard character.</p>
/configuration/**	<p>Selects the <code>configuration</code> object and all its descendants.</p> <p>The two slashes specify all descendant objects of the current object (<code>configuration</code>), and the current object itself. The text between the slashes is called a location step.</p> <p>The asterisk (*) is a wildcard character.</p> <p>For more information, see “Location Steps” on page 1567</p>
/content//folder/report/ parent::folder	<p>Selects every descendant of the content object that is a <code>folder</code> object and that has at least one child <code>report</code> object. In the <code>parent::folder</code> expression, <code>parent</code> is an axis and <code>folder</code> is a node test.</p> <p>For more information, see “Axes and Node Tests” on page 1568</p>
/content/package/folder [@name='Documentation Report Samples']/*	<p>Selects all child objects in the Documentation Report Samples folder. Expressions enclosed in square brackets are predicates used to filter a set of objects.</p> <p>The at sign (@) specifies a property name.</p> <p>The asterisk (*) is a wildcard character.</p> <p>For more information, see “Predicates” on page 1569</p>
/content/package/folder/report[contains(@name, 'Product List')]	<p>Selects all <code>report</code> objects in the path <code>/content/package/folder</code> with names that contain the string <code>Product List</code>. The predicate contains a call to the function <code>contains</code>.</p> <p>The at sign (@) specifies a property name.</p> <p>For more information, see “Functions” on page 1569</p>
storeID("1e08b01ef26b496aac06a14f5ae9a572")//report	<p>Selects all <code>report</code> objects that are descendants of the object that has the <code>storeID</code> with the value <code>"1e08b01ef26b496aac06a14f5ae9a572"</code>.</p> <p>For more information, see “storeID(storeID)” on page 1573</p>

Table 270. Search path descriptions (continued)

Search path	Description
CAMID(":")/*[@routingHints]	<p>Selects all objects in the Cognos namespace where the routingHints property is not nil. Expressions enclosed in square brackets are predicates used to filter a set of objects.</p> <p>CAMID(":") specifies the Cognos namespace.</p> <p>The at sign (@) specifies a property name.</p> <p>The asterisk (*) is a wildcard character.</p> <p>For more information, see “Predicates” on page 1569</p>
CAMID(":")/*[not(@routingHints)]	<p>Selects all objects in the Cognos namespace where the routingHints property is nil. Expressions enclosed in square brackets are predicates used to filter a set of objects.</p> <p>CAMID(":") specifies the Cognos namespace.</p> <p>The at sign (@) specifies a property name.</p> <p>The asterisk (*) is a wildcard character.</p> <p>For more information, see “Predicates” on page 1569</p>

Search Path Syntax

If you are familiar with XPath, you will notice that search path syntax resembles XPath. However, XPath operates on XML documents, and the search path operates on objects in the content store.

For example, in a search path:

- an object is represented by XPath XML element syntax
- an object property is represented by XPath XML attribute syntax
- the node test tests the class of an object instead of the name of an element

Search path syntax is limited to what is described in this section. Other XPath syntax and functions are not valid for search paths.

Note: Search paths are case-sensitive.

Location Steps

A search path contains a series of one or more location steps separated by a slash (/). Each location step is evaluated in sequence, from left to right, to select a set of objects used as the context to evaluate the next location step.

Each location step consists of the following:

- an axis, either implied or explicit
- a node test
- zero or more predicates

The unabbreviated syntax for a location step is the axis name and node test separated by two colons (::), possibly followed by one or more predicate expressions, each in square brackets ([]).

For more information about predicates, see [“Predicates” on page 1569](#)

For example, in the following search path, the location step `child::report[@name='Order Product List Report']` selects all child objects of folder objects that are report objects and that have a name property set to Order Product List Report:

```
//folder/child::report[@name='Order Product List Report']
```

Axes and Node Tests

The axis of a location step indicates where to look for the objects to select. By specifying the relationship to the current object, the axis defines the direction of the search through the object hierarchy of the content store. For example, the `ancestor` axis specifies that the search is to be conducted in the hierarchy above the current object.

If you do not specify an axis in a location step, the `child` axis is implied. For example, the following search path is abbreviated:

```
/content/package/folder
```

The unabbreviated syntax is:

```
/child::content/child::package/child::folder
```

A node test identifies a set of objects in an axis. The node test filters the objects, based on their class.

For example, in the location step `child::folder`, `child` is the axis and `folder` is the node test that filters out all objects but those of class `folder`.

You can use the asterisk (*) as a wildcard character in a node test to match all objects in the set. For example, the following search path selects all objects that are children of the package object `GO Sales and Retailers`:

```
//package[@name='GO Sales and Retailers']/*
```

The Parent Axis

The `parent` axis specifies the parent of the current object. For example, the following search path selects all `folder` objects that contain report objects:

```
//folder/report/parent::folder
```

The Ancestor Axis

The `ancestor` axis specifies the ancestors, such as the parent or grandparent, of the current object. For example, the following search path selects the package object that is the ancestor of the folder object named `Documentation Report Samples`:

```
/content//folder[@name='Documentation Report Samples']/ancestor::package
```

The Child Axis

The `child` axis specifies the immediate child objects of the current object. For example, the following search path selects `folder` objects that are immediately descended from a package object, but not any `folder` objects that they may contain:

```
/content/package/child::folder
```

The Descendant Axis

The `descendant` axis specifies the child objects of the current object, and recursively all the children's children. In descendant queries, traverse functionality is implied so that the search does not descend a branch that cannot be traversed. For example, the following search path selects all objects that are descendants of a `folder` object within a package object:

```
/content/package/folder/ancestor::*
```

This search path does not, however, include the `folder` objects that are immediately descended from a package object.

You cannot use this axis when the directory object is the current object.

The Descendant-or-self Axis

The descendant-or-self axis is the same as the descendant axis, except that it also includes the current object itself. In descendant-or-self queries, traverse functionality is implied so that the search does not descend a branch that cannot be traversed. For example, the following search path selects all objects that are descendants of a package object, including the folder objects immediately descended from the package object:

```
/content/package/folder/descendant-or-self:*
```

You can abbreviate this axis by using two slashes (//). For example, the previous example can be abbreviated as follows:

```
/content/package/folder//*
```

You cannot use this axis when the directory object is the current object.

Predicates

Each location step may include predicates. A predicate is an expression enclosed in square brackets ([]) that filters an object set.

For each object in the set to be filtered, the predicate expression is evaluated with that object as the context object, and with the number of objects in the object set as the context size. All predicates in a location step must be true for an object to be selected.

You can use only searchable properties in a predicate, unless you are testing the property for nil or non-nil values.

A predicate expression can include logical or comparison operators.

For example, the following search path selects all folders that have the hasPermission property set to either read or traverse:

```
//folder[permission('read') or permission('traverse')]
```

If the predicate consists of only a property name, objects for which that property is not nil will be selected. For example, the following search path selects all packages that have a value specified for the routingHint property:

```
/content/package[@routingHints]
```

The following search path selects all packages that have a nil routingHint property:

```
/content/package[not(@routingHints)]
```

Note: You cannot filter nil or non-nil values for a read-only property or a property acquired from a containing object.

Functions

You can use various object-set and boolean functions in your search path to make your search more specific.

Object-set Functions

Use an object-set function as your search path when you want to find specific objects in the content store.

CAMID

object-set CAMID(*CognosAccessManagerID*)

The CAMID function selects a single security object.

For example, the following search path selects the Server Administrators group:

```
CAMID(":Server Administrators")
```

dataSourceSignon(dataSourceName)

object-set `dataSourceSignon(dataSourceName)`

The `dataSourceSignon` function selects an appropriate set of `dataSourceConnection` and `dataSourceSignon` objects for the current user.

dataSourceName specifies the name property of a `dataSource` object.

The function selects `dataSourceConnection` objects where

- the current user has execute permission
- a parent `dataSource` object has a name equal to the value of *dataSourceName*

The function also selects `dataSourceSignon` objects where

- the current user has execute permission
- there is a reference to the `account` object for the current user or to one of the `group`, `namespace`, or selected `role` objects associated with the current user
- there is a parent that is a selected `dataSourceConnection` object

For example, the following search path selects the `dataSource` object `gosales` and returns the `dataSourceConnection` and `dataSourceSignon` objects:

```
dataSourceSignon("gosales")
```

personalDataSourceSignon(dataSourceName)

object-set `personalDataSourceSignon(dataSourceName)`

The `personalDataSourceSignon` function selects an appropriate set of `dataSourceConnection` and `dataSourceSignon` objects for the current user.

dataSourceName specifies the name property of a `dataSource` object.

The function selects `dataSourceConnection` objects where

- the current user has execute permission
- a parent `dataSource` object has a name equal to the value of *dataSourceName*

The function selects `dataSourceSignon` objects where

- the current user has execute permission
- there is a reference to the `account` object for the current user or to one of the `group`, `namespace`, or selected `role` objects associated with the current user
- there is a parent that is a selected `dataSourceConnection` object

If the user has the `canUsePersonalDataSourceCredentials` capability then this function returns `dataSourceCredential` objects where

- there is a parent that matches the primary account (the first namespace the user logged into) associated with the current user session
- the `bibus » dataSourceCredential » dataSourceName` property matches the `dataSourceName` argument passed to the function
- the `bibus » dataSourceCredential » dataSourceConnectionName` property matches a selected `dataSourceConnection` object

For example, the following search path selects the `dataSource` object `gosales` and returns the `dataSourceConnection` and `dataSourceSignon` objects, dependent on the conditions described in this section:

```
dataSourceSignon("gosales")
```

What's new

New in Version 10.1.0 – “Personal Data Source Credentials” on page 1851

This function was added.

defaultOutput(searchPath)

object-set *defaultOutput(searchPath)*

The *defaultOutput* function returns at least one output object.

searchPath is a search path that selects a report, reportView, or query object.

The function selects an output object that is a descendant of the object selected by *searchPath*. Content Manager selects only the following:

- output objects that are children of the most recent *reportVersion* object
- output objects that the current user can read

If any of these objects has a *locale* property value that matches the *contentLocale* user preference of the current session, Content Manager selects only these objects.

If any of these objects has a *defaultOutputFormat* property value that matches the *format* user preference of the current session, Content Manager selects only these objects.

Only one object is returned. If more than one object is selected, the function arbitrarily chooses a single object to return.

For example, the following search path selects the report object named Customer Invoice and returns the appropriate output object:

```
defaultOutput(/content/package/folder[@name='Sample Reports']/  
report[@name='Customer Invoice'])
```

defaultOutput(searchPath, formatHint, localeHint)

object-set *defaultOutput(searchPath, formatHint, localeHint)*

This implementation of the *defaultOutput* function is similar to the *defaultOutput(searchPath)* function, except that it accepts additional parameters. Use *formatHint* to specify your preferred format and *localeHint* to specify your preferred locale. These settings will override the user preferences when the output objects are selected. If you specify an empty hint (""), the user preferences are used.

For example, you can use any of the following:

- *defaultOutput(searchPath, formatHint, localeHint)*
- *defaultOutput(searchPath, "", localeHint)*
- *defaultOutput(searchPath, formatHint, "")*
- *defaultOutput(searchPath, "", "")*

If you specify an empty hint for both *formatHint* and *localeHint*, the function works exactly the same way as the *defaultOutput(searchPath)* function.

For example, using the format PDF and the locale en-us, the following search path selects the report object named Customer Invoice and returns the appropriate output object:

```
defaultOutput(/content/package/folder[@name='Sample Reports']/  
report[@name='Customer Invoice'], "PDF", "en-us")
```

expandDistributionMembers(object_set)

object-set *expandDistributionMembers(object_set)*

The *expandDistributionMembers* function recursively expands *group*, *role*, and *distributionList* membership.

object_set is a set of objects. Each object has a `members` property.

The `expandDistributionMembers` function must select distinct, single objects. These objects must be of the `account`, `group`, `role`, `distributionList`, or `contact` class.

For example, the following search path selects all the account objects that are members of the All Authenticated Users group:

```
expandDistributionMembers(/directory/*[@name="Cognos"]/*[@name="All Authenticated Users"])
```

Multiple objects can be specified using an or operator (`|`), as in the following example:

```
expandDistributionMembers(/directory/  
descendant::*[@id="LDAP:cn=Managers,ou=Ottawa"]|/directory/  
descendant::*[@id="LDAP:cn=MoreManagers,ou=Ottawa"])
```

What's new

New in Version 8.3 — [“New Properties for Group and Role Classes” on page 1919](#)

This function was added.

expandMembers(object_set)

`object-set expandMembers(object_set)`

The `expandMembers` function recursively expands `group`, `role`, and `distributionList` membership.

object_set is a set of objects. Each object has a `members` property.

The `expandMembers` function must select distinct, single objects. These objects must be of the `account`, `group`, `role`, `distributionList`, or `contact` class.

For example, the following search path selects all the account objects that are members of the All Authenticated Users group:

```
expandMembers(/directory/*[@name="Cognos"]/*[@name="All Authenticated Users"])
```

Multiple objects can be specified using an or operator (`|`), as in the following example:

```
expandMembers(/directory/descendant::*[@id="LDAP:cn=Managers,ou=Ottawa"]|/  
directory/descendant::*[@id="LDAP:cn=MoreManagers,ou=Ottawa"])
```

What's new

New in Version 8.3 — [“New Properties for Group and Role Classes” on page 1919](#)

This function is deprecated. Use the [“expandDistributionMembers\(object_set\)” on page 1571](#) and [“expandSecurityMembers\(object_set\)” on page 1572](#) functions instead.

expandSecurityMembers(object_set)

`object-set expandSecurityMembers(object_set)`

The `expandSecurityMembers` function recursively expands `group` and `role` membership.

object_set is a set of objects. Each object has a `members` property.

The `expandSecurityMembers` function must select distinct, single objects. These objects must be of the `account`, `group`, `role`, or `contact` class.

For example, the following search path selects all the account objects that are members of the All Authenticated Users group:

```
expandSecurityMembers(/directory/*[@name="Cognos"]/*[@name="All Authenticated Users"])
```

Multiple objects can be specified using an or operator (`|`), as in the following example:

```
expandSecurityMembers(/directory/  
descendant:*[@id="LDAP:cn=Managers,ou=Ottawa"]|/directory/  
descendant:*[@id="LDAP:cn=MoreManagers,ou=Ottawa"])
```

What's new

New in Version 8.3 — “New Properties for Group and Role Classes” on page 1919

This function was added.

membership(object_set)

object-set *membership(object_set)*

The *membership* function returns all security objects in the Cognos namespace that have a *members* property referencing one of the objects specified by *object_set*. This function is used to determine what groups and roles apply to a particular user.

object_set is a set of security objects specified by search paths separated by an or operator (|).

The *membership* function generates a set of Cognos security objects by performing one of the following tasks for each object referenced in the request:

- Find every Cognos *group* or *role* object whose *members* property references the object.
- For each Cognos *group* or *role* object found, recursively find any *group* or *role* object whose *members* property references the *group* or *role* object.

For example, the following search path selects all *group* and *role* objects in which the *members* property references the Directory Administrators group, and recursively selects all *group* and *role* objects that reference the first set of *group* and *role* objects selected:

```
membership(CAMID(":Directory Administrators"))
```

ownedBy(owner)

object-set *ownedBy(owner)*

The *ownedBy* function returns the objects in which the value of the *owner* property matches the *owner* specified by *owner*. *owner* can be either a call to the CAMID function, or a tilde (~), which specifies the account object associated with the current user.

For example, use the following search path to select all *schedule* objects owned by the current user:

```
//schedule[ownedBy(~)]
```

storeID(storeID)

object-set *storeID("storeID")*

The *storeID* function selects the object with the specified *storeID*. This function is restricted to the first location step in the search path.

For example, the following search path selects the object that has a *storeID* property with a value of "880DE2F0CDD811D1A849006097ABDE17":

```
storeID("880DE2F0CDD811D1A849006097ABDE17")
```

Boolean Functions

Use boolean functions in the predicate of a search path to

- check for the occurrence of a specific string in a property value
- determine whether a property exists for a specified object class
- determine whether a specific permission was granted in the current security context

contains(propertyname,string)

boolean *contains(propertyname, string)*

The *contains* function returns `true` if the value of the property named *propertyname* contains the string *string*. *propertyname* must begin with an at sign (@). If the property is a multilingual property, such as *name*, the value for each locale is checked for the specified string.

For example, the following search path selects `report` objects with `Product List` in the name:

```
//report[contains(@name, 'Product List')]
```

ends-with(propertyname,constraint)

boolean *ends-with(propertyname, constraint)*

The *ends-with* function returns `true` if the value of the property named *propertyname* ends with the string *constraint*. Otherwise, it returns `false`. *propertyname* must begin with an at sign (@). If the property is a multilingual property, such as *name*, the value for each locale is checked for the specified string.

For example, the following search path selects `report` objects with names that end with `Report`:

```
//report[ends-with(@name, 'Report')]
```

hasProperty(propertyname)

boolean *hasProperty(propertyname)*

The *hasProperty* function returns `true` if the property named *propertyname* exists, and `false` if it is not defined for an object of the specified class or if the property value is acquired from an ancestor object. This function returns a value even if the specified property cannot be accessed in the current security context.

For example, use the following search path to select all the objects that have a disabled property:

```
//*[hasProperty("disabled")]
```

last()

boolean *last()*

The *last* function selects a single object based on the value of the `retention` property of the parent object. The objects are ordered as indicated in the `retention` property value of the parent object.

The class of the object must be explicitly specified in the search and it must be subject to retention rules.

For information about retention, see [“Retentions Property” on page 57](#) .

For example, the following search path selects the last version of the report named `Order Product List Report`:

```
/content/package/folder/report[@name='Order Product List Report']/reportVersion[last()]
```

permission(permissionvalue)

boolean *permission(permissionvalue)*

The *permission* function returns `true` if the permission named *permissionvalue* is granted in the current security context.

When you use this function, it must be the only expression in the predicate.

For example, the following search path selects all `folder` objects for which you have `read` or `traverse` permission:

```
//folder[permission('read') or permission('traverse')]
```


starts-with(propertyname,string)

boolean *starts-with(propertyname, string)*

The *starts-with* function returns `true` if the value of the property named *propertyname* starts with the string *string*. *propertyname* must begin with an at sign (@). If the property is a multilingual property, such as `name`, the value for each locale is checked for the specified string.

For example, the following search path selects all `report` objects that have names beginning with `List`:

```
//report[starts-with(@name, 'List')]
```

Chapter 33. Advanced settings configuration

This section describes settings that you can use to perform advanced configuration with IBM Cognos Software Development Kit.

You must have the following access permissions to the configuration and service objects affected to change advanced settings:

- read and write permissions to the object that you want to update
- traverse permission for the parent of the object that you want to update

For information about access permissions, see [Chapter 4, “Managing security,” on page 45](#).

Though the property exists, there are no advanced settings available for the `dataIntegrationService`, the `logService`, the `metadataService` class, or the `systemService`. The `advancedSettings` properties of the `configurationFolder` and `dispatcher` object, while present, should not be set.

You specify global `advancedSettings` by using the `advancedSettings` of the configuration object. Values specified in this object are acquired by all contained objects unless the property of a contained object is set to override the global settings. You may need to override `advancedSettings` to provide customized values for specific service instances, but this can unnecessarily increase the overhead of administration.

Advanced settings reference

This section describes advanced settings for IBM Cognos services.

Agent service advanced settings

This section describes advanced settings for the agent service.

asv.preview.maxRows

Specifies the maximum number of rows to display in a **Preview All** request from IBM Cognos Event Studio.

Data type:

Integer

Default:

500

Note:

You must restart the service for this setting to take effect.

housekeeping.run.startup

Specifies whether state objects from previously run tasks are removed from the content store during startup. If false, the cleanup is only performed at the interval specified by `housekeeping.run.interval`.

Data type:

Boolean

Default:

false

Note:

You must restart the service for this setting to take effect.

housekeeping.run.interval

Specifies the interval, in hours, when housekeeping operations will take place for previously run agents. This value is used only if housekeeping.run.startup is set to false.

Data type:

Integer

Default:

12

Note:

You must restart the service for this setting to take effect.

primary.wait.asv

Specifies the time, in seconds, for the primary wait threshold for the agent service. This setting is used if a value is not set in the request.

Data type:

Integer

Default:

120

secondary.threshold

Specifies the time, in seconds, for the secondary wait threshold for asynchronous requests. The agent service only uses this service in running its tasks (rss, report, sql, and webservice tasks).

Data type:

Integer

Default:

30

Content Manager service advanced settings

This section describes advanced settings for the Content Manager service.

CM.CMSync_CheckActiveTime

Specifies the period within which an active Content Manager enters standby mode if another Content Manager becomes active.

Data type:

Integer

Default:

10000

CM.CMSync_NegotiationTime

Specifies failover election time in milliseconds.

The election time is the wait period after a Content Manager instance fails, before other Content Manager instances attempt to become the active service. This period ensures that another Content Manager service instance does not become active unless the original Content Manager is truly failing.

Data type:

Integer

Default:

2000

CM.CMSync_NegotiationTimeForStartUp

Specifies startup election time in milliseconds, after a computer shutdown.

This election time is the wait period during which the default Content Manager is expected to start before other standby Content Manager instances try to start. This ensures that the preferred Content Manager is started after a computer shutdown.

Data type:

Integer

Default:

60000

CM.CMSync_PingTimeout

Specifies maximum time, in milliseconds, within which a busy Content Manager should send a response.

After the timeout period, the election process begins to select a new Content Manager from the standby Content Manager instances, if any instances exist.

Data type:

Integer

Default:

120000

CM.CMSync_ShortNetworkInterruptionTime

Specifies a short network interruption time, in milliseconds, within which failover will not occur.

Data type:

Integer

Default:

3000

CM.DbConnectPoolMax

Specifies the maximum number of concurrent database connections allowed to the content store.

Valid settings are -1, or 5 to 2147483647, or the database setting; whichever value is less.

A setting of -1 means connections are unlimited.

This setting applies to Content Manager connection pool settings only. If you have other services that access the same content store, there may be more concurrent database connections than specified in this parameter.

Data type:

Integer

Default:

-1

CM.DbConnectPoolTimeout

Specifies the maximum time, in milliseconds, that a thread waits for a connection to be available from the pool.

A setting of 0 specifies that threads never wait for a connection if one is not available immediately. A setting of -1 means the wait time is unlimited.

Data type:

Integer

Default:

-1

CM.DbConnectPoolIdleTime

Specifies the minimum time, in milliseconds, that a connection stays idle in the pool.

This setting is valid only if the value of DbConnectPoolCleanupPeriod setting is positive.

A setting of 0 or -1 specifies that idle connections are closed when Content Manager restarts.

Data type:

Integer

Default:

300000

CM.DbConnectPoolCleanupPeriod

Specifies the time, in milliseconds, between invocations of a cleanup thread that closes idle connections in the pool that exceed the setting of DbConnectPoolIdleTime.

A setting of 0 or -1 specifies no cleanup thread.

Data type:

Integer

Default:

300000

CM.DeploymentIncludeConfiguration

Specifies if configuration objects should be imported from the entire content store archive during deployment.

These objects include dispatchers and the configuration folders used to group dispatchers. For example, you may want to import the configuration because you have a series of advanced settings for your services that you want to bring in from the source environment.

For best results, do not import configuration objects. Configure dispatchers in your target environment before you import data from a source environment.

Data type:

Boolean

Default:

false

CM.DeploymentSkipAllReportOutput

If this setting is set to `true`, report outputs and their child objects (graphic and page) in both **My content** and **Team content** are neither exported nor imported. Use this setting to reduce the size of the content store archives and improve deployment performance.

Data type:

Boolean

Default:

false

CM.DeploymentSkipUserReportOutput

If this setting is set to `true`, report outputs and their child objects (graphic and page) under user accounts are not exported or imported. Use this setting to reduce the size of the content store archives and improve deployment performance.

Data type:

Boolean

Default:
false

CM.DeploymentDetailErrorsOnly

If set to `true`, this setting generates only summary and error information for package and folder deployments. By default, Content Manager generates full details for package and folder deployment histories. Use this setting to reduce the size of the content store archives and to improve deployment performance.

Data type:
Boolean

Default:
false

CM.DeploymentDetailEntireContent

If set to `true`, this setting generates full details for an entire content store deployment history. By default, Content Manager generates only summary and error information for an entire content store deployment.

Data type:
Boolean

Default:
false

CM.DeploymentUpdateScheduleCredential

If set to `true` and the `takeOwnership` option is used during the import of a deployment archive, the credential property of all imported schedule objects is changed to reference the credential contained in the account used to import the deployment.

Data type:
Boolean

Default:
false

CM.OutPutLocation

Specifies the file system location where generated report outputs will be saved.

Each output file also has an output descriptor of the same name, with an XML extension.

Old report versions are not deleted when a new one is saved. You must manage the content of the output directory to keep only the report versions that you want.

Report outputs will always be written to the directory configured for each Delivery Service instance. In order to avoid having report outputs written to multiple locations, ensure that you are either running only one instance of the Delivery Service, or configure all service instances to use a shared network file location. Any Dispatcher running the Delivery Service must have access to the file system or be disabled on all systems not intended to save report output.

Data type:
String

Default:
none

CM.OutputScript

Specifies the location and name of an external script that runs each time a report output is saved.

The script parameters are the report output and output descriptor file names.

Data type:

String

Default:

none

CM.OutputByBurstKey

Specifies whether or not the outputs should be organized on the file system by burst key.

If set to `true`, the output is placed in a subdirectory of the same name as the burst key.

Data type:

Boolean

Default:

false

CM.SecurityQueryRequiresRead

Controls whether Content Manager forces a read permission filter for external namespace query results.

When enabled, Content Manager can prevent browsing of external namespaces, if the external namespace policy is also updated to deny read permission to users or groups.

Data type:

Boolean

Default:

false

CM.SortCollation

The name of the database-specific collation used for sorting in some databases, such as Oracle and SQL Server.

If left empty, the database uses its default collation.

For example, in Oracle, if you specify the collation sequence as `Binary` at the database level, you must provide the same collation sequence value in the connection string.

An example connection string for an Oracle database that uses the sample `gosl` database is:
`ORACLE@GOSL0703@GOSL/GOSL0703@COLSEQ=Binary`

For information about supported collations, see the Oracle and SQL Server documentation.

The `CM.SortCollation` value has no effect on Content Managers running against IBM Db2 or Sybase databases.

Data type:

String

Default:

none

CM.UpdateInitialContentNamesAfterImport

Adds localized object names for previously unsupported locales.

In some locales, if you want to upgrade to IBM Cognos Analytics from IBM Cognos Business Intelligence version 10.1.1 or earlier, and you plan to import a content store that was created with an older version of Cognos BI, use this advanced setting to ensure that all object names are properly localized.

The following locales are affected: Catalan, Croatian, Danish, Greek, Kazakh, Norwegian, Slovak, Slovenian, and Thai. Support for these locales was added in IBM Cognos Business Intelligence versions 10.1.1 and 10.2. If your content store was created with an earlier version, and the

CM.UpdateInitialContentNamesAfterImport setting was not specified before importing the content store, some object names might appear in English, and not in the specified language.

Specify the affected locales, separating each with a comma. For example, for Slovenian and Croatian content locales, type: `s1, h1`

Note: Remove this advanced setting when support for the older content store is no longer needed because there is a performance impact associated with this setting.

Data type:

String

Default:

none

Common configuration settings

This section describes advanced settings common to all services.

trustedSession.pool.max

Specifies the maximum number of trusted sessions that can be used concurrently. Trusted sessions use an internal security mechanism to encrypt the communications of internal components.

The sessions are implemented as a resource pool.

Data type:

Integer

Default:

100

Note:

You must restart the service for this setting to take effect.

axis.timeout

Specifies the timeout value, in seconds, for the internal axis server. This is the time that Axis will wait for a response to service calls before timing out.

Axis is an open-source tool for converting XML objects to Java objects.

Data type:

Integer

Default:

0

COGADMIN.filterInteractiveActivitiesOfUnknownUsers

Specifies whether activities in IBM Cognos Administration are hidden when the user doesn't have permission to view the user performing the activity.

Data type:

Boolean

Default:

false

COGADMIN.restrictInteractiveActivitiesToSystemAdministrators

Specifies whether interactive activities in IBM Cognos Administration are restricted to system administrators.

If this setting is set to `true`, the Current Activities tool will provide non-system administrators access to background activities only.

Data type:

Boolean

Default:

false

DISP.InteractiveProcessUseLimit

Forces the dispatcher to stop sending requests to a report server process after the prescribed limit.

For example, setting the limit to 500 forces the dispatcher to stop sending requests to a process after 500 requests.

Data type:

Integer

Default:

0

DISP.BatchProcessUseLimit

Forces the dispatcher to stop sending requests to a batch report server process after the prescribed limit.

Data type:

Integer

Default:

0

VIEWER_CW_BACKWARDS_COMPATIBLE_DRILL

Specifies whether legacy drill functionality is used in IBM Cognos Workspace.

By default, this setting is not specified and current drill up and down functionality is used in Cognos Workspace.

When this setting is set to `true`, drill functionality in Cognos Workspace reverts to its behavior in version 10.2.0 and earlier.

When this setting is set to `false`, current Cognos Workspace drill functionality is used.

Specify this setting at the configuration level on your system. Do not set for individual services.

Data type:

Boolean

Default:

false

Presentation service advanced settings

This section describes advanced settings for the presentation service.

CPSMaxCacheSizePerPortlet

Specifies the number of markup fragments cached for each portlet, per page, per user.

For example, a value of 5 with 1000 users, 10 pages, and 4 portlets per page can generate a maximum of 200000 entries in the cache (1000 x 10 x 4 x 5).

The following settings are valid:

- -1 saves an unlimited number of markups.
- 0 disables markup caching.
- 1 or an integer greater than 1 limits the number of markups to the specified number.

Data type:

Integer

Default:

-1

properties.config.cps.cache.timeToIdleSeconds

Specifies the length of time, in seconds, to keep the page markup fragments in the cache during a period of inactivity.

If the page is not accessed during that time, its cache contents are deleted.

The cache data saved on disk can be encrypted if the value of **Encrypt temporary files** is set to **True** under the **Environment** folder in IBM Cognos Configuration.

Data type:

Integer

Default:

1800 (30 minutes)

properties.config.cps.cache.timeToLiveSeconds

Specifies the length of time, in seconds, that page markup fragments are saved in the cache.

After the specified time, the markup is deleted, even if the cache is still active.

The cache data saved on disk can be encrypted if the value of **Encrypt temporary files** is set to **True** under the **Environment** folder in IBM Cognos Configuration.

Data type:

Integer

Default:

86400 (24 hours)

properties.config.cps.cache.checkExpiryIntervalSeconds

Specifies the length of time, in seconds, that represents the frequency with which the system checks for expired markup fragments in the cache.

The cache data saved on disk can be encrypted if the value of **Encrypt temporary files?** is set to **True** under the **Environment** folder in IBM Cognos Configuration.

Data type:

Integer

Default:

300 (5 minutes)

xts.tempdir

Specifies the location of the folder on the local drive where the markup fragments are stored.

The value can be any path on the local drive. If no value is specified, the default application server work area is used.

Data type:

String

Default:

blank

CSPPropagatePassport

Specifies whether IBM Cognos passport ID is transferred as a URL parameter.

When set to 0, this flag prevents the transfer of the IBM Cognos passport ID as a URL parameter.

Any value other than 0 allows the transfer of the passport ID.

Data type:

Default:

None

CSPPropagateTicket

Specifies whether IBM Cognos Configuration ticket ID is transferred as a URL parameter.

When set to 0, this flag prevents the transfer of the IBM Cognos Configuration ticket ID as a URL parameter.

Any value other than 0 allows the transfer of the ticket ID.

Data type:

Default:

None.

CSPProtocolScheme

Overrides the protocol scheme used when generating the Web Service Definition Language (WSDL) endpoint for Portal Services for Web Services Remote Portlets (WSRP) Producers.

To generate WSDL for WSRP, Portal Services uses the protocol scheme specified in the IBM Cognos Configuration gateway parameter. When there are multiple gateways that cannot all be configured using the same protocol scheme, for example http or https, this parameter overrides all other settings.

Valid settings are http and https

Data type:

String

Default:

None

portal.showTenantInfoForAllUsers

When set to true, users that do not have administrator permissions, can see tenant information.

For example, on the Set properties page, the tenant of an object is displayed. In object lists, users can see the tenant field.

Users are not able to change tenancy or to impersonate tenants.

Data type:

Boolean

Default:

False

Delivery service advanced settings

This section describes advanced settings for the delivery service.

emf.archive.filetimestamp.enabled

Forces timestamp on archived files.

Data type:
Boolean

Default:
true

max.smtp.connections

Specifies the maximum number of SMTP connections.

This setting limits the number of threads that the delivery service can spawn to send messages.

Valid settings are integers greater than or equal to 1.

Data type:
Integer

Default:
10

Tip: You must restart the service for this setting to take effect.

primary.wait.dls

Specifies the primary wait threshold, in seconds, for the delivery service.

This setting is used if a value is not set in a request.

If the setting is less than 0, it is ignored. If the setting is 0, the client will wait indefinitely.

Data type:
Integer

Default:
120

smtp.reconnection.delay

Specifies the time interval, in seconds, before an attempt to reconnect with an SMTP server is made.

Data type:

Default:
10

Tip: You must restart the service for this setting to take effect.

enable.tide.metrics.smtpqueue

Enables the collection and display of the metrics for the delivery service in the IBM Cognos Administration Console.

The following metrics are tracked:

- Time in queue high water mark
- Time in queue low water mark
- Time in queue
- Number of queue requests

- Queue length high water mark
- Queue length low water mark

Data type:

Boolean

Default:

false

dls.connection.pool.force.clean

Forces the cleanup of SMTP transport connections. This avoids the need to call the close() method, causing sockets to wait. Instead, variables are just set to null.

Data type:

Boolean

Default:

false

Set to true to force a cleanup.

Tip: After you apply changes, set to true to test the setting.

dls.connection.pool.used

Specifies whether the DLS Transport connection pool is used.

Data type:

Boolean

Default:

true

Set to true to use the connection pool.

Tip: Set to false so that the connection pool is not used. The result is that each email causes DLS to open a new SMTP transport connection with the email server. This can be helpful if mail server sockets are dropped after each use.

Dispatcher service advanced settings

This section describes advanced settings for the dispatcher.

DISP.InteractiveProcessUseLimit

Forces the dispatcher to stop sending requests to a report server process after the prescribed limit.

For example, setting the limit to 500 forces the dispatcher to stop sending requests to a process after 500 requests.

Data type:

Integer

Default:

0

DISP.BatchProcessUseLimit

Forces the dispatcher to stop sending requests to a batch report server process after the prescribed limit.

Data type:

Default:

0

Event management service advanced settings

This section describes advanced settings for the event management service.

run.task.max.thread

Specifies the maximum number of threads that are allocated to transfer scheduled requests to a holding queue.

When the event management service runs a task, the task is placed in a queue, awaiting resources to run it. A thread is created to handle the request for the scheduler thread of the event management service.

Default value: 20

Data type:

Integer

Default:

20

Note:

You must restart the service for this setting to take effect.

authenticate_when_scheduled

Determines whether a runAt request header is checked for execute permission for the object that will be executed.

If a check is required and it fails, an exception is thrown.

If set, this check also fails if the user has the permissions but the credentials necessary to run the task at a scheduled time can not be retrieved.

Data type:

Boolean

Default:

false

enable.tide.metrics.jobqueue

Enables the collection and display of specific metrics for the event management service in IBM Cognos Administration.

The following metrics are included:

- Time in queue high water mark
- Time in queue low water mark
- Time in queue
- Number of queue requests
- Queue length high water mark
- Queue length low water mark

Data type:

Boolean

Default:

false

ems.action.requires.permissions.check

Forces the checking of object permissions.

If enabled, a caller with the `canUseMonitorActivityTool` user capability must also meet one of the following conditions before calling the `runSpecification()` method against the event management service:

- The account of the caller must match the account credential used to schedule the event.
- The caller must have traverse and execute permissions on the target object.

Data type:

Boolean

Default:

false

emf.schedule.validation.enabled

Validates schedule properties such as start date, end date, data types, and user account credentials when Content Manager processes requests to add or update schedules. Disables invalid schedules.

Details of disabled schedules are logged in log files.

Data type:

Boolean

Default:

false

emf.dls.attachment.timestamp.enabled

When set to true, email attachments have report names with a date time stamp. The default format for the timestamp is: `yyyy.MM.dd`, where `yyyy` is the four-digit year, `MM` is the two-digit month, and `dd` is the two-digit day.

For example, if you attach the report Annual Result in a message, the email that is sent has the following attachment: Annual result - 2014.07.15.pdf.

Set this advanced property if you need to add a date time stamp to report attachments in email. Optionally, change the default `dateTime` format by setting the advanced property `emf.dls.attachment.timestamp.format`.

Data type:

Boolean

Default:

false

emf.dls.attachment.timestamp.format

Specifies the `dateTime` format that is added to report names in email attachments when the `emf.dls.attachment.timestamp.enabled` advanced property is set to true.

Possible values include various date formats. For example, `15.07.2014` has the format `dd.MM.yyyy` and `140704120856-0700` has the format `yyMMddHHmmssZ`. For more information on `SimpleDateFormat`, see the Oracle website. Do not use a slash or special characters in the format.

Data type:

String

Default:

yyyy-MM-dd

emf.preview.max.items

Use this setting to increase the maximum number of events that can be shown in the event list.

Increasing this value can affect the performance of the system which will need to read more data and render the data in the user interface.

Data type:

Integer

Default:

50

Job service advanced settings

This section describes advanced settings for the job service.

primary.wait.js

Specifies the time, in seconds, for the primary wait threshold for the job service.

This value is used if a value is not set in the request.

Data type:

Integer

Default:

120

Metrics manager service advanced settings

This section describes advanced settings for the metrics manager service.

initialConnections

Specifies the number of connections to create when the connection pool is initialized.

Data type:

Integer

Default:

5

Tip: You must restart the service for these settings to take effect.

incrementConnections

Specifies the number of connections to increment when the connections pool must be increased.

Data type:

Integer

Default:

5

Tip: You must restart the service for these settings to take effect.

maximumConnections

Specifies the maximum number of connections this pool can use.

Data type:

Integer

Default:

200

Tip: You must restart the service for these settings to take effect.

Monitor service advanced settings

This section describes advanced settings for the monitor service.

emf.scheduling.priority.capability.check.disabled

When set to True, a scheduled task is run with the priority specified in the task regardless of whether the user has the Schedule Priority capability. When set to False (Default), the tasks schedule will run as the default priority.

This scenario can occur when a Users task is modified by an Administrator to have a higher priority.

Data type:

Boolean

Default:

false

enable.session.affinity

Indicates whether session affinity is enabled.

This setting is used in conjunction with the `session.affinity.services` advanced setting.

Data type:

Boolean

Default:

false

event.check.active

Specifies whether the consistency check is active.

Possible values: 1 for true, 0 (or anything else) for false

Data type:

Integer

Default:

0

event.check.interval

Specifies the interval, in minutes, when a consistency check is made to ensure that the monitor service record of events matches that in Content Store.

An event consistency checker thread cleans up any discrepancies.

Data type:

Integer

Default:

10

event.finished.check.active

Enables or disables the bulk cleanup process of finished tasks in NC tables. The process uses the `BulkFinishedTaskCleanerThread` script. The script is initiated by the monitor service when the service starts as part of the Cognos service startup.

When the system detects that this property is enabled, the cleanup script is loaded from `BulkCleanStmtsObjectFactory`. The script is database-specific and runs in a single transaction to delete any finished records that fulfill the removal criteria.

Data type:

Boolean

Default:
true

event.finished.check.interval

Specifies the interval, in seconds, when the bulk cleanup process checks for finished tasks in NC tables. The tasks that are finished more than 24 hours ago are candidates for the bulk cleanup.

The default is 3600 seconds (1 hour), but ideally it should be 86400 seconds (24 hours).

Data type:
Integer

Default:
3600

event.finished.check.threshold

Defines the maximum number of finished tasks in the NC tables that are selected for removal.

Data type:
Integer

Default:
10

primary.wait.ms

Specifies the primary wait threshold, in seconds, for the monitor service.

This setting is used if a value is not set in the request.

Data type:
Integer

Default:
120

session.affinity.services

If `enable.session.affinity` is set to `true`, this setting specifies the services to configure for session affinity.

In an N/N-1 scenario, this setting is supported by the following IBM Cognos Planning services only: `planningAdministrationConsoleService`, `planningDataService`, `planningRuntimeService`, and `planningTaskService`. Otherwise, in a homogeneous distributed environment, this setting is supported by all services.

To specify the service(s), use the mandatory `serviceName` parameter. To configure multiple services, separate each with a semi-colon (;). Here are two examples:

- `serviceName=planningTaskService`
- `serviceName=planningTaskService;serviceName=planningDataService`

Two optional parameters provide more specific configuration choices:

- `serverGroup`: Specifies the name of the server group.
- `numThreads`: Specifies the maximum number of concurrent tasks allowed. Default is 2.

Parameters must be separated by a comma (,). For example,

```
serviceName=planningTaskService,serverGroup=mygroup,numThreads=4
```

Data type:
String

Default:

None

sds.instance.interval

Specifies the update interval, in seconds, for service instances to register that they are running.

The monitor service uses this mechanism to determine that other monitor services are active. If a monitor service fails, another monitor service can elect to clean up on behalf of the failed service, including updating the history for tasks that failed.

Services can elect to clean up on behalf of another service if that service has not updated its registration within a reasonable time limit. Currently that limit is twice the `sds.instance.interval` setting.

Data type:

Integer

Default:

30

Note:

You must restart the service for this setting to take effect.

enable.tide.metrics.taskqueue

Enables the collection and display of specific metrics for the monitor service in IBM Cognos Administration.

The following metrics are included:

- Time in queue high water mark
- Time in queue low water mark
- Time in queue
- Number of queue requests
- Queue length high water mark
- Queue length low water mark

Data type:

Boolean

Default:

false

sdk.service.poll.interval

The length of time in seconds that the monitor service waits before retrying a client application request to a reconnecting service.

Data type:

Integer

Default:

30

advanced.history.write

Indicates whether final histories are written using the advanced (enhanced) thread pool.

If `true`, the final histories are written using multiple threads. If `false`, the final histories are written on a single thread.

Data type:

Boolean

Default:
true

advanced.parent.history.threads

The number of worker threads used to create root history objects in the content store.

Set `advanced.history.write` to `true` to enable this setting.

Data type:
Integer

Default:
2

Note:

You must restart the service for this setting to take effect.

advanced.child.history.threads

The number of threads used to create child history objects for steps in the content store.

Set `advanced.history.write` to `true` to enable this setting.

Data type:
Integer

Default:
5

Note:

You must restart the service for this setting to take effect.

write.child.histories

Controls the writing of child history objects to the content store.

When `true`, the final history objects for all child tasks are written. When `false`, only the final history object for the root task is written and the history objects for the child tasks are discarded. You can use this setting to improve performance for tasks where child history object write time is very high.

Data type:
Boolean

Default:
true

Note:

You must restart the service for this setting to take effect.

write.child.histories.during.failover

Specifies whether final history objects for a task are written to the content store during a failover.

If the value of `write.child.histories` is set to `true`, child history objects and history objects for root tasks are written.

Data type:
Boolean

Default:
true

Note:

You must restart the service for this setting to take effect.

connection.tracker.use

Tracks connection usage.

When true, java proxy objects are used to track the activities of JDBC objects.

Data type:

Boolean

Default:

false

Note:

You must restart the service for this setting to take effect.

connection.write.maxwaittime

The maximum period of time, in seconds, that an object waits to get a read-write connection from the JDBC connection pool.

Data type:

Integer

Default:

10

Note:

You must restart the service for this setting to take effect.

connection.write.maxConnections

The maximum number of read-write JDBC connections used in the connection pool.

Any value set that is less than the minimum has no effect and the minimum value that is specified is applied.

Minimum value: 5

Data type:

Integer

Default:

10

connection.read.maxwaittime

The maximum period of time, in seconds, that an object waits to get a read-only connection from the JDBC connection pool.

Data type:

Integer

Default:

10

Note:

You must restart the service for this setting to take effect.

connection.read.maxConnections

The maximum number of read-only JDBC connections that are used in the connection pool.

Any value that is set less than the minimum has no effect and the minimum value that is specified is applied.

Data type:

Integer

Default:

8

Note:

You must restart the service for this setting to take effect.

Report service and batch report service advanced settings

This section describes advanced settings for the report service and batch report service.

BDS.split.maxKeysPerChunk

Specifies the maximum key limit for burst reports processing. Setting the key limit lets you avoid complex SQL clauses when the RSVP.BURST_DISTRIBUTION setting is set to `true`. The value of 0 sets no limit on this parameter.

Data type:

Positive integer

Default:

1000

EnableChartTransparencyIE

Specifies whether charts use Internet Explorer display filters to enable transparency.

Data type:

Boolean

Default:`true`

HyperlinkButtonNewWindow

Specifies that when a hyperlink button is clicked, a new window is created.

Data type:

Boolean

Default:`false`

HyperlinkMultipleToolbars

Specifies that duplicate toolbars in HTML reports are permitted. Set to `false` to eliminate duplicate toolbars from appearing.

Data type:

Boolean

Default:`true`

RSVP.ATTACHMENTENCODING.BASE64EXTENDED

Specifies whether base64 encoding is used when generating report output in MHT or XLWA format.

In some instances, if custom applications specify MHT or XLWA output format for reports, problems with end of line characters used in the XML output can prevent applications from opening the report.

Data type:

Boolean

Default:
false

RSVP.BURST_DISTRIBUTION

Specifies whether burst reports run in parallel or sequentially. If you use the default value of `false`, jobs run sequentially, which takes more time.

This setting corresponds to the **Run in parallel** burst option in the user interface. This setting is valid only when **Run in parallel** is set to **Default**. When the **Run in parallel** option is set to **Disabled** or **Enabled**, it overrides this setting.

Data type:
Boolean

Default:
false

RSVP.BURST_QUERY_PREFETCH

When you set this option to `true`, you enable query prefetching. As a result, the burst report outputs are produced much faster because the queries run in parallel with the report rendering. This setting is applicable to dynamic query mode relational models only.

Data type:
Boolean

Default:
false

RSVP.CHARTS.ALTERNATECOLOURS

Specifies that each chart instance assigns colors in palette order, and does not attempt to preserve the color of items from one chart instance to another.

Data type:
Boolean

Default:
false

RSVP.CONCURRENTQUERY.ENABLEDFORINTERACTIVEOUTPUT

Enables concurrent query execution when the report service is producing interactive output.

Data type:
Boolean

Default:
false

RSVP.CONCURRENTQUERY.MAXNUMHELPERSPERREPORT

Specifies the maximum number of query execution helpers for each report. This parameter is used to prevent a single report from consuming all available query execution helpers.

Data type:
Integer

Default:
1

RSVP.CONCURRENTQUERY.NUMHELPERSPERPROCESS

Enables concurrent query execution and set the maximum number of query execution helpers for each report service or batch report service process. The default value is 0, meaning that concurrent query execution is disabled.

Data type:

Integer

Default:

0

RSVP.CSV.DELIMITER

Specifies the field delimiter character used for CSV output.

Data type:

String

Default:

TAB

RSVP.CSV.ENCODING

Specifies the encoding that is used when generating CSV output.

Data type:

String

Default:

utf-16le

RSVP.GROUP_METADATA_REQUESTS

Specifies if metadata requests are grouped, when possible, to improve performance. Users can disable the grouping of metadata requests by setting this parameter to false.

Data type:

Boolean

Default:

true

RSVP.CSV.MIMETYPE

Specifies the MIME type that is attributed to the CSV output.

Data type:

String

Default:

application/vnd.ms-excel/

RSVP.CSV.QUALIFIER

Specifies the string qualifier that is used for CSV output.

Data type:

String

Default:

"

RSVP.CSV.REPEAT_XTAB_LABELS

Specifies whether to repeat the edge labels in a nested crosstab report.

Data type:
Boolean

Default:
false

RSVP.CSV.TERMINATOR

Specifies the line terminator that is used for CSV output.

Data type:
String

Default:
LF

RSVP.DRILL.clearAllMappedParamsOnMismatch

Specifies how mapping of passed parameter values is processed during a drill-through operation when some parameters fail to map. The parameter mapping is continued (default), or all the mapping is discarded and the user is prompted for values.

When you set this property to 1, if any parameter fails to map, all other mapped parameters are removed from the mapping table. This could cause re-prompting for all missing parameters. When you set this property to 0, if any parameter fails to map while the drill-through component attempts to map the parameters, the mapping of the remaining parameters is not affected.

Data type:
Integer

Default:
0

RSVP.CSV.TRIMSPACES

Specifies that trailing spaces are removed from CSV output.

Data type:
Boolean

Default:
false

RSVP.DRILL.DynamicFilterUsesBusinessKey

Specifies dynamic drill-through filter behavior. Set this option to 1 if you want drill-through to generate a filter using the Member Business Key instead of the default Member Caption.

Data type:
Positive integer

Default:
0

RSVP.DRILL.ExtractSourceContextFromRequest

Specifies whether the report server makes an attempt to extract the metadata for the parameters of the drill-through request from the source context of the request instead of issuing a new metadata request. This type of processing improves performance of a drill-through operation. It is turned on by default.

When you set this property to 0, metadata requests are always issued.

Data type:

Integer

Default:

1

RSVP.EXCEL.EXCEL_XLS2007_ENABLE_SHARED_STRINGS_TABLE_SIZE_LIMIT

This setting determines whether RSVP.EXCEL.EXCEL_2007_XLS2007_SHARED_STRINGS_TABLE_SIZE_LIMIT is enabled.

Data type:

Boolean

Default:

true

RSVP.EXCEL.EXCEL_2007_LARGE_WORKSHEET

Enables support for large Microsoft Excel 2007 worksheets. When this option is set to true, worksheets with up to 1,048,576 rows are supported.

Data type:

Boolean

Default:

false

RSVP.EXCEL.EXCEL_2007_OUTPUT_FRAGMENT_SIZE

Adjusts the internal memory fragment size, in rows, that the IBM Cognos Analytics server generates before flushing to a disk. This property can be useful when there are issues, such as running out of memory, when generating reports with the default value. The values might need to be lowered to allow the report to run successfully.

Data type:

Integer

Default:

45000 (approximate)

RSVP.EXCEL.EXCEL_2007_XLS2007_SHARED_STRINGS_TABLE_SIZE_LIMIT

This setting determines whether to limit the shared strings in the Excel output. Limiting shared strings increases the file size of the excel output. When shared strings are unlimited and too high it can cause Excel performance problems.

Data type:

Integer

Default:

10000

RSVP.EXCEL.EXCEL_2007_WORKSHEET_MAXIMUM_ROWS

Specifies the number of rows to output before moving to a new worksheet.

Data type:

Integer

RSVP.EXCEL.PAGEGROUP_WSNAME_ITEMVALUE

Specifies that, when producing output in Microsoft Excel 2007 format and page breaks are specified, the worksheet tabs are named for the data items used to break the pages.

Note: This property does not apply to Analysis Studio.

Data type:
Boolean

Default:
false

RSVP.EXCEL.XLS2007_ALLOW_WRAPPING_SINGLE_CELL

Specifies whether text is wrapped within a cell in Excel outputs.

Data type:
Boolean

Default:
false

RSVP.EXCEL.XLS2007_COLUMN_WIDTH_CONTROL

Prevents cell merging in Excel 2007 report outputs if you set the Size and Overflow values for a column.

Data type:
Boolean

Default:
false

RSVP.EXCEL.XLS2007_FULLDECIMALPRECISION

When set to true, numbers in Excel output are rendered with up to 15 decimal places. When set to false (the default value), numbers can contain a maximum of 12 decimal places.

Data type:
Boolean

Default:
false

RSVP.EXCEL.XLS2007_PRINT_MEDIA

Specifies whether the Don't Print style is applied to Excel 2007 report outputs.

Data type:
Boolean

Default:
true

RSVP.FILE.EXTENSION.XLS

Specifies to use XLS as the file extension on XLS output format email attachments instead of HTML.

Data type:
String

Default:
false

RSVP.PARAMETER_CACHE

Specifies whether parameters caching is enabled or disabled at the server level. By default, parameters caching is enabled.

When RSVP issues a `getParameters` request, it stores the results in a child object under the report object in IBM Cognos Content Manager. This allows the cache to be created or updated without modifying the report specification. When RSVP needs parameter information, it uses the cached information from Content Manager. If the cache does not contain the information required by RSVP, RSVP calls the query engine directly to get the information.

The cache is populated by making a `ReportService getParameters` SOAP request to the batch report service with the run option `http://developer.cognos.com/ceba/constants/runOptionEnum#createParameterCache`. This way, if RSVP determines the cache is missing or stale, creating the cache does not affect the execution of the report, since the cache is created by an independent request. However, since the request is handled by the batch report service, a history entry is created which is visible in the run history of a report.

The cache creation is triggered when a report is created or updated from Cognos Analytics Reporting as well as when a report is executed and RSVP determines the existing cache is stale. RSVP uses the version of the module or root model of the report to determine if the cache is stale.

Data type:

Boolean

Default:

true

RSVP.PARAMETERS.LOG

Specifies whether the report run options and prompt parameters must be logged to the logging system.

Data type:

Boolean

Default:

false

RSVP.PARAMETERS.SAVE

Specifies that report prompt values that are entered by a user are saved automatically.

Data type:

Boolean

Default:

false

RSVP.PRINT.POSTSCRIPT

Specifies which interface to use to print PDF documents from a UNIX operating system. When this option is set to `false`, the Adobe Acrobat PDF interface is used. Otherwise, the internal postscript interface is used.

Data type:

Boolean

Default:

true

RSVP.PROMPT.CASTNUMERICSEARCHKEYTOSTRING

Specifies to convert numeric data items into a string (varchar) format. This may be required if your data source does not convert numeric data items to strings.

Data type:
Boolean

Default:
true

RSVP.PROMPT.EFFECTIVEPROMPTINFO.IGNORE

Disables the issuing of the `effectivePromptInfo` attribute in metadata requests and effectively disables moving the prompt information from under the `caption` attribute of a level to the level itself. This is the default behavior.

Data type:
Boolean

Default:
false

RSVP.PROMPT.RECONCILIATION

Specifies a system-wide configuration that defines how queries and query groups are processed.

See the topic on setting query prioritization in the *IBM Cognos Analytics Administration and Security Guide* for a description of the possible values of this setting.

Data type:
Positiver integer or string

Default:
0 or COMPLETE

RSVP.PROMPT.RECONCILIATION.CHUNKSIZE

Specifies the chunk size when the value of the `RSVP.PROMPT.RECONCILIATION` setting is `CHUNKED`, `GROUPED` or `CHUNKED`.

Data type:
Positive integer

Default:
5

RSVP.PROMPTCACHE.LOCALE

Specifies the locale to use instead of the locale specified in the report whenever prompt cache data is created, updated, or used. This means that a single prompt cache is used for each report regardless of the report user's locale.

Data type:
String

RSVP.RENDER.PDF_FONT_SWITCHING

Specifies that each character in a string is displayed in the preferred font. The preferred font is any font listed in a report specification, followed by the fonts listed in the global styles cascading stylesheet (css) file. When a character is not available in the preferred font, it is displayed using the next font on the list.

In previous versions, a font was used only if all characters in a string could be displayed using that font. Starting with IBM Cognos Business Intelligence 10.1, the preferred font is applied at the character level.

As a result, one word can be displayed using different fonts, or some fonts might be bigger, which can cause word wrapping.

Set the parameter value to `false` to restore the font-choosing behavior of earlier versions.

Data type:

Boolean

Default:

`true`

RSVP.RENDER.ROUNDING

Specifies the rounding rule for data formatting.

In previous versions, the `halfEven` rule was used when rounding numbers. This rule is often used in bookkeeping. However, precision regulations in some regions require different rounding rules, for example, the `halfUp` rule. Starting with version IBM Cognos Business Intelligence 10.2.0, you can choose a rounding rule that complies with the precision regulations in your organization.

The following rounding rules are available:

halfEven

Rounds to the nearest neighbor, where an equidistant value is rounded to the nearest even neighbor.

halfDown

Rounds to the nearest neighbor, where an equidistant value is rounded down.

halfUp

Rounds to the nearest neighbor, where an equidistant value is rounded up.

ceiling

Rounds to a more positive number.

floor

Rounds to a more negative number.

down

Rounds towards zero.

up

Rounds away from zero.

Data type:

String

Default:

`halfEven`

RSVP.RENDER.VALIDATEURL

Specifies whether IBM Cognos Application Firewall validation is imposed on URLs that are contained within a report specification (including URLs on image tags, buttons, hyperlinks, and background images in CSS rules) or are specified by the `cssURL` run option of the report.

When this option is set to `true` and CAF is enabled, validation occurs using the following rules:

- Fully qualified, or absolute URLs:

`protocol://host[:port]/path[?query]`

Where `protocol` is either 'http' or 'https' and the `host` is validated against the valid domain list

- URLs relative to the server installation web root:

`/<install root>/.*`

Where `<install root>` is the gateway file path, taken from the Gateway URI in IBM Cognos Configuration. For example, `/ibmcognos/ps/portal/images/action_delete.gif`

- One of the following specifically allowed URLs:

- `about:blank` (case insensitive)
- `JavaScript>window.close()` (case insensitive, with or without trailing semi-colon)
- `JavaScript>parent.close()` (case insensitive, with or without trailing semi-colon)
- `JavaScript>history.back()` (case insensitive, with or without trailing semi-colon)
- `parent.cancelErrorPage()` (case insensitive, with or without trailing semi-colon)
- `doCancel()` (case insensitive, with or without trailing semi-colon)

Data type:

Boolean

Default:

false

RSVP.REPORTSPEC.LOG

Specifies whether report specifications must be logged to the logging system.

Data type:

Boolean

Default:

false

Repository service advanced settings

This section describes advanced settings for the repository service.

repository.maxCacheDocSize

The maximum size, in MB, of an individual report that can be stored in the cache.

The value must be a positive integer (greater than 0). Reports greater than the specified size will not be cached and will be retrieved from the repository.

Data type:

Integer

Default:

10

Chapter 34. Initial content store settings

This appendix describes the initial access permissions that predefined groups and roles have to predefined objects.

Initial Objects

The following table defines access permissions granted to a particular object.

Symbol	Permission
R	read
Sp	setPolicy
T	traverse Allows user to read the children of the object.
W	write
X	execute

Predefined Objects

Description

This section describes the set of objects created by Content Manager when a content store is initialized.

The content store is organized into seven main areas, each identified by an object that is created under the single root object ["/](#) on page 1614. Each area serves a distinct purpose and is administered by a single [bibus » role](#). Each area is described in further detail.

Content Manager sets a default security policy on many of the objects created during the initialization of the content store. The security policy of an object is stored in the [bibus » baseClass » policies](#) property. Many other objects acquire their security policy from a containing object using property acquisition. When constructing a policy to allow all users access to an object, the ["Everyone" on page 1698 bibus » group](#) is used rather than the ["All Authenticated Users" on page 1689 bibus » group](#) or the ["Anonymous" on page 1690 bibus » account](#).

The policy of the ["/](#) on page 1614 grants every user `read`, `traverse`, and `execute` permissions. Since no user has `write` permission, neither the ["/](#) on page 1614 nor any of its child objects can be deleted.

The Administration area (["Administration" on page 1615](#)) is dedicated to storing objects related to content store maintenance, such as deployment definitions, content maintenance tasks, and the jobs necessary to run these objects on a scheduled basis. ["Controller Administrators" on page 1695](#), ["Metrics Administrators" on page 1700](#), ["PowerPlay Administrators" on page 1705](#), ["Report Administrators" on page 1707](#), and ["Server Administrators" on page 1708](#) have full control of objects in this area.

The Capabilities area (["Capability" on page 1616](#)) is dedicated to the administration of capabilities. Each object contained in this area represents a user capability. To have a capability, a user must be able to navigate to the object representing the user capability (`traverse` permission is required on all ancestors of the object), and the user must have `execute` permission on the object. This area is administered by ["Directory Administrators" on page 1697](#). Members of this [bibus » role](#) also have `setPolicy` permission

on objects in this area. Note that [“Directory Administrators” on page 1697](#) are not granted all user capabilities by default.

The Catalog area ([“Catalog - removed” on page 1669](#)) is dedicated to storing objects used by IBM Cognos Connection. Objects stored in this area contain configuration information related to visualizations for Active Reports. This area is administered by [“Authors” on page 1693](#). Other users have read and traverse permission for objects in this area.

The Configuration area ([“Configuration” on page 1673](#)) is dedicated to storing runtime configuration information used by IBM Cognos Analytics components, such as service configuration parameters, page definitions used to render printed reports, and the default user profile, which is used to initialize a user's preferences and personal folders when they first log on to IBM Cognos Analytics. This area is primarily administered by [“Server Administrators” on page 1708](#), although [“Directory Administrators” on page 1697](#) administer the default user profile. Other users have limited permissions to use objects in this area, such as the page definitions, but do not have read permission for most runtime configuration information.

The Content area ([“Public Folders” on page 1759](#)) is dedicated to storing your applications. This area is administered by [“Controller Administrators” on page 1695](#), [“Metrics Administrators” on page 1700](#), [“Planning Rights Administrators” on page 1703](#), [“PowerPlay Administrators” on page 1705](#), and [“Report Administrators” on page 1707](#). Other users have restricted permissions in this area, based on their role. For example, [“Authors” on page 1693](#) have read, write and execute permissions for every object in this area, but are not able to change the policy on such an object (unless the user is the owner). In contrast, [“Analysis Users” on page 1689](#), [“Consumers” on page 1694](#), [“Controller Users” on page 1696](#), [“Metrics Authors” on page 1700](#), [“Metrics Users” on page 1701](#), [“Planning Contributor Users” on page 1703](#), [“PowerPlay Users” on page 1705](#), and [“Query Users” on page 1706](#) have read and execute permissions for objects in this area only.

The Directory area ([“Directory” on page 1683](#)) is dedicated to storing objects based on your external security providers, as well as security objects used only with IBM Cognos Analytics, such as the Cognos namespace. The Cognos namespace contains the predefined IBM Cognos Analytics `bibus » group` and `bibus » role` objects and the anonymous `bibus » account` object. This area also stores directory information related to data sources and printer definitions used by IBM Cognos Analytics applications. This area is administered by [“Directory Administrators” on page 1697](#). Other users have limited access to most of the objects in this area.

The Portal area ([“Portal” on page 1711](#)) is dedicated to storing objects used by IBM Cognos Connection. Objects stored in this area contain configuration information related to external portlet producers. This area is administered by [“Portal Administrators” on page 1704](#). Other users have read permission for objects in this area.

The transient area ([“Transient” on page 1762](#)) is used to store transient data used by various IBM Cognos Analytics components. Since objects in this area are administered by IBM Cognos Analytics components, all users have restricted permissions to objects in this area.

Object Hierarchy

- [“/” on page 1614](#)
 - [“Administration” on page 1615](#)
 - [“Capability” on page 1616](#)
 - [“Adaptive Analytics” on page 1616](#)
 - [“Administration” on page 1617](#)
 - [“Adaptive Analytics Administration” on page 1618](#)
 - [“Administration tasks” on page 1618](#)
 - [“Configure and manage the system” on page 1619](#)
 - [“Controller Administration” on page 1620](#)
 - [“Data Source Connections” on page 1620](#)
 - [“Directory - obsolete” on page 1621](#)

- [“Distribution Lists and Contacts” on page 1621](#)
- [“Metric Studio Administration” on page 1622](#)
- [“Mobile Administration” on page 1623](#)
- [“My Data Sets Administration” on page 1623](#)
- [“Planning Administration” on page 1624](#)
- [“PowerPlay Servers” on page 1625](#)
- [“Printers” on page 1625](#)
- [“Query Service Administration” on page 1626](#)
- [“Run activities and schedules” on page 1626](#)
- [“Set capabilities and manage UI profiles” on page 1627](#)
- [“Styles and portlets” on page 1628](#)
- [“Users, Groups, and Roles” on page 1629](#)
- [“Analysis Studio” on page 1629](#)
 - [“Open PowerPlay Reports with Analysis Studio” on page 1630](#)
- [“Cognos Insight” on page 1631](#)
- [“Cognos Viewer” on page 1631](#)
 - [“Context Menu” on page 1632](#)
 - [“Run With Options” on page 1633](#)
 - [“Selection” on page 1634](#)
 - [“Toolbar” on page 1634](#)
- [“Collaborate” on page 1635](#)
 - [“Allow collaboration features” on page 1635](#)
 - [“Launch collaboration tools” on page 1636](#)
- [“Controller Studio” on page 1636](#)
- [“Data Manager” on page 1637](#)
- [“Detailed Errors” on page 1637](#)
- [“Drill Through Assistant” on page 1638](#)
- [“Event Studio” on page 1639](#)
- [“EVStudio” on page 1639](#)
- [“Execute Indexed Search” on page 1640](#)
- [“Executive Dashboard” on page 1640](#)
 - [“Use Advanced Dashboard Features” on page 1641](#)
 - [“Use Interactive Dashboard Features” on page 1642](#)
 - [“Use the Edit Features” on page 1643](#)
- [“External Repositories” on page 1644](#)
 - [“Manage repository connections” on page 1644](#)
 - [“View external documents” on page 1645](#)
- [“Generate CSV Output” on page 1645](#)
- [“Generate PDF Output” on page 1646](#)
- [“Generate XLS Output” on page 1647](#)
- [“Generate XML Output” on page 1647](#)
- [“Glossary” on page 1648](#)
- [“Hide Entries” on page 1648](#)

- [“Import relational metadata” on page 1649](#)
- [“Lineage” on page 1650](#)
- [“Manage own data source signons” on page 1650](#)
- [“Metric Studio” on page 1651](#)
 - [“Edit View” on page 1651](#)
- [“Mobile” on page 1652](#)
- [“My Data Sets” on page 1653](#)
- [“Package Data Sources” on page 1653](#)
- [“Planning Contributor” on page 1654](#)
- [“PowerPlay Studio” on page 1655](#)
- [“Query Studio” on page 1655](#)
 - [“Advanced” on page 1656](#)
 - [“Create” on page 1656](#)
- [“Reporting” on page 1657](#)
 - [“Allow External Data” on page 1657](#)
 - [“Bursting” on page 1658](#)
 - [“Create/Delete” on page 1659](#)
 - [“HTML Items in Report” on page 1659](#)
 - [“Open PowerPlay Reports with Reporting” on page 1660](#)
 - [“User Defined SQL” on page 1661](#)
- [“Scheduling” on page 1661](#)
 - [“Schedule by day” on page 1662](#)
 - [“Schedule by hour” on page 1663](#)
 - [“Schedule by minute” on page 1663](#)
 - [“Schedule by month” on page 1664](#)
 - [“Schedule by trigger” on page 1664](#)
 - [“Schedule by week” on page 1665](#)
 - [“Schedule by year” on page 1665](#)
 - [“Scheduling Priority” on page 1665](#)
- [“SDK” on page 1666](#)
- [“Self Service Package Wizard” on page 1667](#)
- [“Set Entry-Specific Capabilities” on page 1667](#)
- [“Specification Execution” on page 1668](#)
- [“Statistics” on page 1668](#)
- [“Watch Rules” on page 1669](#)
- [“Library” on page 1670](#)
 - [“Visualizations” on page 1671](#)
 - [“User Interface Profiles” on page 1672](#)
 - [“Cognos Workspace Do More Profiles” on page 1672](#)
 - [“Reporting Profiles” on page 1673](#)
- [“Catalog - removed” on page 1669](#)
- [“Configuration” on page 1673](#)

- [“11x17” on page 1674](#)
- [“A3” on page 1674](#)
- [“A4” on page 1675](#)
- [“B4 JIS” on page 1676](#)
- [“B5 JIS” on page 1676](#)
- [“Legal” on page 1677](#)
- [“Letter” on page 1677](#)
- [“User Interface Profiles - deprecated” on page 1678](#)
 - [“Reporting Profiles - deprecated” on page 1678](#)
 - [“Express - deprecated” on page 1679](#)
 - [“Professional - deprecated” on page 1680](#)
- [“User Profile” on page 1681](#)
 - [“Most Recently Used list” on page 1681](#)
 - [“My Folders” on page 1682](#)
 - [“My Watch Items” on page 1682](#)
- [“Directory” on page 1683](#)
- [“<Indeterminate>” on page 1684](#)
 - [“<Known User>” on page 1684](#)
 - [“<Indeterminate>” on page 1685](#)
 - [“Most Recently Used List” on page 1686](#)
 - [“My Folders” on page 1686](#)
 - [“My Watch Items” on page 1687](#)
- [“Cognos” on page 1687](#)
 - [“Adaptive Analytics Administrators” on page 1688](#)
 - [“Adaptive Analytics Users” on page 1688](#)
 - [“All Authenticated Users” on page 1689](#)
 - [“Analysis Users” on page 1689](#)
 - [“Anonymous” on page 1690](#)
 - [“<Indeterminate>” on page 1691](#)
 - [“Most Recently Used list” on page 1691](#)
 - [“My Folders” on page 1692](#)
 - [“My Watch Items” on page 1692](#)
 - [“Authors” on page 1693](#)
 - [“Catalog Administrators - removed” on page 1693](#)
 - [“Cognos Insight Users” on page 1694](#)
 - [“Consumers” on page 1694](#)
 - [“Controller Administrators” on page 1695](#)
 - [“Controller Users” on page 1696](#)
 - [“Data Manager Authors” on page 1696](#)
 - [“Directory Administrators” on page 1697](#)
 - [“Everyone” on page 1698](#)
 - [“Express Authors” on page 1698](#)

- [“Library Administrators” on page 1699](#)
- [“Metrics Administrators” on page 1700](#)
- [“Metrics Authors” on page 1700](#)
- [“Metrics Users” on page 1701](#)
- [“Mobile Administrators” on page 1702](#)
- [“Mobile Users” on page 1702](#)
- [“Planning Contributor Users” on page 1703](#)
- [“Planning Rights Administrators” on page 1703](#)
- [“Portal Administrators” on page 1704](#)
- [“PowerPlay Administrators” on page 1705](#)
- [“PowerPlay Users” on page 1705](#)
- [“Query Users” on page 1706](#)
- [“Readers” on page 1707](#)
- [“Report Administrators” on page 1707](#)
- [“Server Administrators” on page 1708](#)
- [“System Administrators” on page 1708](#)
- [“Tenant Administrators” on page 1709](#)
- [“Tenants” on page 1683](#)
- [“Export” on page 1710](#)
- [“Import” on page 1710](#)
- [“Portal” on page 1711](#)
- [“Administration” on page 1711](#)
 - [“Pagelets” on page 1712](#)
 - [“Console” on page 1712](#)
 - [“Configuration” on page 1713](#)
 - [“Content Administration” on page 1713](#)
 - [“Data Source Connections” on page 1714](#)
 - [“Dispatchers and Services” on page 1714](#)
 - [“Distribution Lists and Contacts” on page 1715](#)
 - [“Portlets” on page 1715](#)
 - [“Printers” on page 1716](#)
 - [“Styles” on page 1717](#)
 - [“Index Search” on page 1717](#)
 - [“Index” on page 1718](#)
 - [“Search” on page 1718](#)
 - [“Storage” on page 1719](#)
 - [“Library” on page 1720](#)
 - [“User Interface Profiles” on page 1721](#)
 - [“Visualizations” on page 1721](#)
 - [“PowerPlay” on page 1722](#)
 - [“Security” on page 1722](#)
 - [“Capabilities” on page 1723](#)

- [“User Interface Profiles” on page 1723](#)
- [“Users, Groups, and Roles” on page 1724](#)
- [“Status” on page 1725](#)
 - [“Current Activities” on page 1725](#)
 - [“Data Sets” on page 1726](#)
 - [“Dynamic Cubes” on page 1732](#)
 - [“Past Activities” on page 1726](#)
 - [“Schedules” on page 1727](#)
 - [“System” on page 1727](#)
 - [“Upcoming Activities” on page 1728](#)
- [“Portlets” on page 1728](#)
 - [“Console” on page 1729](#)
 - [“cogadmin” on page 1729](#)
 - [“Capabilities” on page 1730](#)
 - [“Content Administration” on page 1730](#)
 - [“Current Activities” on page 1731](#)
 - [“Data Source Connections” on page 1732](#)
 - [“Data Sets” on page 1732](#)
 - [“Dispatchers and Services” on page 1733](#)
 - [“Distribution Lists and Contacts” on page 1734](#)
 - [“Dynamic Cubes” on page 1734](#)
 - [“Past Activities” on page 1735](#)
 - [“Portlets” on page 1735](#)
 - [“PowerPlay” on page 1736](#)
 - [“Printers” on page 1736](#)
 - [“Profiles Administration” on page 1737](#)
 - [“Schedules” on page 1737](#)
 - [“Styles” on page 1738](#)
 - [“System” on page 1738](#)
 - [“Upcoming Activities” on page 1739](#)
 - [“Users Groups and Roles” on page 1739](#)
 - [“IBM Cognos Go! Search” on page 1740](#)
 - [“IBM Cognos Enhanced Search - command panel” on page 1740](#)
 - [“IBM Cognos Enhanced Search - External Search Results” on page 1741](#)
 - [“IBM Cognos Enhanced Search - main UI” on page 1741](#)
 - [“IBM Cognos Enhanced Search - Refinement Viewer” on page 1742](#)
 - [“IBM Cognos Enhanced Search - Results Viewer” on page 1742](#)
 - [“IBM Cognos Go! Search Admin” on page 1743](#)
 - [“Index” on page 1743](#)
 - [“Search” on page 1744](#)
 - [“Storage” on page 1744](#)
 - [“Multitenancy” on page 1719](#)

- [“Tenants” on page 1720](#)
- [“Connection” on page 1746](#)
- [“Pages” on page 1746](#)
- [“Portlets” on page 1747](#)
 - [“Dashboard” on page 1747](#)
 - [“Multi-page” on page 1748](#)
 - [“IBM Cognos Content” on page 1748](#)
 - [“IBM Cognos Navigator” on page 1749](#)
 - [“IBM Cognos Search” on page 1749](#)
 - [“IBM Cognos Viewer” on page 1750](#)
 - [“IBM Cognos Extended Applications” on page 1750](#)
 - [“IBM Cognos Extended Applications Portlet” on page 1751](#)
 - [“IBM Cognos Metric Studio” on page 1751](#)
 - [“IBM Cognos History Chart” on page 1752](#)
 - [“IBM Cognos Metric List” on page 1753](#)
 - [“IBM Cognos Utility” on page 1753](#)
 - [“Bookmarks Viewer” on page 1754](#)
 - [“HTML Source” on page 1754](#)
 - [“HTML Viewer” on page 1755](#)
 - [“Image Viewer” on page 1755](#)
 - [“RSS Viewer” on page 1756](#)
- [“Styles” on page 1756](#)
 - [“Business” on page 1757](#)
 - [“Classic” on page 1757](#)
 - [“Contemporary” on page 1758](#)
 - [“Corporate” on page 1758](#)
 - [“Modern” on page 1758](#)
 - [“Presentation” on page 1759](#)
- [“Public Folders” on page 1759](#)
- [“Transient” on page 1762](#)

/

The root object contains all objects in the content store. The root object cannot be deleted.

Properties

searchPath

/

objectClass

[bibus](#) » [root](#)

parent

—

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the / object.

securityObject	permissions
“Everyone” on page 1698	X R T

What's new

New in Version 8.4 — [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

Administration

This object contains objects that allow Administrators to automate repetitive tasks, such as deployments.

Properties

searchPath

[/adminFolder](#)

objectClass

[bibus](#) » [adminFolder](#)

parent

[“/” on page 1614](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Administration object.

securityObject	permissions
“Controller Administrators” on page 1695	X R Sp T W
“Library Administrators” on page 1699	X T
“Metrics Administrators” on page 1700	X R Sp T W
“PowerPlay Administrators” on page 1705	X R Sp T W
“Report Administrators” on page 1707	X R Sp T W
“Server Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.2.2 — [“Documentation Updates” on page 1827](#)

The documentation for this object was updated to include Library Administrators as a security object..

New in Version 8.4 — [“PowerPlay 8 Integration” on page 1886](#)

The policy of this object was changed to include the PowerPlay Administrators role.

Capability

This object contains objects that are used to determine the capabilities of all users.

Properties

searchPath

/capability

objectClass

[bibus](#) » [capability](#)

parent

["/](#) on page 1614

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Capability object.

<i>Table 274. Policies for the Capability object.</i>	
securityObject	permissions
"Directory Administrators" on page 1697	Sp T
"Everyone" on page 1698	T

What's new

New in Version 8.4 — ["Bug Fixes" on page 1897](#)

The name of this object has been corrected in the documentation.

Adaptive Analytics

This object is used to determine which users have the [canUseAdaptiveAnalytics](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Adaptive Analytics']

objectClass

[bibus](#) » [securedFunction](#)

parent

["Capability" on page 1616](#)

displaySequence

163

iconURI

access_adaptive_analytics_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Adaptive Analytics object.

<i>Table 275. Policies for the Adaptive Analytics object.</i>	
securityObject	permissions
"Adaptive Analytics Administrators" on page 1688	X T

<i>Table 275. Policies for the Adaptive Analytics object. (continued)</i>	
securityObject	permissions
“Adaptive Analytics Users” on page 1688	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 GA – [“Bug Fixes” on page 1882](#)

The [iconURI](#) property of this object has been corrected in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Administration

This object is used to determine which users have the [canUseAdministrationPortal](#) capability. This secured function provides access to the secured features it contains.

Properties

searchPath

[/capability/securedFunction\[@name='Administration'\]](#)

objectClass

[bibus » securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

135

iconURI

[access_cognos_administration_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Administration object.

<i>Table 276. Policies for the Administration object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T
“Adaptive Analytics Administrators” on page 1688	X T
“Controller Administrators” on page 1695	X T
“Metrics Administrators” on page 1700	X T
“Mobile Administrators” on page 1702	X T
“Planning Rights Administrators” on page 1703	X T
“Portal Administrators” on page 1704	X T
“PowerPlay Administrators” on page 1705	X T
“Report Administrators” on page 1707	X T

<i>Table 276. Policies for the Administration object. (continued)</i>	
securityObject	permissions
“Server Administrators” on page 1708	X T

What's new

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

The policy of this object was changed to include the [“Adaptive Analytics Administrators” on page 1688](#) role.

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

The policy of this object was changed to include the PowerPlay Administrators role.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Adaptive Analytics Administration

This object is used to determine which users have the [canUseAdaptiveAnalyticsAdministration](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Adaptive Analytics Administration']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

5

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Adaptive Analytics Administration object.

<i>Table 277. Policies for the Adaptive Analytics Administration object.</i>	
securityObject	permissions
“Adaptive Analytics Administrators” on page 1688	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Administration tasks

This object is used to determine which users have the [canUseContentStoreTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Administration tasks']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

45

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Administration tasks object.

securityObject	permissions
“Controller Administrators” on page 1695	X T
“Metrics Administrators” on page 1700	X T
“PowerPlay Administrators” on page 1705	X T
“Report Administrators” on page 1707	X T
“Server Administrators” on page 1708	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

The policy of this object was changed to include the PowerPlay Administrators role.

New in Version 8.4 – [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Configure and manage the system

This object is used to determine which users have the [canUseServerAdministrationTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Configure and manage the system']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

70

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Configure and manage the system object.

<i>Table 279. Policies for the Configure and manage the system object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 — “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Controller Administration

This object determines which users have the [canUseControllerAdministration](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Administration']/  
securedFeature[@name='Controller Administration']
```

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

15

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Controller Administration object.

<i>Table 280. Policies for the Controller Administration object.</i>	
securityObject	permissions
“Controller Administrators” on page 1695	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Data Source Connections

This object is used to determine which users have the [canUseDataSourcesTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Data Source Connections']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

55

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Data Source Connections object.

<i>Table 281. Policies for the Data Source Connections object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 8.3 — [“Capabilities Refinements” on page 1918](#)

This object is now created in the content store during upgrade or initialization. It replaces the [“Directory - obsolete” on page 1621](#).

New in Version 8.4 — [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Directory - obsolete

What's new

New in Version 8.3 — [“Capabilities Refinements” on page 1918](#)

This object is obsolete and is no longer created in the content store during upgrade or initialization. Use the following objects instead:

- [“Data Source Connections” on page 1620](#)
- [“Distribution Lists and Contacts” on page 1621](#)
- [“Printers” on page 1625](#)
- [“Users, Groups, and Roles” on page 1629](#)

Distribution Lists and Contacts

This object is used to determine which users have the [canUseDistributionListsAndContactsTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Distribution Lists and Contacts']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

40

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Distribution Lists and Contacts object.

<i>Table 282. Policies for the Distribution Lists and Contacts object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 8.3 — [“Capabilities Refinements” on page 1918](#)

This object is now created in the content store during upgrade or initialization. It replaces the [“Directory - obsolete” on page 1621](#).

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Metric Studio Administration

This object is used to determine which users have the [canUseMetricsManagerAdministration](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Metric Studio Administration']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

10

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Metric Studio Administration object.

<i>Table 283. Policies for the Metric Studio Administration object.</i>	
securityObject	permissions
“Metrics Administrators” on page 1700	X T

<i>Table 283. Policies for the Metric Studio Administration object. (continued)</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Mobile Administration

Reserved.

Properties

searchPath

/capability/securedFunction[@name='Administration']/securedFeature[@name='Mobile Administration']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Mobile Administration object.

<i>Table 284. Policies for the Mobile Administration object.</i>	
securityObject	permissions
“Mobile Administrators” on page 1702	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.1 – [“Administrative changes for Cognos Analytics Mobile Reports” on page 1830](#)

This object is now created in the content store during upgrade or initialization.

My Data Sets Administration

This object is used to determine which users have the [canUseMyDataSetsAdministration](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/securedFeature[@name='My Data Sets Administration']

objectClassbibus » [securedFeature](#)**parent**[“Administration” on page 1617](#)**displaySequence**

0

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the 'My Data Sets Administration' object.

<i>Table 285. Policies for the 'My Data Sets Administration' object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.2.2 – “My data sets” on page 1821**

This object was added.

Planning Administration

This object is used to determine which users have the [canUsePlanningAdministration](#).

Properties**searchPath**

```
/capability/securedFunction[@name='Administration']/
securedFeature[@name='Planning Administration']
```

objectClassbibus » [securedFeature](#)**parent**[“Administration” on page 1617](#)**displaySequence**

20

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Planning Administration object.

<i>Table 286. Policies for the Planning Administration object.</i>	
securityObject	permissions
“Planning Rights Administrators” on page 1703	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.4 – “Bug Fixes” on page 1897**

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

PowerPlay Servers

This object is used to determine which users have the [canUsePowerPlayAdministration](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='PowerPlay Servers']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

25

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the PowerPlay Servers object.

securityObject	permissions
“PowerPlay Administrators” on page 1705	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Printers

This object is used to determine which users have the [canUsePrintersTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Printers']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

35

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Printers object.

<i>Table 288. Policies for the Printers object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 8.3 — “Capabilities Refinements” on page 1918

This object is now created in the content store during upgrade or initialization. It replaces the [“Directory - obsolete” on page 1621](#).

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Query Service Administration

This object is used to determine which users have the [canUseQueryServiceTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Query Service Administration']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Query Service Administration object.

<i>Table 289. Policies for the Query Service Administration object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 — “Query Service Administration Task” on page 1869

This object is now created in the content store during upgrade or initialization.

Run activities and schedules

This object is used to determine which users have the [canUseMonitorActivityTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Run activities and schedules']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

75

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Run activities and schedules object.

securityObject	permissions
“Controller Administrators” on page 1695	X T
“Metrics Administrators” on page 1700	X T
“PowerPlay Administrators” on page 1705	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 — [“Capabilities Refinements” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 — [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Set capabilities and manage UI profiles

This object is used to determine which users have the [canUseCapabilitiesTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Set capabilities and manage UI profiles']

objectClass

bibus » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

60

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Set capabilities and manage UI profiles object.

<i>Table 291. Policies for the Set capabilities and manage UI profiles object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 8.4 – “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Styles and portlets

This object is used to determine which users have the [canUsePortalAdministrationTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Styles and portlets']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

30

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Styles and portlets object.

<i>Table 292. Policies for the Styles and portlets object.</i>	
securityObject	permissions
“Library Administrators” on page 1699	X T
“Portal Administrators” on page 1704	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

The documentation for this object was updated to include Library Administrators as a security object..

New in Version 8.4 – “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Users, Groups, and Roles

This object is used to determine which users have the [canUseUsersGroupsAndRolesTool](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Administration']/
securedFeature[@name='Users, Groups, and Roles']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Administration” on page 1617](#)

displaySequence

65

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Users, Groups, and Roles object.

<i>Table 293. Policies for the Users, Groups, and Roles object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 8.3 – “Capabilities Refinements” on page 1918

This object is now created in the content store during upgrade or initialization. It replaces the [“Directory - obsolete” on page 1621](#).

New in Version 8.4 – “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Analysis Studio

This object is used to determine which users have the [canUseAnalysisStudio](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Analysis Studio']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

150

iconURI

access_analysis_studio_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Analysis Studio object.

<i>Table 294. Policies for the Analysis Studio object.</i>	
securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

Open PowerPlay Reports with Analysis Studio

This object is used to determine which users have the [canOpenPowerPlayInAnalysisStudio](#) capability.

Properties**searchPath**

/capability/securedFunction[@name='Analysis Studio']/
securedFeature[@name='Open PowerPlay Reports with Analysis Studio']

objectClass

bibus » [securedFeature](#)

parent

[“Analysis Studio” on page 1629](#)

displaySequence

15

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Open PowerPlay Reports with Analysis Studio object.

<i>Table 295. Policies for the Open PowerPlay Reports with Analysis Studio object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.4 – “Migration Capabilities” on page 1891**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

New in Version 10.2.0 – “Deprecation of IBM Cognos PowerPlay capabilities and objects” on page 1839

This object is deprecated and is no longer created in the content store during upgrade or initialization.

Cognos Insight

This object is used to determine which users have the [canUseCognosInsight](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Cognos Insight']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

access_cognos_desktop_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Cognos Insight object.

<i>Table 296. Policies for the Cognos Insight object.</i>	
securityObject	permissions
“Cognos Insight Users” on page 1694	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.0 – “Restricting access to Cognos Insight in IBM Cognos Analytics” on page 1841

This object is now created in the content store during upgrade or initialization.

Cognos Viewer

This object is used to determine which users have the [canUseCognosViewer](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Cognos Viewer']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

180

iconURI

access_cognos_viewer_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Cognos Viewer object.

Table 297. Policies for the Cognos Viewer object.

securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Express Authors” on page 1698	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Readers” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – [“New Security Roles” on page 1918](#)

The policy of this object was changed to include the roles [“Express Authors” on page 1698](#) and [“Readers” on page 1707](#).

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

The policy of this object was changed to include the PowerPlay Administrators role and the PowerPlay Users role.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Context Menu

This object is used to determine which users have the [canUseCognosViewerContextMenu](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Cognos Viewer']/
securedFeature[@name='Context Menu']
```

objectClass

bibus » [securedFeature](#)

parent

[“Cognos Viewer” on page 1631](#)

displaySequence

20

iconURI

—

Policies are acquired from [“Cognos Viewer” on page 1631](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Run With Options

This object is used to determine which users have the [canUseCognosViewerRunWithOptions](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Cognos Viewer']/securedFeature[@name='Run With Options']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Cognos Viewer” on page 1631](#)

displaySequence

10

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Run With Options object.

securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Express Authors” on page 1698	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – [“New Security Roles” on page 1918](#)

This object now has a security policy.

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

The policy of this object was changed to include the PowerPlay Administrators role and the PowerPlay Users role.

New in Version 8.4 – [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Selection

This object is used to determine which users have the [canUseCognosViewerSelection](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Cognos Viewer']/
securedFeature[@name='Selection']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Cognos Viewer” on page 1631](#)

displaySequence

30

iconURI

—

Policies are acquired from [“Cognos Viewer” on page 1631](#).

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Toolbar

This object is used to determine which users have the [canUseCognosViewerToolbar](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Cognos Viewer']/
securedFeature[@name='Toolbar']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Cognos Viewer” on page 1631](#)

displaySequence

40

iconURI

—

Policies are acquired from [“Cognos Viewer” on page 1631](#).

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Collaborate

This object is used to determine which users have the [canCollaborate](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Collaborate']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

access_collaborate_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Collaborate object.

securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Express Authors” on page 1698	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 — [“Collaboration Tool Integration” on page 1867](#)

This object is now created in the content store during upgrade or initialization.

Allow collaboration features

This object is used to determine which users have the [canUseCollaborationFeatures](#) capability.

This secured feature controls access to the **Collaborate** icon and to IBM Connections **Search? Results** within IBM Cognos Workspace.

Properties

searchPath

/capability/securedFunction[@name='Collaborate']/securedFeature[@name='Allow collaboration features']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Collaborate” on page 1635](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Collaborate” on page 1635](#).

What's new**New in Version 10.1.0 — “Collaboration Tool Integration” on page 1867**

This object is now created in the content store during upgrade or initialization.

Launch collaboration tools

This object is used to determine which users have the [canLaunchCollaborationTools](#) capability.

The secured feature allows users to launch IBM Connections from any launch menu within the IBM Cognos IBM Cognos Analytics environment, including the IBM Cognos Workspace **Actions Menu**. The **Actions Menu** is available from the **Getting Started Page** and within dashboard reports. The links will go to the user's IBM Connections home page, if it is configured, or to **Activities**.

Properties**searchPath**

```
/capability/securedFunction[@name='Collaborate']/  
securedFeature[@name='Launch collaboration tools']
```

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Collaborate” on page 1635](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Collaborate” on page 1635](#).

What's new**New in Version 10.1.0 — “Collaboration Tool Integration” on page 1867**

This object is now created in the content store during upgrade or initialization.

Controller Studio

This object determines which users have the [canUseControllerStudio](#) capability.

Properties**searchPath**

```
/capability/securedFunction[@name='Controller Studio']
```

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

170

iconURI

access_controller_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Controller Studio object.

<i>Table 300. Policies for the Controller Studio object.</i>	
securityObject	permissions
“Controller Administrators” on page 1695	X T
“Controller Users” on page 1696	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

Data Manager

This object is used to determine which users have the [canUseDataManager](#) capability.

Properties**searchPath**

/capability/securedFunction[@name='Data Manager']

objectClassbibus » [securedFunction](#)**parent**[“Capability” on page 1616](#)**displaySequence**

130

iconURI

access_data_manager_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Data Manager object.

<i>Table 301. Policies for the Data Manager object.</i>	
securityObject	permissions
“Data Manager Authors” on page 1696	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

Detailed Errors

This object is used to determine which users have the [canReceiveDetailedErrors](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Detailed Errors']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

110

iconURI

access_detailed_error_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Detailed Errors object.

<i>Table 302. Policies for the Detailed Errors object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Drill Through Assistant

This object is used to determine which users have the [canUseDrillThroughAssistant](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Drill Through Assistant']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

120

iconURI

access_drill_through_assistant_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Drill Through Assistant object.

<i>Table 303. Policies for the Drill Through Assistant object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – [“Drill Through Assistant Capability” on page 1913](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Event Studio

This object is used to determine which users have the [canUseEventStudio](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Event Studio']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

140

iconURI

access_event_studio_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Event Studio object.

securityObject	permissions
“Authors” on page 1693	X T
“Metrics Administrators” on page 1700	X T
“Metrics Authors” on page 1700	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

EVStudio

This object is used to determine which users have the [canUseEV](#) capability.

Properties

searchPath

/capability/securedFunction[@name='EVStudio']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

The following table lists the security objects and permissions that comprise the policies for the EVStudio object.

<i>Table 305. Policies for the EVStudio object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Execute Indexed Search

This object is used to determine which users have the [canUseIndexSearch](#) capability.

This object is not created during content store initialization. It is created by the [indexSearchService](#) service when required.

Properties

searchPath

[/capability/securedFunction\[@name='Execute Indexed Search'\]](#)

objectClass

[bibus » securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

[access_search_grant.gif](#)

Policies are acquired from [“Capability” on page 1616](#).

What's new

New in Version 8.3 – [“Search Capability” on page 1920](#)

The documentation for this object was updated to specify the correct class.

New in Version 10.1.0 – [“Object Updates” on page 1867](#)

The [iconURI](#) property of this object has been updated.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Executive Dashboard

This object is used to determine which users have the [canUseDashboardViewer](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Executive Dashboard']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

access_dashboard_viewer_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Executive Dashboard object.

securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Express Authors” on page 1698	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Readers” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 — [“Dashboards” on page 1894](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Updates” on page 1867](#)

The [iconURI](#) property of this object has been updated.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.2.0 — [“Graduated dashboard capabilities” on page 1835](#)

This object is now created in the content store during upgrade or initialization.

Use Advanced Dashboard Features

This object is used to determine which users have the [canUseAdvancedDashboardFeatures](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Executive Dashboard']/
securedFeature[@name='Use Advanced Dashboard Features']

objectClass

bibus » [securedFeature](#)

parent

[“Executive Dashboard” on page 1640](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Use Advanced Dashboard Features object.

<i>Table 307. Policies for the Use Advanced Dashboard Features object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.0 – [“Graduated dashboard capabilities” on page 1835](#)

This object is now created in the content store during upgrade or initialization.

Use Interactive Dashboard Features

This object is used to determine which users have the [canUseInteractiveDashboardFeatures](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Executive Dashboard']/
securedFeature[@name='Use Interactive Dashboard Features']

objectClass

bibus » [securedFeature](#)

parent

[“Executive Dashboard” on page 1640](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Use Interactive Dashboard Features object.

<i>Table 308. Policies for the Use Interactive Dashboard Features object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.0 – [“Graduated dashboard capabilities” on page 1835](#)

This object is now created in the content store during upgrade or initialization.

Use the Edit Features

This object is used to determine which users have the [canUseDashboardViewerFileManagement](#) capability.

This object is not created during content store initialization. It is created by the dashboard components when required.

Properties

searchPath

```
/capability/securedFunction[@name='Executive Dashboard']/
securedFeature[@name='Use the Edit Features']
```

objectClass

bibus » [securedFeature](#)

parent

[“Executive Dashboard” on page 1640](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Use the Edit Features object.

<i>Table 309. Policies for the Use the Edit Features object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Dashboards” on page 1894](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace” on page 1858

This object is obsolete and is no longer created in the content store during upgrade or initialization.

External Repositories

This object is used to determine which users have the [canUseRepository](#) capability.

Properties

searchPath

/capability/securedFunction[@name='External Repositories']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the External Repositories object.

securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

Manage repository connections

This object is used to determine which users have the [canUpdateRepositoryRules](#) capability.

Properties

searchPath

/capability/securedFunction[@name='External Repositories']/
securedFeature[@name='Manage repository connections']

objectClass

bibus » [securedFeature](#)

parent

[“External Repositories” on page 1644](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Manage repository connections object.

<i>Table 311. Policies for the Manage repository connections object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

View external documents

This object is used to determine which users have the [canViewContentInRepository](#) capability.

Properties

searchPath

`/capability/securedFunction[@name='External Repositories']/securedFeature[@name='View external documents']`

objectClass

[bibus](#) » [securedFeature](#)

parent

[“External Repositories” on page 1644](#)

displaySequence

—

iconURI

—

Policies are acquired from [“External Repositories” on page 1644](#).

What's new

New in Version 10.1.0 – “IBM Cognos Content Manager/Enterprise Content Management Integration” on page 1866

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.1 – “Support for IBM Cognos Content Archival” on page 1845

This previously reserved feature, introduced in Version 10.1.0, is now available for use.

Generate CSV Output

This object is used to determine which users have the [canGenerateCSVOutput](#) capability.

Properties

searchPath

`/capability/securedFunction[@name='Generate CSV Output']`

objectClassbibus » [securedFunction](#)**parent**[“Capability” on page 1616](#)**displaySequence**

0

iconURI[access_generate_csv_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Generate CSV Output object.

<i>Table 312. Policies for the Generate CSV Output object.</i>	
securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.2.2 – Report output format restriction**

This object was added.

Generate PDF Output

This object is used to determine which users have the [canGeneratePDFOutput](#) capability.

Properties**searchPath**[/capability/securedFunction\[@name='Generate PDF Output'\]](#)**objectClass**bibus » [securedFunction](#)**parent**[“Capability” on page 1616](#)**displaySequence**

0

iconURI[access_generate_pdf_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Generate PDF Output object.

<i>Table 313. Policies for the Generate PDF Output object.</i>	
securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.2.2 – Report output format restriction**

This object was added.

Generate XLS Output

This object is used to determine which users have the [canGenerateXLSOutput](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Generate XLS Output']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

0

iconURI

access_generate_xls_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Generate XLS Output object.

securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – Report output format restriction

This object was added.

Generate XML Output

This object is used to determine which users have the [canGenerateXMLOutput](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Generate XML Output']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

0

iconURI

access_generate_xml_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Generate XML Output object.

securityObject	permissions
“Everyone” on page 1698	X T

<i>Table 315. Policies for the Generate XML Output object. (continued)</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – [Report output format restriction](#)

This object was added.

Glossary

This object is used to determine which users have the [canUseGlossary](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Glossary']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

[access_glossary_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Glossary object.

<i>Table 316. Policies for the Glossary object.</i>	
securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Support for IBM® WebSphere® Business Glossary” on page 1897](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Updates” on page 1867](#)

The [iconURI](#) property of this object has been updated.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Hide Entries

This object is used to determine which users have the [canUseShowHiddenObjectsPreference](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Hide Entries']

objectClass

[bibus » securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

112

iconURI

[access_show_hidden_entries_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Hide Entries object.

<i>Table 317. Policies for the Hide Entries object.</i>	
securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.4 – [“Hiding Objects in the Portal” on page 1883](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Import relational metadata

This object is used to determine which users have the [canImportRelationalMetadata](#) capability.

Properties**searchPath**

[/capability/securedFunction\[@name='Import relational metadata'\]](#)

objectClass

[bibus » securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

[access_import_relational_metadata_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Import relational metadata object.

<i>Table 318. Policies for the Import relational metadata object.</i>	
securityObject	permissions
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – [“Relational Metadata Service” on page 1858](#)

This object is now created in the content store during upgrade or initialization.

Lineage

This object is used to determine which users have the [canUseLineage](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Lineage']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

117

iconURI

access_lineage_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Lineage object.

securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“Lineage Metadata” on page 1891](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Manage own data source signons

This object is used to determine which users have the [canUsePersonalDataSourceCredentials](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Manage own data source signons']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

95

iconURI

access_manage_own_signons_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Manage own data source signons object.

<i>Table 320. Policies for the Manage own data source signons object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – [“Personal Data Source Credentials” on page 1851](#)**

This object is now created in the content store during upgrade or initialization.

Metric Studio

This object is used to determine which users have the [canUseMetricStudio](#) capability.

Properties**searchPath**

/capability/securedFunction[@name='Metric Studio']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

165

iconURI

access_metric_studio_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Metric Studio object.

<i>Table 321. Policies for the Metric Studio object.</i>	
securityObject	permissions
“Metrics Administrators” on page 1700	X T
“Metrics Authors” on page 1700	X T
“Metrics Users” on page 1701	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Edit View

This object is used to determine which users have the [canUseMetricStudioEditView](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Metric Studio']/
securedFeature[@name='Edit View']

objectClass

bibus » [securedFeature](#)

parent

[“Metric Studio” on page 1651](#)

displaySequence

27

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Edit View object.

<i>Table 322. Policies for the Edit View object.</i>	
securityObject	permissions
“Metrics Administrators” on page 1700	X T
“Metrics Authors” on page 1700	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Mobile

This object is used to determine which users have the [canUseMobileService](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Mobile']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

access_mobile_service_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Mobile object.

<i>Table 323. Policies for the Mobile object.</i>	
securityObject	permissions
“Mobile Users” on page 1702	X T
“Mobile Administrators” on page 1702	X T

<i>Table 323. Policies for the Mobile object. (continued)</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.0 – “Restricting access to Cognos Analytics Mobile Reports Service in IBM Cognos Analytics” on page 1842

This object is now created in the content store during upgrade or initialization.

My Data Sets

This object is used to determine which users have the [canUseMyDataSets](#) capability.

Properties

- searchPath**
/capability/securedFunction[@name='My Data Sets']
- objectClass**
[bibus](#) » [securedFunction](#)
- parent**
[“Capability” on page 1616](#)
- displaySequence**
0
- iconURI**
access_my_data_sets_grant.gif

The following table lists the security objects and permissions that comprise the policies for the My Data Sets object.

<i>Table 324. Policies for the My Data Sets object.</i>	
securityObject	permissions
“Everyone” on page 1698	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – [My data sets](#)

This object was added.

Package Data Sources

Reserved.

Properties

- searchPath**
/capability/securedFunction[@name='Package Data Sources']
- objectClass**
[bibus](#) » [securedFunction](#)
- parent**
[“Capability” on page 1616](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Package Data Sources object.

<i>Table 325. Policies for the Package Data Sources object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 — [“Package Data Sources” on page 1895](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Planning Contributor

Reserved.

Properties

searchPath

/capability/securedFunction[@name='Planning Contributor']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

175

iconURI

access_contributor_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Planning Contributor object.

<i>Table 326. Policies for the Planning Contributor object.</i>	
securityObject	permissions
“Planning Rights Administrators” on page 1703	X T
“Planning Contributor Users” on page 1703	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

PowerPlay Studio

This object is used to determine which users have the [canUsePowerPlay](#) capability.

Properties

searchPath

/capability/securedFunction[@name='PowerPlay Studio']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

155

iconURI

access_powerplay_studio_grant.gif

The following table lists the security objects and permissions that comprise the policies for the PowerPlay Studio object.

securityObject	permissions
“Authors” on page 1693	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Query Studio

This object is used to determine which users have the [canUseQueryStudio](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Query Studio']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

160

iconURI

access_query_studio_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Query Studio object.

<i>Table 328. Policies for the Query Studio object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Advanced

This object is used to determine which users have the [canUseQueryStudioAdvancedMode](#) capability.

Properties

searchPath

`/capability/securedFunction[@name='Query Studio']/
securedFeature[@name='Advanced']`

objectClass

bibus » [securedFeature](#)

parent

[“Query Studio” on page 1655](#)

displaySequence

10

iconURI

—

Policies are acquired from [“Query Studio” on page 1655](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Create

This object is used to determine which users have the [canUseQueryStudioFileManagement](#) capability.

Properties

searchPath

`/capability/securedFunction[@name='Query Studio']/
securedFeature[@name='Create']`

objectClass

bibus » [securedFeature](#)

parent

[“Query Studio” on page 1655](#)

displaySequence

15

iconURI

—

Policies are acquired from [“Query Studio” on page 1655](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Reporting

This object is used to determine which users have the [canUseReportStudio](#) capability.

Properties

searchPath

[/capability/securedFunction\[@name='Reporting'\]](#)

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

145

iconURI

[access_report_studio_grant.gif](#)

The following table lists the security objects and permissions that comprise the policies for the Reporting object.

securityObject	permissions
“Authors” on page 1693	X T
“Express Authors” on page 1698	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – [“New Security Roles” on page 1918](#)

The policy of this object was changed to include the role [“Express Authors” on page 1698](#).

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Allow External Data

This object is used to determine which users have the [canUseExternalData](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='Allow External Data']

objectClass

bibus » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

40

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Allow External Data object.

<i>Table 330. Policies for the Allow External Data object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.1.0 – [“External Data” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Bursting

This object is used to determine which users have the [canUseBursting](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='Bursting']

objectClass

bibus » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

20

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Bursting object.

<i>Table 331. Policies for the Bursting object.</i>	
securityObject	permissions
“Authors” on page 1693	X T

Table 331. Policies for the Bursting object. (continued)

securityObject	permissions
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – [“New Security Roles” on page 1918](#)

This object now has a security policy.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Create/Delete

This object is used to determine which users have the [canUseReportStudioFileManagement](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='Create/Delete']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

25

iconURI

—

Policies are acquired from [“Reporting” on page 1657](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

HTML Items in Report

This object is used to determine which users have the [canUseHTML](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='HTML Items in Report']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

10

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the HTML Items in Report object.

securityObject	permissions
“Authors” on page 1693	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 — [“New Security Roles” on page 1918](#)

This object now has a security policy.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Open PowerPlay Reports with Reporting

This object is used to determine which users have the [canOpenPowerPlayInReportStudio](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='Open PowerPlay Reports with Reporting']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

35

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Open PowerPlay Reports with Reporting object.

securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 — [“Migration Capabilities” on page 1891](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.2.0 – “Deprecation of IBM Cognos PowerPlay capabilities and objects” on page 1839

This object is deprecated and is no longer created in the content store during upgrade or initialization.

User Defined SQL

This object is used to determine which users have the [canUseUserDefinedSQL](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Reporting']/securedFeature[@name='User Defined SQL']

objectClass

bibus » [securedFeature](#)

parent

[“Reporting” on page 1657](#)

displaySequence

15

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the User Defined SQL object.

securityObject	permissions
“Authors” on page 1693	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 – “New Security Roles” on page 1918

This object now has a security policy.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Scheduling

This object is used to determine which users have the [canUseScheduling](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Scheduling']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

115

iconURI

access_schedule_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Scheduling object.

<i>Table 335. Policies for the Scheduling object.</i>	
securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Controller Administrators” on page 1695	X T
“Controller Users” on page 1696	X T
“Express Authors” on page 1698	X T
“Metrics Administrators” on page 1700	X T
“Metrics Authors” on page 1700	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.3 – “New Security Roles” on page 1918**

The policy of this object was changed to include the role [“Express Authors” on page 1698](#).

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

The policy of this object was changed to include the PowerPlay Administrators role and the PowerPlay Users role.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Schedule by day

This object is used to determine which users have the [canUseSchedulingByDay](#) capability.

Properties**searchPath**

```
/capability/securedFunction[@name='Scheduling']/
securedFeature[@name='Schedule by day']
```

objectClass

```
bibus » securedFeature
```

parent

```
“Scheduling” on page 1661
```

displaySequence

7

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by hour

This object is used to determine which users have the [canUseSchedulingByHour](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Scheduling']/  
securedFeature[@name='Schedule by hour']
```

objectClass

bibus » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

8

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by minute

This object is used to determine which users have the [canUseSchedulingByMinute](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Scheduling']/  
securedFeature[@name='Schedule by minute']
```

objectClass

bibus » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

9

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by month

This object is used to determine which users have the [canUseSchedulingByMonth](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Scheduling']/
securedFeature[@name='Schedule by month']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

5

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by trigger

This object is used to determine which users have the [canUseSchedulingByTrigger](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Scheduling']/
securedFeature[@name='Schedule by trigger']

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

3

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by week

This object is used to determine which users have the [canUseSchedulingByWeek](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Scheduling']/  
securedFeature[@name='Schedule by week']
```

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

6

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Schedule by year

This object is used to determine which users have the [canUseSchedulingByYear](#) capability.

Properties

searchPath

```
/capability/securedFunction[@name='Scheduling']/  
securedFeature[@name='Schedule by year']
```

objectClass

[bibus](#) » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

4

iconURI

—

Policies are acquired from [“Scheduling” on page 1661](#).

What's new

New in Version 10.1.0 – [“Flexible Scheduling” on page 1851](#)

This object is now created in the content store during upgrade or initialization.

Scheduling Priority

This object determines which users have the [canUseSchedulingPriority](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Scheduling']/
securedFeature[@name='Scheduling Priority']

objectClass

bibus » [securedFeature](#)

parent

[“Scheduling” on page 1661](#)

displaySequence

10

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Scheduling Priority object.

securityObject	permissions
“Controller Administrators” on page 1695	X T
“Metrics Administrators” on page 1700	X T
“PowerPlay Administrators” on page 1705	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.3 — “Schedule Priority” on page 1912

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

The policy of this object was changed to include the PowerPlay Administrators role.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

SDK

Properties

searchPath

/capability/securedFunction[@name='SDK']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the SDK object.

<i>Table 337. Policies for the SDK object.</i>	
securityObject	permissions
“Everyone” on page 1698	~X T
“Directory Administrators” on page 1697	Sp T

Self Service Package Wizard

This object is used to determine which users have the [canUseSelfServicePackageWizard](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Self Service Package Wizard']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

—

iconURI

access_ss_package_wizard_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Self Service Package Wizard object.

<i>Table 338. Policies for the Self Service Package Wizard object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X Sp T

What's new

New in Version 10.1.0 – [“Personal Packages” on page 1863](#)

This object is now created in the content store during upgrade or initialization.

Set Entry-Specific Capabilities

This object is used to determine which users have the [canUseObjectCapabilities](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Set Entry-Specific Capabilities']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

120

iconURI

access_entry_specific_capabilities_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Set Entry-Specific Capabilities object.

<i>Table 339. Policies for the Set Entry-Specific Capabilities object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.4 – “Object Capabilities” on page 1885**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Specification Execution

This object is used to determine which users have the [canUseSpecifications](#) capability.

Properties**searchPath**

/capability/securedFunction[@name='Specification Execution']

objectClass

[bibus](#) » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

100

iconURI

access_inline_specification_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Specification Execution object.

<i>Table 340. Policies for the Specification Execution object.</i>	
securityObject	permissions
“Data Manager Authors” on page 1696	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 10.1.0 – “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

Statistics

This object is used to determine which users have the [canUseDescriptiveStatistics](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Statistics']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

116

iconURI

access_statistics_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Statistics object.

<i>Table 341. Policies for the Statistics object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 8.4 GA – [“Statistics” on page 1881](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Catalog - removed

Contains objects that store instances of [bibus » catalogFolder](#) objects and [bibus » visualization](#) modules required to create visualizations used in reports. This object has been replaced by the [Library](#) object.

What's new

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was updated.

New in Version 10.2.1 – [“Visualization support” on page 1831](#)

This object is now created in the content store during upgrade or initialization.

Watch Rules

This object determines which users have the [canUseConditionalSubscriptions](#) capability.

Properties

searchPath

/capability/securedFunction[@name='Watch Rules']

objectClass

bibus » [securedFunction](#)

parent

[“Capability” on page 1616](#)

displaySequence

125

iconURI

access_watch_rule_grant.gif

The following table lists the security objects and permissions that comprise the policies for the Watch Rules object.

Table 342. Policies for the Watch Rules object.

securityObject	permissions
“Analysis Users” on page 1689	X T
“Authors” on page 1693	X T
“Consumers” on page 1694	X T
“Controller Administrators” on page 1695	X T
“Controller Users” on page 1696	X T
“Express Authors” on page 1698	X T
“Metrics Administrators” on page 1700	X T
“Metrics Authors” on page 1700	X T
“PowerPlay Administrators” on page 1705	X T
“PowerPlay Users” on page 1705	X T
“Query Users” on page 1706	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new**New in Version 8.3 — “Conditional Subscriptions” on page 1899**

This object is now created in the content store during upgrade or initialization.

New in Version 8.3 — “New Security Roles” on page 1918

The policy of this object was changed to include the role [“Express Authors” on page 1698](#).

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

The policy of this object was changed to include the PowerPlay Administrators role and the PowerPlay Users role.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Library

Contains objects that store instances of [bibus » catalogFolder](#) objects, [bibus » userInterfaceProfile](#) objects, and [bibus » visualization](#) objects.

Properties**searchPath**

/catalog

objectClassbibus » [catalog](#)**parent**["/](#) on page 1614**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Catalog object.

<i>Table 343. Policies for the Library object.</i>	
securityObject	permissions
“Library Administrators” on page 1699	X R Sp T W
“Everyone” on page 1698	X T

What's new**New in Version 10.2.2 – [“Reporting user interface profiles” on page 1825](#)**

This object was updated to include Reporting user interface profiles.

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was added.

New in Version 10.2.1 – [“Visualization support” on page 1831](#)

This object is now created in the content store during upgrade or initialization.

Visualizations

This folder can be used to store reusable visualizations.

Properties**searchPath**

/catalog/catalogFolder[@name='Visualizations']

objectClassbibus » [catalogFolder](#)**parent**[“Library” on page 1720](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Visualizations object.

<i>Table 344. Policies for the Visualizations object.</i>	
securityObject	permissions
“Library Administrators” on page 1699	X R Sp T W
“Everyone” on page 1698	R X T

What's new

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was updated.

New in Version 10.2.1 – [“Visualization support” on page 1831](#)

This object is now created in the content store during upgrade or initialization.

User Interface Profiles

This folder can be used to store folders of Reporting and Cognos Workspace user interface profiles.

Properties

searchPath

[/catalog/catalogFolder\[@name='User Interface Profiles'\]](#)

objectClass

[bibus](#) » [catalogFolder](#)

parent

[“Library” on page 1720](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the User Interface Profiles object.

securityObject	permissions
“Library Administrators” on page 1699	X R Sp T W
“Everyone” on page 1698	R X T

What's new

New in Version 10.2.2 – [“Reporting user interface profiles” on page 1825](#)

This object was added.

Cognos Workspace Do More Profiles

This folder can be used to store Cognos Workspace user interface profiles.

Properties

searchPath

[/catalog/catalogFolder\[@name='User Interface Profiles'\]/
catalogFolder\[@name='Cognos Workspace Do More Profiles'\]](#)

objectClass

[bibus](#) » [catalogFolder](#)

parent

[User Interface Profiles](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Cognos Workspace Do More Profiles object.

<i>Table 346. Policies for the Cognos Workspace Do More Profiles object.</i>	
securityObject	permissions
“Library Administrators” on page 1699	X R Sp T W
“Everyone” on page 1698	R X T

What's new

New in Version 10.2.2 – [“Reporting user interface profiles” on page 1825](#)

This object was added.

Reporting Profiles

This folder can be used to store Reporting user interface profiles.

Properties

searchPath

/catalog/catalogFolder[@name='User Interface Profiles']/
catalogFolder[@name='Reporting Profiles']

objectClass

[bibus](#) » [catalogFolder](#)

parent

[User Interface Profiles](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Reporting Profiles object.

<i>Table 347. Policies for the Reporting Profiles object.</i>	
securityObject	permissions
“Library Administrators” on page 1699	X R Sp T W
“Everyone” on page 1698	R X T

What's new

New in Version 10.2.2 – [“Reporting user interface profiles” on page 1825](#)

This object was added.

Configuration

Properties

searchPath

/configuration

objectClassbibus » [configuration](#)**parent**["/](#) on page 1614**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Configuration object.

<i>Table 348. Policies for the Configuration object.</i>	
securityObject	permissions
"Server Administrators" on page 1708	X R Sp T W
"Everyone" on page 1698	X T

11x17

Properties**searchPath**

/configuration/pageDefinition[@name='11x17']

objectClassbibus » [pageDefinition](#)**parent**["Configuration" on page 1673](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the 11x17 object.

<i>Table 349. Policies for the 11x17 object.</i>	
securityObject	permissions
"Server Administrators" on page 1708	X R Sp T W
"Everyone" on page 1698	X R T

What's new**New in Version 10.1.0 – ["Object Documentation Updates" on page 1872](#)**

The documentation now describes the correct search path property value for this object.

A3

Properties**searchPath**

/configuration/pageDefinition[@name='A3']

objectClassbibus » [pageDefinition](#)**parent**[“Configuration” on page 1673](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the A3 object.

<i>Table 350. Policies for the A3 object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

A4**Properties****searchPath**

/configuration/pageDefinition[@name='A4']

objectClassbibus » [pageDefinition](#)**parent**[“Configuration” on page 1673](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the A4 object.

<i>Table 351. Policies for the A4 object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

B4 JIS

Properties

searchPath

/configuration/pageDefinition[@name='B4 JIS']

objectClass

[bibus](#) » [pageDefinition](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the B4 JIS object.

<i>Table 352. Policies for the B4 JIS object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

B5 JIS

Properties

searchPath

/configuration/pageDefinition[@name='B5 JIS']

objectClass

[bibus](#) » [pageDefinition](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the B5 JIS object.

<i>Table 353. Policies for the B5 JIS object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Legal

Properties

searchPath

/configuration/pageDefinition[@name='Legal']

objectClass

[bibus](#) » [pageDefinition](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Legal object.

<i>Table 354. Policies for the Legal object.</i>	
securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Letter

Properties

searchPath

/configuration/pageDefinition[@name='Letter']

objectClass

[bibus](#) » [pageDefinition](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Letter object.

Table 355. Policies for the Letter object.

securityObject	permissions
“Server Administrators” on page 1708	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

User Interface Profiles - deprecated

This object contains all the user interface profiles.

Properties

searchPath

/configuration/uiProfileFolder[@name='User Interface Profiles']

objectClass

[bibus](#) » [uiProfileFolder](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Configuration” on page 1673](#).

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This object is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Reporting Profiles - deprecated

This object contains profiles for report authoring.

Properties

searchPath

/configuration/uiProfileFolder[@name='User Interface Profiles']/
uiProfileFolder[@name='Reporting Profiles']

objectClass

[bibus](#) » [uiProfileFolder](#)

parent

[“User Interface Profiles - deprecated” on page 1678](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Reporting Profiles object.

securityObject	permissions
“Authors” on page 1693	X T
“Express Authors” on page 1698	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This object is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.3 – [“New Security Roles” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Express - deprecated

This [bibus](#) » [uiProfile](#) defines the user interface profile for IBM Cognos Workspace Advanced authoring.

Properties

searchPath

```
/configuration/uiProfileFolder[@name='User Interface Profiles']/  
uiProfileFolder[@name='Reporting Profiles']/uiProfile[@name='Express']
```

objectClass

[bibus](#) » [uiProfile](#)

parent

[“Reporting Profiles - deprecated” on page 1678](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Express object.

<i>Table 357. Policies for the Express object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Express Authors” on page 1698	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This object is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.3 – [“New Security Roles” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Professional - deprecated

This [bibus » uiProfile](#) defines the user interface profile for IBM Cognos Analytics - Reporting authoring.

Properties

searchPath

```
/configuration/uiProfileFolder[@name='User Interface Profiles']/
uiProfileFolder[@name='Reporting Profiles']/uiProfile[@name='Professional']
```

objectClass

[bibus » uiProfile](#)

parent

[“Reporting Profiles - deprecated” on page 1678](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Professional object.

<i>Table 358. Policies for the Professional object.</i>	
securityObject	permissions
“Authors” on page 1693	X T
“Report Administrators” on page 1707	X T
“Directory Administrators” on page 1697	Sp T

What's new

New in Version 10.2.2 – [Deprecation of Reporting profiles](#)

This object is deprecated.

New in Version 8.3 – [“Reporting Profiles” on page 1910](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.3 – [“New Security Roles” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

User Profile

Properties

searchPath

/configuration/account[@name='User Profile']

objectClass

[bibus](#) » [account](#)

parent

[“Configuration” on page 1673](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the User Profile object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Most Recently Used list

Properties

searchPath

/configuration/account[@name='User Profile']/mruFolder[@name='Most Recently Used list']

objectClass

[bibus](#) » [mruFolder](#)

parent

[“User Profile” on page 1681](#)

displaySequence

—

iconURI

—

Policies are acquired from [“User Profile” on page 1681](#).

What's new**New in Version 8.4 — [“Bug Fixes” on page 1897](#)**

This object is now included in the documentation.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

My Folders

Properties**searchPath**

/configuration/account[@name='User Profile']/folder[@name='My Folders']

objectClass

[bibus](#) » [folder](#)

parent

[“User Profile” on page 1681](#)

displaySequence

—

iconURI

icon_myfolder.gif

Policies are acquired from [“User Profile” on page 1681](#).

What's new**New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

My Watch Items

Properties**searchPath**

/configuration/account[@name='User Profile']/subscriptionFolder[@name='My Watch Items']

objectClass

[bibus](#) » [subscriptionFolder](#)

parent

[“User Profile” on page 1681](#)

displaySequence

—

iconURI

icon_mysubscriptions.gif

Policies are acquired from [“User Profile” on page 1681](#).

What's new

New in Version 8.4 – [“Bug Fixes” on page 1897](#)

This object is now included in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Directory

Properties

searchPath

/directory

objectClass

[bibus](#) » [directory](#)

parent

[“/” on page 1614](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Directory object.

<i>Table 360. Policies for the Directory object.</i>	
securityObject	permissions
“Everyone” on page 1698	T

Tenants

This object contains tenant objects in a multi-tenancy environment.

Properties

searchPath

/directory/tenants[@name='Tenants']

objectClass

[bibus](#) » [tenants](#)

parent

[“Directory” on page 1683](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Tenants object.

<i>Table 361. Policies for the Tenants object.</i>	
securityObject	permissions
“Everyone” on page 1698	T

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

The documentation for this object was added.

<Indeterminate>

The name of the namespace is set in IBM Cognos Configuration.

Every user ([bibus » account](#) class) in the same namespace is granted `traverse` permission to the containing namespace.

Properties

searchPath

CAMID("<Indeterminate>")

objectClass

[bibus » namespace](#)

parent

[“Directory” on page 1683](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the <Indeterminate> object.

securityObject	permissions
“Everyone” on page 1698	R
“<Known User>” on page 1684	T

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

<Known User>

Represents a user [bibus » account](#) in an external namespace. This object is not created until the user is authenticated by IBM Cognos Analytics.

Each user who is authenticated with an external security provider has a corresponding account object in Content Manager. The account object contains objects that represent the packages most recently used by the user. The user has full access to these objects, including the ability to set their security policies.

The account object also contains objects that represent the user's sessions. A session object is created automatically when the user logs on to IBM Cognos Analytics. The user can read and write to objects within the session objects, but cannot change their security policies.

The Known User's [“My Folders” on page 1686](#) object does not specify an explicit capability policy. Therefore, capabilities for objects in this folder are based on the object's [owner](#) property.

Properties

searchPath

CAMID("<Indeterminate>")

objectClass

bibus » [account](#)

parent

["<Indeterminate>" on page 1684](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the <Known User> object.

<i>Table 363. Policies for the <Known User> object.</i>	
securityObject	permissions
"Directory Administrators" on page 1697	X R Sp T W
"<Known User>" on page 1684	X R Sp T W

What's new

New in Version 10.1.0 – ["Documentation Updates" on page 1878](#)

This object is now included in the documentation.

<Indeterminate>

A [bibus » session](#) object is created each time a user is authenticated with IBM Cognos Analytics.

Content Manager removes old [bibus » session](#) objects once the user session has been completed.

The name of the session is determined at run time.

Properties

searchPath

CAMID("<Indeterminate>")/session[@name='<Indeterminate>']

objectClass

bibus » [session](#)

parent

["<Known User>" on page 1684](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the <Indeterminate> object.

<i>Table 364. Policies for the <Indeterminate> object.</i>	
securityObject	permissions
"Directory Administrators" on page 1697	X R Sp T W

<i>Table 364. Policies for the <Indeterminate> object. (continued)</i>	
securityObject	permissions
“<Known User>” on page 1684	X R T W

What's new

New in Version 10.1.0 – [“Documentation Updates” on page 1878](#)

This object is now included in the documentation.

Most Recently Used List

Properties

searchPath

CAMID("<Indeterminate>")/mruFolder[@name='Most Recently Used List']

objectClass

[bibus](#) » [mruFolder](#)

parent

[“<Known User>” on page 1684](#)

displaySequence

—

iconURI

—

Policies are acquired from [“<Known User>” on page 1684](#).

What's new

New in Version 10.1.0 – [“Documentation Updates” on page 1878](#)

This object is now included in the documentation.

My Folders

Properties

searchPath

CAMID("<Indeterminate>")/folder[@name='My Folders']

objectClass

[bibus](#) » [folder](#)

parent

[“<Known User>” on page 1684](#)

displaySequence

—

iconURI

—

Policies are acquired from [“<Known User>” on page 1684](#).

What's new

New in Version 10.1.0 – [“Documentation Updates” on page 1878](#)

This object is now included in the documentation.

My Watch Items

Properties

searchPath

CAMID("<Indeterminate>")/subscriptionFolder[@name='My Watch Items']

objectClass

[bibus](#) » [subscriptionFolder](#)

parent

[“<Known User>” on page 1684](#)

displaySequence

—

iconURI

—

Policies are acquired from [“<Known User>” on page 1684](#).

What's new

New in Version 10.1.0 – [“Documentation Updates” on page 1878](#)

This object is now included in the documentation.

Cognos

Properties

searchPath

CAMID(" : ")

objectClass

[bibus](#) » [namespace](#)

parent

[“Directory” on page 1683](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Cognos object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Everyone” on page 1698	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Adaptive Analytics Administrators

Properties

searchPath

CAMID(":Adaptive Analytics Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Adaptive Analytics Administrators object.

<i>Table 366. Policies for the Adaptive Analytics Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Adaptive Analytics Administrators” on page 1688	X T

What's new

New in Version 8.4 — [“Adaptive Analytics Integration” on page 1893](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Adaptive Analytics Users

Properties

searchPath

CAMID(":Adaptive Analytics Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Adaptive Analytics Users object.

<i>Table 367. Policies for the Adaptive Analytics Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W

<i>Table 367. Policies for the Adaptive Analytics Users object. (continued)</i>	
securityObject	permissions
“Adaptive Analytics Users” on page 1688	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 8.4 – [“Adaptive Analytics Integration” on page 1893](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

All Authenticated Users

Represents the set of all authenticated users. This group does not include the [“Anonymous” on page 1690](#) user.

The membership of this group is determined at run time.

Properties

searchPath

CAMID("::All Authenticated Users")

objectClass

[bibus](#) » group

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the All Authenticated Users object.

<i>Table 368. Policies for the All Authenticated Users object.</i>	
securityObject	permissions
“All Authenticated Users” on page 1689	X R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Analysis Users

Represents the set of users who have access to Analysis Studio and default read and execute permissions on Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Analysis Users")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Analysis Users object.

<i>Table 369. Policies for the Analysis Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Analysis Users” on page 1689	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Anonymous

Represents a user account shared by members of the general public who can access IBM Cognos Analytics without being prompted for authentication.

You can disable this account by changing the configuration parameters in IBM Cognos Configuration.

Properties

searchPath

CAMID(" : :Anonymous")

objectClass

[bibus » account](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Anonymous object.

<i>Table 370. Policies for the Anonymous object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W

<i>Table 370. Policies for the Anonymous object. (continued)</i>	
securityObject	permissions
“Anonymous” on page 1690	X R T
“Everyone” on page 1698	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

<Indeterminate>

A [bibus » session](#) object is created each time a user is authenticated with IBM Cognos Analytics. Content Manager removes old [bibus » session](#) objects once the user session has been completed. The name of the session is determined at run time.

Properties

searchPath

CAMID("::Anonymous")/session[@name='<Indeterminate>']

objectClass

[bibus » session](#)

parent

[“Anonymous” on page 1690](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the <Indeterminate> object.

<i>Table 371. Policies for the <Indeterminate> object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Anonymous” on page 1690	X R T W

What's new

New in Version 10.1.0 – [“Documentation Updates” on page 1878](#)

This object is now included in the documentation.

Most Recently Used list

Properties

searchPath

CAMID("::Anonymous")/mruFolder[@name='Most Recently Used list']

objectClass

[bibus » mruFolder](#)

parent

[“Anonymous” on page 1690](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Anonymous” on page 1690](#).

What's new**New in Version 8.4 — “Bug Fixes” on page 1897**

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

My Folders

Properties**searchPath**

CAMID("::Anonymous")/folder[@name='My Folders']

objectClass

[bibus](#) » [folder](#)

parent

[“Anonymous” on page 1690](#)

displaySequence

—

iconURI

icon_myfolder.gif

Policies are acquired from [“Anonymous” on page 1690](#).

What's new**New in Version 10.1.0 — “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

My Watch Items

Properties**searchPath**

CAMID("::Anonymous")/subscriptionFolder[@name='My Watch Items']

objectClass

[bibus](#) » [subscriptionFolder](#)

parent

[“Anonymous” on page 1690](#)

displaySequence

—

iconURI

icon_mysubscriptions.gif

Policies are acquired from [“Anonymous”](#) on page 1690.

What's new

New in Version 8.4 – [“Bug Fixes”](#) on page 1897

This object is now included in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates”](#) on page 1872

The documentation now describes the correct search path property value for this object.

Authors

Represents the set of users who act as report authors. The members of this group can navigate Public Folders, and to read, write and execute objects contained within Public Folders.

Authors have the capabilities [canUseAdvancedDashboardFeatures](#), [canUseAnalysisStudio](#), [canUseCognosViewer](#), [canUseEventStudio](#), [canUseInteractiveDashboardFeatures](#), [canUseQueryStudio](#), [canUseReportStudio](#) and [canUseScheduling](#).

Initial Membership: [“Everyone”](#) on page 1698

Properties

searchPath

CAMID(" :Authors")

objectClass

bibus » [role](#)

parent

[“Cognos”](#) on page 1687

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Authors object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Authors” on page 1693	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates”](#) on page 1872

The documentation now describes the correct search path property value for this object.

Catalog Administrators - removed

Represents the set of users who can manage visualization content. This object has been replaced by the [Library Administrators](#) object.

What's new

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was updated.

New in Version 10.2.1 – [“Visualization support” on page 1831](#)

This object is now created in the content store during upgrade or initialization.

Cognos Insight Users

Represents the set of users who can access Cognos Insight features.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Cognos Insight Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Cognos Insight Users object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Cognos Insight Users” on page 1694	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.2.0 – [“Restricting access to Cognos Insight in IBM Cognos Analytics” on page 1841](#)

This object is now created in the content store during upgrade or initialization.

Consumers

Represents the set of users who act as report consumers. The members of this group can navigate Public Folders, and to read and execute objects contained within Public Folders.

Consumers have the capabilities [canUseCognosViewer](#) and [canUseScheduling](#).

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Consumers")

objectClassbibus » [role](#)**parent**[“Cognos” on page 1687](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Consumers object.

<i>Table 374. Policies for the Consumers object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Consumers” on page 1694	X R T
“All Authenticated Users” on page 1689	R T

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Controller Administrators

Represents the set of users who have access to IBM Cognos Controller administration functions. In addition, Controller Administrators have default read, write, execute, and setPolicy permissions on Public Folders.

Initially, this role has no members.

Properties**searchPath**

CAMID(":Controller Administrators")

objectClassbibus » [role](#)**parent**[“Cognos” on page 1687](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Controller Administrators object.

<i>Table 375. Policies for the Controller Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Controller Administrators” on page 1695	X T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Controller Users

Represents the set of users who can use IBM Cognos Controller. Controller users also have read and execute permissions on Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Controller Users")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Controller Users object.

<i>Table 376. Policies for the Controller Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Controller Users” on page 1696	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Data Manager Authors

Represents the set of users who have access to Data Manager.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Data Manager Authors")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Data Manager Authors object.

<i>Table 377. Policies for the Data Manager Authors object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Data Manager Authors” on page 1696	X R T
“All Authenticated Users” on page 1689	R T

What's new**New in Version 10.1.0 – “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

Directory Administrators

Represents the set of users who administer namespaces and assign capabilities to users.

Directory administrators have full access rights to the default user profile, which is used to construct the user desktop and establish user preferences when a user first logs into the system.

Directory administrators have full access rights to the Cognos namespace and the objects it contains. The Cognos namespace contains all the accounts, groups and roles described in this table. The groups and roles in the Cognos namespace are defined exclusively for use with IBM Cognos Analytics.

Directory administrators have the capabilities [canUseAdministrationPortal](#), [canUseCapabilitiesTool](#) and [canUseDirectoryTool](#).

Initially, this role has no members.

Properties**searchPath**

CAMID(" :Directory Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Directory Administrators object.

<i>Table 378. Policies for the Directory Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Everyone

Represents the list of all authenticated users, as well as the “Anonymous” on page 1690 user, if anonymous access is enabled.

Because most of the initial policies are specified using the Everyone group, there is no need to specify new policies for the “All Authenticated Users” on page 1689 group if you disable Anonymous access.

The membership of this group is determined at run time.

Properties

searchPath

CAMID("::Everyone")

objectClass

[bibus](#) » [group](#)

parent

“Cognos” on page 1687

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Everyone object.

securityObject	permissions
“All Authenticated Users” on page 1689	X R T
“Anonymous” on page 1690	X T

What's new

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Express Authors

Represents the set of users who have access to the Express Reporting profile.

Initial Membership: “Everyone” on page 1698

Properties

searchPath

CAMID(":Express Authors")

objectClass

[bibus](#) » [role](#)

parent

“Cognos” on page 1687

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Express Authors object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Express Authors” on page 1698	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 8.3 — [“New Security Roles” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Library Administrators

Represents the set of users who have access to IBM Cognos Library administration functions. Members can access, import, and administer the contents of the **Library** tab in IBM Cognos Administration.

Initially, this role has no members.

Properties

searchPath

CAMID(":Library Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Controller Administrators object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Library Administrators” on page 1699	X T

What's new

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was added.

Metrics Administrators

Represents the set of users who have access to Metrics Manager administration functions such as creating new metric packages and data integration tasks. In addition, Metrics Administrators have default read, write, execute, and setPolicy permissions on Public Folders.

Initially, this role has no members.

Properties

searchPath

CAMID(" :Metrics Administrators")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Metrics Administrators object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Metrics Administrators” on page 1700	X T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Metrics Authors

Represents the set of users who can create and edit scorecard applications in Metric Studio. Metrics Authors also have default read and execute permissions on Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Metrics Authors")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Metrics Authors object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Metrics Authors” on page 1700	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Metrics Users

Represents the set of users who can monitor performance in Metric Studio. Metrics Users also have read and execute permissions on Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Metrics Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Metrics Users object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Metrics Users” on page 1701	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Mobile Administrators

Reserved.

Properties

searchPath

CAMID(":Mobile Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Mobile Administrators object.

<i>Table 385. Policies for the Mobile Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Mobile Administrators” on page 1702	X T

What's new

New in Version 10.2.1 – [“Administrative changes for Cognos Analytics Mobile Reports” on page 1830](#)

This object is now created in the content store during upgrade or initialization.

Mobile Users

Represents the set of users who can access Mobile features.

Initial Membership: [“Everyone” on page 1698](#) [“Mobile Administrators” on page 1702](#)

Properties

searchPath

CAMID(":Mobile Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Mobile Users object.

<i>Table 386. Policies for the Mobile Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Mobile Users” on page 1702	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.2.0 – [“Restricting access to Cognos Analytics Mobile Reports Service in IBM Cognos Analytics” on page 1842](#)

This object is now created in the content store during upgrade or initialization.

Planning Contributor Users

Reserved.

Properties

searchPath

CAMID(":Planning Contributor Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Planning Contributor Users object.

<i>Table 387. Policies for the Planning Contributor Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Planning Contributor Users” on page 1703	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Planning Rights Administrators

Reserved.

Properties

searchPath

CAMID(":Planning Rights Administrators")

objectClassbibus » [role](#)**parent**[“Cognos” on page 1687](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Planning Rights Administrators object.

<i>Table 388. Policies for the Planning Rights Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Planning Rights Administrators” on page 1703	X T

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Portal Administrators

Represents the set of administrator users who are responsible for the portlets and pagelets in IBM Cognos Connection.

Initially, this role has no members.

Properties**searchPath**

CAMID(" :Portal Administrators")

objectClassbibus » [role](#)**parent**[“Cognos” on page 1687](#)**displaySequence**

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Portal Administrators object.

<i>Table 389. Policies for the Portal Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Portal Administrators” on page 1704	X T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

PowerPlay Administrators

Properties

searchPath

CAMID(":PowerPlay Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the PowerPlay Administrators object.

securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“PowerPlay Administrators” on page 1705	X T

What's new

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

PowerPlay Users

Properties

searchPath

CAMID(":PowerPlay Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the PowerPlay Users object.

<i>Table 391. Policies for the PowerPlay Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“PowerPlay Users” on page 1705	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Query Users

Represents the set of users who have access to Query Studio and default read and execute permissions on Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Query Users")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Query Users object.

<i>Table 392. Policies for the Query Users object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Query Users” on page 1706	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Readers

Represents the set of users who are allowed to read public content. The members of this role can navigate Public Folders and read objects contained within Public Folders.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID(" :Readers")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Readers object.

<i>Table 393. Policies for the Readers object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Readers” on page 1707	X R T
“All Authenticated Users” on page 1689	R T

What's new

New in Version 8.3 — [“New Security Roles” on page 1918](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Report Administrators

Represents the set of users who have access to Reporting and Query Studio. These users also have default read, write, execute and setPolicy permissions on Public Folders.

Initially, this role has no members.

Properties

searchPath

CAMID(" :Report Administrators")

objectClass

[bibus » role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Report Administrators object.

<i>Table 394. Policies for the Report Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Report Administrators” on page 1707	X T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Server Administrators

Represents the set of users who are responsible for administering dispatchers, jobs, and servers.

Initially, this role has no members.

Properties

searchPath

CAMID(":Server Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Server Administrators object.

<i>Table 395. Policies for the Server Administrators object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	X R Sp T W
“Server Administrators” on page 1708	X T

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

System Administrators

Represents the set of users who have full access to modify any object in the content store, regardless of any security policies set for the object.

Note: Any policy specified for any object in the content store is ignored when members of this role manipulate the object in any way or perform an action on its child objects.

Only members of this role can modify its membership. This role cannot be empty. If you do not want to use this role, create an empty group in the Cognos namespace or in your authentication provider, and add the empty group to this role.

Access to this role should be minimized.

Initial Membership: [“Everyone” on page 1698](#)

Properties

searchPath

CAMID("::System Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

access_mobile_service_grant.gif

The following table lists the security objects and permissions that comprise the policies for the System Administrators object.

<i>Table 396. Policies for the System Administrators object.</i>	
securityObject	permissions
“System Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Tenant Administrators

Represents the set of users who are responsible for administering tenants.

Initially, this role has no members.

Properties

searchPath

CAMID("::Tenant Administrators")

objectClass

[bibus](#) » [role](#)

parent

[“Cognos” on page 1687](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Tenant Administrators object.

<i>Table 397. Policies for the Tenant Administrators object.</i>	
securityObject	permissions
“System Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.2.2 – [“Support for delegated tenant administration” on page 1826](#)

This object was added.

Export

Properties

searchPath

/exportDeploymentFolder

objectClass

[bibus](#) » exportDeploymentFolder

parent

[“/” on page 1614](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Export object.

<i>Table 398. Policies for the Export object.</i>	
securityObject	permissions
“Report Administrators” on page 1707	X R Sp T W

Import

Properties

searchPath

/importDeploymentFolder

objectClass

[bibus](#) » importDeploymentFolder

parent

[“/” on page 1614](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Import object.

<i>Table 399. Policies for the Import object.</i>	
securityObject	permissions
“Report Administrators” on page 1707	X R Sp T W

Portal

Properties

searchPath
/portal

objectClass
[bibus » portal](#)

parent
[“/” on page 1614](#)

displaySequence
—

iconURI
—

The following table lists the security objects and permissions that comprise the policies for the Portal object.

<i>Table 400. Policies for the Portal object.</i>	
securityObject	permissions
“Portal Administrators” on page 1704	X R Sp T W
“Everyone” on page 1698	X R T

Administration

This [bibus » portalPackage](#) object stores administration console objects.

Properties

searchPath
/portal/portalPackage[@name='Administration']

objectClass
[bibus » portalPackage](#)

parent
[“Portal” on page 1711](#)

displaySequence
—

iconURI
—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Pagelets

This object is the main pagelet folder ([bibus](#) » [pageletFolder](#)).

Properties

searchPath

`/portal/portalPackage[@name='Administration']/
pageletFolder[@name='Pagelets']`

objectClass

[bibus](#) » [pageletFolder](#)

parent

[“Administration” on page 1711](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Console

This pagelet folder ([bibus](#) » [pageletFolder](#)) contains administration console pagelets.

Properties

searchPath

`/portal/portalPackage[@name='Administration']/
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']`

objectClass

[bibus](#) » [pageletFolder](#)

parent

[“Pagelets” on page 1712](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Configuration

This pagelet folder ([bibus » pageletFolder](#)) contains pagelets related to system configuration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']
```

objectClass

[bibus » pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

2

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Content Administration

This [bibus » pagelet](#) renders administrative tasks such as import deployments ([bibus » importDeployment](#)) and content maintenance tasks.

The user must have the [canUseContentStoreTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Content Administration']
```

objectClass

[bibus » pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

6

iconURI

[icon_content_maintenance.gif](#)

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 – “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Data Source Connections

This [bibus](#) » [pagelet](#) renders data source information.

The user must have the [canUseDataSourcesTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Data Source  
Connections']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

8

iconURI

[icon_data_source.gif](#)

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Dispatchers and Services

This [bibus](#) » [pagelet](#) renders dispatchers and services information.

The user must have the [canUseServerAdministrationTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Dispatchers  
and Services']
```

objectClass

[bibus » pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

1

iconURI

icon_dispatcher_and_service.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Distribution Lists and Contacts

This [bibus » pagelet](#) renders distribution lists and contacts.

The user must have the [canUseDistributionListsAndContactsTool](#) capability to access this pagelet.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Distribution  
Lists and Contacts']
```

objectClass

[bibus » pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

5

iconURI

icon_distribution_list_and_contact.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Portlets

This [bibus » pagelet](#) renders portlet imports ([“Portlets” on page 1735](#)).

The user must have the [canUsePortalAdministrationTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Portlets']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

2

iconURI

icon_portlet_group.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Printers

This [bibus](#) » [pagelet](#) renders available system printers.

The user must have the [canUsePrintersTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Printers']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

4

iconURI

icon_printer.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Styles

This [bibus » pagelet](#) renders available system styles.

The user must have the [canUsePortalAdministrationTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Configuration']/pagelet[@name='Styles']
```

objectClass

[bibus » pagelet](#)

parent

[“Configuration” on page 1713](#)

displaySequence

3

iconURI

icon_style_portal.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Index Search

This pagelet folder ([bibus » pageletFolder](#)) contains pagelets related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Index Search']
```

objectClass

[bibus » pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Index

This pagelet ([bibus](#) » [pagelet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Index Search']/pagelet[@name='Index']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Index Search” on page 1717](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Search

This pagelet ([bibus](#) » [pagelet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Index Search']/pagelet[@name='Search']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Index Search” on page 1717](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Storage

This pagelet ([bibus](#) » [pagelet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Index Search']/pagelet[@name='Storage']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Index Search” on page 1717](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Multitenancy

This pagelet folder ([bibus](#) » [pageletFolder](#)) contains pagelets related to multi-tenancy.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Multitenancy']
```

objectClass

[bibus](#) » [pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

2

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

The documentation for this object was added.

Tenants

This pagelet ([bibus](#) » [pagelet](#)) is related to multi-tenancy in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/pageletFolder[@name='Pagelets']/  
pageletFolder[@name='Console']/pageletFolder[@name='Multitenancy']/pagelet[@name='Tenants']
```

objectClass

[bibus](#) » [portlet](#)

parent

[“Multitenancy” on page 1719](#)

displaySequence

1

iconURI

icon_tenants.gif

The following table lists the security objects and permissions that comprise the policies for the Tenants pagelet object.

<i>Table 401. Policies for the Tenants pagelet object.</i>	
securityObject	permissions
“System Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

The documentation for this object was added.

Library

This pagelet folder ([bibus](#) » [pageletFolder](#)) contains pagelets related to libraries.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Library']
```

objectClass

[bibus](#) » [pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

30

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

The documentation for this object was added.

User Interface Profiles

This pagelet ([bibus](#) » [pagelet](#)) renders user interface profiles.

This object is created in the Content Store when Reporting user interface profiles have been enabled.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Library']//pagelet[@name='User Interface Profiles']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Library” on page 1720](#)

displaySequence

1

iconURI

icon_user_interface_profiles.gif

The following table lists the security objects and permissions that comprise the policies for the User Interface Profiles object.

<i>Table 402. Policies for the User Interface Profiles object.</i>	
securityObject	permissions
“System Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.2.2 – “Documentation Updates” on page 1827

This object was added.

Visualizations

This pagelet ([bibus](#) » [pagelet](#)) renders visualizations.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Library']//pagelet[@name='Visualizations']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Library” on page 1720](#)

displaySequence

0

iconURI

icon_visualizations.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)**

The documentation for this object was added.

PowerPlay

This is the [bibus » pagelet](#) for the PowerPlay 8 configuration tool. It exposes functionality equivalent to the IBM Cognos Series 7 PowerPlay administration tool, without the monitoring features.

The user must have the [canUsePowerPlayAdministration](#) capability to access this pagelet.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pagelet[@name='PowerPlay']
```

objectClass

[bibus » pagelet](#)

parent

[“Console” on page 1712](#)

displaySequence

1

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.4 – [“PowerPlay 8 Integration” on page 1886](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – [“Object Updates” on page 1876](#)

This object was renamed.

Security

This pagelet folder ([bibus » pageletFolder](#)) contains pagelets related to system security.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Security']
```

objectClass

[bibus » pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

3

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Capabilities

This [bibus](#) » [pagelet](#) renders available capabilities.

The user must have the [canUseCapabilitiesTool](#) capability to access this pagelet.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Security']/pagelet[@name='Capabilities']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Security” on page 1722](#)

displaySequence

3

iconURI

icon_capabilities.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

User Interface Profiles

This object is the [bibus](#) » [pagelet](#) for the user interface profiles task.

The user must have the [canUseCapabilitiesTool](#) capability to access this pagelet.

This object is deleted from the Content Store if Reporting user interface profiles have been enabled.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Security']/pagelet[@name='User Interface Profiles']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Security” on page 1722](#)

displaySequence

1

iconURI

icon_user_interface_profiles.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 – [“Reporting user interface profiles” on page 1825](#)

The documentation for this object was updated.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 8.3 – [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

Users, Groups, and Roles

This [bibus](#) » [pagelet](#) renders directory access.

The user must have the [canUseUsersGroupsAndRolesTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Security']/pagelet[@name='Users, Groups, and Roles']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Security” on page 1722](#)

displaySequence

4

iconURI

icon_portlet_group.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Status

This pagelet folder ([bibus » pageletFolder](#)) contains pagelets related to the system status.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']
```

objectClass

[bibus » pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

4

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Current Activities

This [bibus » pagelet](#) renders current system activities.

The user must have the [canUseMonitorActivityTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Current Activities']
```

objectClass

[bibus » pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

5

iconURI

icon_current_run_activities.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Data Sets

This [bibus](#) » [pagelet](#) renders data sets.

The user must have the [canUseMyDataSets](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Data Sets']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

2

iconURI

icon_datasets.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 – [My data sets](#)

This object was added.

Past Activities

This [bibus](#) » [pagelet](#) renders past system activities.

The user must have the [canUseMonitorActivityTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Past Activities']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

4

iconURI

icon_completed_run_activities.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Schedules

This [bibus » pagelet](#) renders schedules.

The user must have the [canUseMonitorActivityTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Schedules']
```

objectClass

[bibus » pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

1

iconURI

icon_schedule.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

System

This object is the [bibus » pagelet](#) for the system task.

The user must have the [canUseServerAdministrationTool](#) capability to access this pagelet.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='System']
```

objectClass

[bibus » pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

2

iconURI

icon_system.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 – [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Upcoming Activities

This [bibus](#) » [pagelet](#) renders future system activities.

The user must have the [canUseMonitorActivityTool](#) capability to access this pagelet.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Upcoming Activities']
```

objectClass

[bibus](#) » [pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

3

iconURI

icon_future_run_activities.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 – [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Portlets

This object is the main portlet folder ([bibus](#) » [portletFolder](#)).

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']
```

objectClass

[bibus](#) » [portletFolder](#)

parent

[“Administration” on page 1711](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — “Administration Console” on page 1906**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Console

This portlet folder contains [bibus](#) » [portlet](#) producers.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']
```

objectClass

[bibus](#) » [portletFolder](#)

parent

[“Portlets” on page 1728](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — “Administration Console” on page 1906**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

cogadmin

This object is the portlet producer for the administration console.

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']
```

objectClass

[bibus » portletProducer](#)

parent

[“Console” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Capabilities

This object is the [bibus » portlet](#) used by the [“Capabilities” on page 1723](#) [bibus » pagelet](#).

Properties**searchPath**

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Capabilities']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 8.3 — [“Administration Console” on page 1906](#)**

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Content Administration

This object is the [bibus » portlet](#) used by the [“Content Administration” on page 1713](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Content Administration']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 — [“Bug Fixes” on page 1897](#)

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Current Activities

This object is the [bibus » portlet](#) used by the [“Current Activities” on page 1725](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Current Activities']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Data Sets

This object is the [bibus » portlet](#) used by the [bibus » pagelet](#) » [“Data Sets” on page 1726](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Data Sets']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 — [My data sets](#)

This object was added.

Dynamic Cubes

This [bibus » pagelet](#) renders dynamic cubes.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Status']/pagelet[@name='Dynamic Cubes']
```

objectClass

[bibus » pagelet](#)

parent

[“Status” on page 1725](#)

displaySequence

30

iconURI

icon_datastores.gif

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 — [Data Stores renamed to Dynamic Cubes in Cognos Administration](#)

This object was added.

Data Source Connections

This object is the [bibus » portlet](#) used by the [“Data Source Connections” on page 1714](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Data Source Connections']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Dispatchers and Services

This object is the [bibus » portlet](#) used by the [“Dispatchers and Services” on page 1714](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Dispatchers and Services']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Distribution Lists and Contacts

This object is the [bibus » portlet](#) used by the [“Distribution Lists and Contacts” on page 1715](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Distribution  
Lists and Contacts']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Dynamic Cubes

This object is the [bibus » portlet](#) used by the [bibus » pagelet » Dynamic cubes](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/portletFolder[@name='Portlets']/  
portletFolder[@name='Console']/portletProducer[@name='cogadmin']/portlet[@name='Dynamic  
Cubes']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 — [Data Stores renamed to Dynamic Cubes in Cognos Administration](#)

This object was added.

Past Activities

This object is the [bibus » portlet](#) used by the [“Past Activities” on page 1726](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Past Activities']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Portlets

This object is the [bibus » portlet](#) used by the [“Portlets” on page 1715](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Portlets']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 – “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

PowerPlay

This object is the [bibus » portlet](#) used by the [“PowerPlay” on page 1722](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='PowerPlay']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.4 – “PowerPlay 8 Integration” on page 1886

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – “Object Updates” on page 1876

This object was renamed.

Printers

This object is the [bibus » portlet](#) used by the [“Printers” on page 1716](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Printers']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Profiles Administration

This object is the [bibus » portlet](#) used by the [“User Interface Profiles” on page 1723](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Profiles Administration']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Schedules

This object is the [bibus » portlet](#) used by the [“Schedules” on page 1727](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Schedules']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Styles

This object is the [bibus » portlet](#) used by the [“Styles” on page 1717](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Styles']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

System

This object is the [bibus » portlet](#) used by the [“System” on page 1727](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='System']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 — “Bug Fixes” on page 1897

The name of this object has been corrected in the documentation.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Upcoming Activities

This object is the [bibus » portlet](#) used by the [“Upcoming Activities” on page 1728](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Upcoming Activities']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — “Administration Console” on page 1906

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

Users Groups and Roles

This object is the [bibus » portlet](#) used by the [“Users, Groups, and Roles” on page 1724](#) [bibus » pagelet](#).

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletFolder[@name='Console']/  
portletProducer[@name='cogadmin']/portlet[@name='Users Groups and Roles']
```

objectClass

[bibus » portlet](#)

parent

[“cogadmin” on page 1729](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Administration Console” on page 1906](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

IBM Cognos Go! Search

This object is the portlet producer for portlets that are used for enhanced search in the search results page in IBM Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']
```

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1728](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Enhanced Search - command panel

This portlet ([bibus » portlet](#)) is related to enhanced search in the search results page in IBM Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']/portlet[@name='IBM Cognos Enhanced Search - command panel']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Go! Search” on page 1740](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Enhanced Search - External Search Results

This portlet ([bibus » portlet](#)) is related to enhanced search in the search results page within Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']/portlet[@name='IBM Cognos  
Enhanced Search - External Search Results']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Go! Search” on page 1740](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Enhanced Search - main UI

This portlet ([bibus » portlet](#)) is related to enhanced search in the search results page in IBM Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']/portlet[@name='IBM Cognos Enhanced Search - main UI']
```

objectClass

[bibus](#) » [portlet](#)

parent

[“IBM Cognos Go! Search” on page 1740](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Enhanced Search - Refinement Viewer

This portlet ([bibus](#) » [portlet](#)) is related to enhanced search in the search results page in IBM Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']/portlet[@name='IBM Cognos Enhanced Search - Refinement Viewer']
```

objectClass

[bibus](#) » [portlet](#)

parent

[“IBM Cognos Go! Search” on page 1740](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Enhanced Search - Results Viewer

This portlet ([bibus](#) » [portlet](#)) is related to enhanced search in the search results page in IBM Cognos Connection.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search']/portlet[@name='IBM Cognos Enhanced Search - Results Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Go! Search” on page 1740](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

IBM Cognos Go! Search Admin

This object is the portlet producer for portlets that are used for enhanced search in the search results page in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search Admin']
```

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1728](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Index

This portlet ([bibus » portlet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search Admin']/portlet[@name='Index']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Go! Search Admin” on page 1743](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Search

This portlet ([bibus » portlet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search Admin']/portlet[@name='Search']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Go! Search Admin” on page 1743](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Storage

This portlet ([bibus » portlet](#)) is related to enhanced search in IBM Cognos Administration.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos  
Go! Search Admin']/portlet[@name='Storage']
```

objectClass

[bibus](#) » [portlet](#)

parent

[“IBM Cognos Go! Search Admin” on page 1743](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

This object is now included in the documentation.

Multitenancy

This pagelet folder ([bibus](#) » [pageletFolder](#)) contains pagelets related to multi-tenancy.

Properties

searchPath

```
/portal/portalPackage[@name='Administration']/  
pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/  
pageletFolder[@name='Multitenancy']
```

objectClass

[bibus](#) » [pageletFolder](#)

parent

[“Console” on page 1712](#)

displaySequence

2

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.2.2 — [“Documentation Updates” on page 1827](#)

The documentation for this object was added.

Tenants

This pagelet ([bibus](#) » [pagelet](#)) is related to multi-tenancy in IBM Cognos Administration.

Properties

searchPath

/portal/portalPackage[@name='Administration']/pageletFolder[@name='Pagelets']/pageletFolder[@name='Console']/pageletFolder[@name='Multitenancy']/pagelet[@name='Tenants']

objectClass

bibus » portlet

parent

[“Multitenancy” on page 1719](#)

displaySequence

1

iconURI

icon_tenants.gif

The following table lists the security objects and permissions that comprise the policies for the Tenants pagelet object.

<i>Table 403. Policies for the Tenants pagelet object.</i>	
securityObject	permissions
“System Administrators” on page 1708	X R Sp T W

What's new

New in Version 10.2.2 – [“Documentation Updates” on page 1827](#)

The documentation for this object was added.

Connection

Properties

searchPath

/portal/portalPackage[@name='Connection']

objectClass

bibus » portalPackage

parent

[“Portal” on page 1711](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Pages

Properties

searchPath

/portal/portalPackage[@name='Connection']/pageletFolder[@name='Pages']

objectClass

bibus » [pageletFolder](#)

parent

[“Connection” on page 1746](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Portlets

Properties**searchPath**

/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']

objectClass

bibus » [portletFolder](#)

parent

[“Connection” on page 1746](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Dashboard

This object is the portlet producer for dashboard portlets.

Properties**searchPath**

/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/
portletProducer[@name='Dashboard']

objectClass

bibus » [portletProducer](#)

parent

[“Portlets” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Dashboard Fragments” on page 1914](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Multi-page

This object is the [bibus » portlet](#) for tabbed pages. Use it to create a dashboard with multiple pages.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='Dashboard']/portlet[@name='Multi-page']
```

objectClass

[bibus » portlet](#)

parent

[“Dashboard” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Dashboard Fragments” on page 1914](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

IBM Cognos Content

This object is the portlet producer for IBM Cognos Content portlets.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Content']
```

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 — [“Object Updates” on page 1876](#)

This object was renamed.

IBM Cognos Navigator

This object is the [bibus » portlet](#) used to navigate the folders that contain IBM Cognos content and applications.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Content']/portlet[@name='IBM Cognos  
Navigator']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Content” on page 1748](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 — [“Object Updates” on page 1876](#)

This object was renamed.

IBM Cognos Search

This object is the [bibus » portlet](#) used to search folders for IBM Cognos content and applications.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Content']/portlet[@name='IBM Cognos  
Search']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Content” on page 1748](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 — “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 — “Object Updates” on page 1876

This object was renamed.

IBM Cognos Viewer

This object is the [bibus » portlet](#) used to view IBM Cognos content and applications.

Properties**searchPath**

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Content']/portlet[@name='IBM Cognos  
Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Content” on page 1748](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 — “Object Documentation Updates” on page 1872**

The documentation now describes the correct search path property value for this object.

IBM Cognos Extended Applications

This object is the portlet producer for the IBM Cognos Extended Applications portlet.

Properties**searchPath**

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Extended Applications']
```

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – [“Object Updates” on page 1876](#)

This object was renamed.

IBM Cognos Extended Applications Portlet

This portlet provides the link between custom portlets you develop and IBM Cognos Connection. The IBM Cognos Software Development Kit must be installed to run this portlet.

For more information about developing custom portlets and using the IBM Cognos Extended Applications toolkit, see [Creating extended applications using IBM Cognos Portal Services](#).

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Extended Applications']/portlet[@name='IBM  
Cognos Extended Applications Portlet']
```

objectClass

[bibus](#) » portlet

parent

[“IBM Cognos Extended Applications” on page 1750](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – [“Object Updates” on page 1876](#)

This object was renamed.

IBM Cognos Metric Studio

This object is the portlet producer for IBM Cognos Metric Studio portlets.

This object is not created during content store initialization. It is created during the first startup after the installation of IBM Cognos Metrics Manager.

Properties

searchPath

/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos Metric Studio']

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Dashboard Fragments” on page 1914](#)

This object is now included in the documentation.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 — [“Object Updates” on page 1876](#)

This object was renamed.

IBM Cognos History Chart

This object is the [bibus » portlet](#) for a chart that shows the historical data for a particular metric, as defined in Metric Studio.

This object is not created during content store initialization. It is created during the first startup after the installation of IBM Cognos Metrics Manager.

Properties

searchPath

/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/portletProducer[@name='IBM Cognos Metric Studio']/portlet[@name='IBM Cognos History Chart']

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Metric Studio” on page 1751](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Dashboard Fragments” on page 1914](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – “Object Updates” on page 1876

This object was renamed.

IBM Cognos Metric List

This object is the [bibus » portlet](#) used to show a list of metrics. These metrics are defined in Metric Studio.

This object is not created during content store initialization. It is created during the first startup after the installation of IBM Cognos Metrics Manager.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Metric Studio']/portlet[@name='IBM Cognos  
Metric List']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Metric Studio” on page 1751](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 – “Dashboard Fragments” on page 1914

This object is now included in the documentation.

This object has been renamed.

New in Version 10.1.0 – “Object Documentation Updates” on page 1872

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – “Object Updates” on page 1876

This object was renamed.

IBM Cognos Utility

This object is the portlet producer for portlets that are used for HTML pages and other Web content.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']
```

objectClass

[bibus » portletProducer](#)

parent

[“Portlets” on page 1747](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

New in Version 10.1.0 – [“Object Updates” on page 1876](#)

This object was renamed.

Bookmarks Viewer

This object is the [bibus » portlet](#) for entering and showing a user's favorite URLs.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']/portlet[@name='Bookmarks  
Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Utility” on page 1753](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

HTML Source

This object is the [bibus » portlet](#) for entering and displaying HTML code. Use it to show custom HTML content.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']/portlet[@name='HTML Source']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Utility” on page 1753](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.3 — [“Dashboard Fragments” on page 1914](#)

This object is now created in the content store during upgrade or initialization.

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

HTML Viewer

This object is the [bibus » portlet](#) used to display HTML content from different sources.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']/portlet[@name='HTML Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Utility” on page 1753](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Image Viewer

This object is the [bibus » portlet](#) that shows images.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']/portlet[@name='Image Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Utility” on page 1753](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

RSS Viewer

This object is the [bibus » portlet](#) used to display Really Simple Syndication (RSS) news feeds.

Properties

searchPath

```
/portal/portalPackage[@name='Connection']/portletFolder[@name='Portlets']/  
portletProducer[@name='IBM Cognos Utility']/portlet[@name='RSS Viewer']
```

objectClass

[bibus » portlet](#)

parent

[“IBM Cognos Utility” on page 1753](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Styles

This object contains objects that define portal skins.

Properties

searchPath

```
/portal/portalSkinFolder[@name='Styles']
```

objectClass

[bibus » portalSkinFolder](#)

parent

[“Portal” on page 1711](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Business

Properties

searchPath

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Business']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 8.4 – [“Bug Fixes” on page 1897](#)

This object is now included in the documentation.

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Classic

Properties

searchPath

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Classic']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Contemporary

Properties

searchPath

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Contemporary']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Corporate

Properties

searchPath

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Corporate']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new

New in Version 10.1.0 — [“Object Documentation Updates” on page 1872](#)

The documentation now describes the correct search path property value for this object.

Modern

Properties

searchPath

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Modern']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Presentation

Properties**searchPath**

/portal/portalSkinFolder[@name='Styles']/portalSkin[@name='Presentation']

objectClass

[bibus](#) » [portalSkin](#)

parent

[“Styles” on page 1756](#)

displaySequence

—

iconURI

—

Policies are acquired from [“Portal” on page 1711](#).

What's new**New in Version 10.1.0 – [“Object Documentation Updates” on page 1872](#)**

The documentation now describes the correct search path property value for this object.

Public Folders

Properties**searchPath**

/content

objectClass

[bibus](#) » [content](#)

parent

[“/” on page 1614](#)

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Public Folders object.

<i>Table 404. Policies for the Public Folders object.</i>	
securityObject	permissions
“Controller Administrators” on page 1695	X R Sp T W
“Metrics Administrators” on page 1700	X R Sp T W
“Planning Rights Administrators” on page 1703	X R Sp T W
“PowerPlay Administrators” on page 1705	X R Sp T W
“Report Administrators” on page 1707	X R Sp T W
“Analysis Users” on page 1689	X R T
“Consumers” on page 1694	X R T
“Controller Users” on page 1696	X R T
“Data Manager Authors” on page 1696	X R T
“Metrics Authors” on page 1700	X R T
“Metrics Users” on page 1701	X R T
“Planning Contributor Users” on page 1703	X R T
“Query Users” on page 1706	X R T
“PowerPlay Users” on page 1705	X R T
“Authors” on page 1693	X R T W
“Express Authors” on page 1698	X R T W
“Readers” on page 1707	R T

The following table lists the security objects and permissions that comprise the user capability policies for the Public Folders object.

<i>Table 405. User capability policies for the Public Folders object.</i>	
securityObject	permissions
canUseEV	“Everyone” on page 1698
canOpenPowerPlayInAnalysisStudio	“Everyone” on page 1698
canOpenPowerPlayInReportStudio	“Everyone” on page 1698
canUseAdaptiveAnalytics	“Everyone” on page 1698
canUseAdaptiveAnalyticsAdministration	“Everyone” on page 1698
canUseAnalysisStudio	“Everyone” on page 1698
canUseBursting	“Everyone” on page 1698
canUseConditionalSubscriptions	“Everyone” on page 1698

Table 405. User capability policies for the Public Folders object. (continued)

securityObject	permissions
canUseEventStudio	"Everyone" on page 1698
canUseHTML	"Everyone" on page 1698
canUseGlossary	"Everyone" on page 1698
canUseLineage	"Everyone" on page 1698
canUseMetricsManagerAdministration	"Everyone" on page 1698
canUseMetricStudio	"Everyone" on page 1698
canUseMetricStudioEditView	"Everyone" on page 1698
canUsePackageDataSources	"Everyone" on page 1698
canUsePlanningAdministration	"Everyone" on page 1698
canUsePlanningContributor	"Everyone" on page 1698
canUsePowerPlay	"Everyone" on page 1698
canUseQueryStudio	"Everyone" on page 1698
canUseQueryStudioAdvancedMode	"Everyone" on page 1698
canUseReportStudio	"Everyone" on page 1698
canUseSpecifications	"Everyone" on page 1698
canUseUserDefinedSQL	"Everyone" on page 1698

What's new

New in Version 9.0.0 – [“Support for IBM Cognos Express” on page 1879](#)

The capability policy of this object was changed to grant [“Everyone” on page 1698](#) the [canUseEV](#) capability.

New in Version 8.3 – [“New Security Roles” on page 1918](#)

The policy of this object was changed to include the roles [“Express Authors” on page 1698](#) and [“Readers” on page 1707](#).

New in Version 8.4 – [“Object Capabilities” on page 1885](#)

This object is now created in the content store during upgrade or initialization.

New in Version 8.4 — “PowerPlay 8 Integration” on page 1886

The policy of this object was changed to include the PowerPlay Administrators role and the PowerPlay Users role.

Transient

This object contains objects that hold transient data required by the [agentService](#).

Properties

searchPath

/transientStateFolder

objectClass

bibus » [transientStateFolder](#)

parent

“/” on page 1614

displaySequence

—

iconURI

—

The following table lists the security objects and permissions that comprise the policies for the Transient object.

<i>Table 406. Policies for the Transient object.</i>	
securityObject	permissions
“Everyone” on page 1698	T

Job

This object is used to determine which users have the [canUseJob](#) capability.

Properties

searchPath

/capability/productCapability[@name='Job']

objectClass

bibus » [productCapability](#)

parent

[Capability](#)

displaySequence

0

iconURI

icon_job.gif

The following table lists the security objects and permissions that comprise the policies for the Job object.

<i>Table 407. Policies for the Job object.</i>	
securityObject	permissions
Analytics Explorers	X T
Analytics Users	X T

<i>Table 407. Policies for the Job object. (continued)</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

Notebook

This object is used to determine which users have the [canUseNotebook](#) capability.

Properties

searchPath

[/capability/productCapability\[@name='Notebook'\]](#)

objectClass

[bibus](#) » [productCapability](#)

parent

[Capability](#)

displaySequence

0

iconURI

–

The following table lists the security objects and permissions that comprise the policies for the Notebook object.

<i>Table 408. Policies for the Notebook object.</i>	
securityObject	permissions
“Directory Administrators” on page 1697	Sp T

Chapter 35. Metadata schema reference

For validation purposes, two metadata schema files are included with IBM Cognos Software Development Kit: MAQuery.xsd and MAResponse.xsd. They reference the elements, attributes, and data types used in requests and responses involving the published IBM Cognos Framework Manager model.

All IBM Cognos interfaces that feature a view of this published model represent the model functions and metadata in the form of a metadata tree. We recommend that you review this reference and its associated XSD files before you attempt to adapt the hierarchical display format.

Metadata requests and responses may be valid for these schemas, yet still fail due to validity problems with the data itself. For more information about resolving model-related problems, see the *IBM Cognos Framework Manager Developer Guide* and the Release Notes delivered with your version of the product.

MAQuery.xsd and MAResponse.xsd are located in *installation_location/sdk*.

MAQuery Elements

The MAQuery.xsd file defines the elements, attributes, and data types used in requests related to the published IBM Cognos Framework Manager model, or metadata tree.

The MAQuery schema shows you how to build a specification string for the `reportServiceMetadataSpecification` class to use with the `runSpecification(specification, parameterValues, options)` method.

Before you review the alphabetical list of elements that make up this query, you must understand its overall structure. There is one top-level [“request” on page 1772](#) element, which contains [“Functions” on page 1768](#), [“Metadata” on page 1768](#), and [“Actions” on page 1765](#) elements.

Note: The top-level element `metadataRequest` is not shown in the schema file because it is not part of the request that is forwarded to the metadata service. It is required as part of the request to the report service, and must be included in each instance of class `reportServiceMetadataSpecification`.

The query engine will navigate through the request until it reaches this instance, and process it. Multiple piggy-backed requests are not supported.

Each of these elements typically contains other elements, which are used to narrow the scope of the request.

If your request includes a property, constraints, or additional metadata attributes such as `authoringLocale` and `depth`, these become required response attributes.

For more information about the properties of IBM Cognos Framework Manager models, see the model schema reference topic in the *IBM Cognos Framework Manager Developer Guide*.

Actions

Used when requesting actions-related information. IBM Cognos Analytics supports only URL actions. The actions query is used to retrieve URL links for supported datasources.

Attributes

authoringLocale

Specifies the locale that determines the rendering language for reports and the metadata tree.

Usage: required Type: language

Content model

[Properties](#) then [Constraints](#)

Parent elements

[request](#)

BlockConstraint

Contains sets of constraints to be applied when requesting function-related information.

Attributes

Name

Specifies the name of the object.

Usage: optional Type: string

Operation

Specifies how the constraints are processed.

Usage: optional Default: AND Type: string

Type

Usage: optional Default: Block Type: string.

Attribute values are restricted to the following.

Block

Set

Content model

([Constraint](#) or [BlockConstraint](#)) (*one or more*)

Parent elements

[BlockConstraint](#) , [Constraints](#)

Constraint

Specifies one or more constraints to be applied when requesting function-related information.

Attributes

Condition

Specifies the nature of the constraint used to narrow the scope of the request.

Usage: required Type: string

Content model

Empty element.

Parent elements

[Constraints](#)

Constraint

Specifies one or more constraints to be applied when requesting metadata-related information.

Attributes

CaseSensitive

Specifies whether the Condition is case-sensitive.

Usage: optional Default: true Type: boolean

Condition

Specifies the nature of the constraint used to narrow the scope of the request.

Usage: required Type: string.

Pattern: contains* Pattern: starts-with* Pattern: ends-with* Pattern: = Pattern: !=

Content model

Empty element.

Parent elements

[BlockConstraint](#) , [Constraints](#)

Constraints

Specifies a collection of constraints for function-related information.

Content model

([Constraint](#)) (one or more)

Parent elements

[Actions](#) , [Functions](#)

Constraints

Specifies a collection of constraints for metadata-related information.

Content model

([Constraint](#) or [BlockConstraint](#)) (one or more)

Parent elements

[Metadata](#)

Functions

Used to retrieve information on functions provided in IBM Cognos Analytics and model-specific functions, such as those imported from a database.

Attributes

Depth

Specifies the depth of the sub-tree returned by the function request.

Usage: optional Default: 0 Type: integer

authoringLocale

Specifies the locale that determines the rendering language for reports and the metadata tree.

Usage: optional Type: string

start_at

Specifies the point in the model where the request is started, such as [NamespaceName] . [QuerySubjectName] . [QueryItemName]. This attribute is deprecated. Use start_atPath instead.

Usage: required Type: string

Content model

[Properties](#) then [Constraints](#) (*optional*)

Parent elements

[request](#)

Metadata

Specifies complex content related to model metadata-type requests.

Attributes

Depth

Specifies the depth of the sub-tree returned by the function request.

Usage: optional Default: 0 Type: nonNegativeInteger

OutputXML

Usage: optional Default: true Type: boolean

ResultCount

Limits the number of returned entities for chunk retrieval.

Usage: optional Type: nonNegativeInteger

ResultStart

Specifies the start of the entities to be returned for chunk retrieval.

Usage: optional Type: nonNegativeInteger

_enumLabels

Set to true to reference the requested properties by label name rather than number.

Usage: optional Default: false Type: boolean

authoringLocale

Specifies the locale that determines the rendering language for reports and the metadata tree.

Usage: required Type: language

no_collections

Serves as a flag to switch to no collection in the output.

Usage: optional Default: 1 Type: nonNegativeInteger

start_at

Specifies the point in the model where the request is started, such as [NamespaceName] . [QuerySubjectName] . [QueryItemName]. This attribute is deprecated. Use start_atPath instead.

Usage: optional Type: string

start_atPath

Specifies the point in the model where the request is started, such as [NamespaceName] . [QuerySubjectName] . [QueryItemName]. start_atPath or start_atRef must be specified.

Usage: optional Type: string

start_atRef

Specifies a reference point in the model where the request is started, such as [Namespace Folder] . [Query Subject] . [Query Item] or [Namespace Folder] . [ShortcutToQS] . [Query Item]. start_atPath or start_atRef must be specified.

Usage: optional Type: string

Content model

Start_at (*optional*) then Properties then Constraints (*optional*)

Parent elements

request

Path

Specifies a search path.

Content model

Content type is string.

Parent elements

Start_at

Properties

Specifies the properties to filter on when requesting function-related or metadata-type information.

Content model

(Property) (*one or more*)

Parent elements

[Functions](#) , [Metadata](#)

Properties

Specifies the properties to filter on when requesting action-related information.

Content model

([Property](#)) *(one or more)*

Parent elements

[Actions](#)

Property

Specifies one or more properties to filter on, when requesting function-related or metadata-type information.

If you know the name and context of an item, you can specify it after the @ symbol, with or without a leading dot (.) to indicate the path from the root. Or, for a broader search that does not specify a particular set of properties, you can request elements using the asterisk wildcard. For example, requesting `queryItem/@*` retrieves all query items and all of their properties.

You can specify any of the following attributes to narrow the scope of your request:

- */@_path
- */@_ref
- */@name
- */@screenTip
- */@dimType
- */@caption
- ./calculation
- ./filter
- ./folder
- ./queryItem
- ./queryItemFolder
- ./querySubject
- ./member
- */@memberUniqueName
- */@memberCaption
- */@levelUniqueName
- */@levelNumber
- */@levelLabel
- */@parentUniqueName
- */@dimensionUniqueName
- */@hierarchyUniqueName
- ./dimension

- ./hierarchy
- ./levelRfe
- ./level
- */@datatype
- */@displayType
- */@promptCascadeOn
- */@promptDisplayItems
- */@promptFilterItems
- */@promptType
- */@regularAggregate
- */@usage
- */@sortOnRef
- */@promptCascadeOnRef
- */@decimalSeparator
- */@listSeparator
- */@modelSearchPath

Attributes

Name

Specifies the name of the object.

Usage: required Type: string

Content model

Empty element.

Parent elements

[Properties](#)

Property

Specifies the list of action properties.

The allowed values are

- Name
- Description
- CommandType
- Command
- CoordinateType
- CoordinateValue

Attributes

Name

Specifies the name of the object.

Usage: required Type: string

Content model

Empty element.

Parent elements

[Properties](#)

request

Specifies the request elements.

Content model

[Functions](#) or [Metadata](#) or [Actions](#)

Start_at

Specifies a search path for an additional set of data to be returned from the same request.

Content model

[Path](#) (*one or more*)

Parent elements

[Metadata](#)

MAResponse Elements

The MAResponse.xsd file defines the elements, attributes, and data types used in responses related to the published IBM Cognos Framework Manager model, or metadata tree.

This schema shows you examples of some of the elements and attributes returned as part of the response string for the `runSpecification(specification, parameterValues, options)` method.

Before you review the alphabetical list of elements that make up this response, you must understand its overall structure.

There is one top-level element, [“ResponseRoot” on page 1785](#), that must appear before any other response element.

Below are examples of the attributes that may be included in a response. Other than the ID attribute, which always appears, only those properties requested in the query will appear as attributes of the response.

- `description`
- `name`
- `_path`
- `_ref`
- `screenTip`
- `usage`

Most response attributes are of type `string`. For details about the data types used for each element in the metadata schemas, see the `MAQuery.xsd` and `MAResponse.xsd` files.

For more information about the properties of IBM Cognos Framework Manager models, see the model schema reference topic in the *IBM Cognos Framework Manager Developer Guide*.

calculation

Specifies the properties of any calculations used in the response.

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

description

Usage: optional Type: string

name

Usage: required Type: string

usage

Usage: optional Type: boolean

Content model

Empty element.

Parent elements

[folder](#)

decimalSeparator

Specifies how whole and decimal numbers are separated.

Attributes

value

Usage: required Type: NMTOKEN

Content model

Empty element.

Parent elements

[ResponseRoot](#)

defaultLocale

Specifies the model-specified default locale.

The locale string is expressed in the form of a language-region pair.

Content model

Content type is string.

Parent elements

[dimension](#) , [querySubject](#) , [ResponseRoot](#)

dimension

Attributes

ID

Usage: required Type: string

_path

Usage: optional Type: string

_ref

Usage: optional Type: string

bmtId

Usage: optional Type: string

description

Usage: optional Type: string

name

Usage: optional Type: string

Content model

([queryItem](#) (*any number*) then [defaultLocale](#)) then ([member](#) (*any number*) then (([level](#) or [hierarchy](#)) (*any number*) or [measure](#) (*any number*)))

Parent elements

[folder](#) , [ResponseRoot](#)

effectiveLocale

Specifies which of the supported model locales applies to this response.

This value is determined by the system. The locale string is expressed in the form of a language-region pair.

Content model

Content type is string.

Parent elements

[ResponseRoot](#)

filter

Specifies the properties of any filters used to narrow the scope of the response.

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

description

Usage: optional Type: string

name

Usage: required Type: string

Content model

Empty element.

Parent elements

[folder](#)

folder

Specifies the properties of any containers used for the response

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

description

Usage: optional Type: string

name

Usage: required Type: string

Content model

([querySubject](#) (*any number*) then [folder](#) (*any number*) then [calculation](#) (*any number*) then [filter](#) (*any number*) then [dimension](#) (*any number*)) (*any number*)

Parent elements

[folder](#) , [ResponseRoot](#)

function

Specifies the properties of the function used to narrow the scope of the response.

Attributes

ID

Usage: required Type: string

parameterType

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

date

numeric

string

time

unknown

unlimited

syntax

Usage: optional Type: string

tip

Usage: optional Type: string

type

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

function

literal

operator

summary

Content model

Empty element.

Parent elements

[group](#)

functionsRoot

Specifies a collection of [group](#) elements.

Content model

[group](#) (*one or more*)

Parent elements

[ResponseRoot](#)

group

Specifies the properties of any containers used for the response.

Attributes

ID

Usage: required Type: string

name

Usage: required Type: string

Content model

[function](#) (*any number*) then [group](#) (*any number*)

Parent elements

[functionsRoot](#) , [group](#)

hierarchy

Defines the type used for the hierarchy element.

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

description

Usage: optional Type: string

name

Usage: required Type: string

Content model

[level](#) (*any number*) then [member](#) (*any number*)

Parent elements

[dimension](#) , [ResponseRoot](#)

level

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

name

Usage: required Type: string

Content model

[queryItem](#) (*any number*)

Parent elements

[dimension](#) , [hierarchy](#) , [ResponseRoot](#)

listSeparator

Specifies how listed items are separated.

Attributes

value

Usage: required Type: string

Content model

Empty element.

Parent elements

[ResponseRoot](#)

locale

Specifies the model-specified locale.

The value must be expressed in the form of a paired language-region string, such as en-us or fr-fr. All IBM Cognos languages and regions are supported.

Content model

Content type is string.

Parent elements

[locales](#)

locales

Identifies the container used to hold the model-related locale information.

Defines the type used for the locale element

Content model

locale (*one or more*)

Parent elements

ResponseRoot

measure

Defines the type used for the Measure element

Attributes

ID

Usage: required Type: string

_path

Usage: optional Type: string

_ref

Usage: optional Type: string

bmtId

Usage: optional Type: string

datatype

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

characterlength16

datetime

decimal

float64

description

Usage: optional Type: string

displayType

Usage: optional Type: string

name

Usage: optional Type: string

promptCascadeOnRef

Usage: optional Type: string

promptDisplayItemRef

Usage: optional Type: string

promptFilterItemRef

Usage: optional Type: string

promptType

Usage: optional Type: string

promptUseItemRef

Usage: optional Type: string

regularAggregate

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

count

sum

unsupported

screenTip

Usage: optional Type: string

unSortable

Usage: optional Type: boolean

usage

Usage: required Type: NMTOKEN.

Attribute values are restricted to the following.

0

2

3

7

attribute

fact

identifier

unknown

Content model

measure (*any number*)

Parent elements

dimension , measure , ResponseRoot

member

Defines the type used for the member element. Note: for custom properties, the attribute name will be custom property name.

Attributes**_path**

Usage: optional Type: string

_ref

Usage: optional Type: string

dimensionUniqueName

Unique name of the dimension.

Usage: optional Type: string

hierarchyUniqueName

Unique name of the hierarchy.

Usage: optional Type: string

levelLabel

Level label for the member.

Usage: optional Type: string

levelNumber

Number of levels descended from root, starting at zero.

Usage: optional Type: unsignedInt

levelUniqueName

Unique name of the level.

Usage: optional Type: string

memberCaption

A user-friendly name for display purposes.

Usage: optional Type: string

memberUniqueName

Unique name of the member.

Usage: optional Type: string

parentUniqueName

Name of parent. If null, the member has no parent.

Usage: optional Type: string

Content model

member (*any number*)

Parent elements

dimension , hierarchy , member , ResponseRoot

modelSearchPath

Specifies the location of the model in Content Manager.

Attributes**value**

Usage: required Type: string

Content model

Empty element.

Parent elements

[ResponseRoot](#)

package

Identifies the package as published from the IBM Cognos Framework Manager component. The single name attribute must be a valid package name.

Attributes

name

Usage: required Type: string

Content model

Empty element.

Parent elements

[ResponseRoot](#)

queryItem

Specifies the properties of the query items in the response, as generated by the original request.

For more information about values and data types, see the IBM Cognos Framework Manager *Developer Guide*.

Attributes

ID

Usage: required Type: string

_path

Usage: optional Type: string

_ref

Usage: optional Type: string

bmtId

Usage: optional Type: string

datatype

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

characterlength16

datetime

decimal

float64

description

Usage: optional Type: string

displayType

Usage: optional Type: string

name

Usage: optional Type: string

promptCascadeOnRef

Usage: optional Type: string

promptDisplayItemRef

Usage: optional Type: string

promptFilterItemRef

Usage: optional Type: string

promptType

Usage: optional Type: string

promptUseItemRef

Usage: optional Type: string

regularAggregate

Usage: optional Type: NMTOKEN.

Attribute values are restricted to the following.

count

sum

unsupported

screenTip

Usage: optional Type: string

unSortable

Usage: optional Type: boolean

usage

Usage: required Type: NMTOKEN.

Attribute values are restricted to the following.

0

2

3

7

attribute

fact

identifier

unknown

Content model

Empty element.

Parent elements

[dimension](#) , [level](#) , [queryItemFolder](#) , [querySubject](#) , [ResponseRoot](#)

queryItemFolder

Specifies the properties of the container used for query items.

Attributes

ID

Usage: required Type: string

_path

Usage: required Type: string

_ref

Usage: required Type: string

name

Usage: required Type: string

Content model

[queryItem](#) (*any number*)

Parent elements

[ResponseRoot](#)

querySubject

Specifies the properties of the query subjects.

Attributes

ID

Usage: required Type: string

_path

Usage: optional Type: string

_ref

Usage: optional Type: string

bmtId

Usage: optional Type: string

description

Usage: optional Type: string

name

Usage: optional Type: string

Content model

[queryItem](#) (any number) then [defaultLocale](#)

Parent elements

[folder](#) , [ResponseRoot](#)

ResponseRoot

Specifies the properties of the query subjects.

Content model

[folder](#) (any number) then [functionsRoot](#) (optional) then [queryItem](#) (any number) then [queryItemFolder](#) (any number) then [querySubject](#) (any number) then [dimension](#) (any number) then [measure](#) (any number) then [member](#) (any number) then [level](#) (any number) then [hierarchy](#) (any number) then ([defaultLocale](#) (optional) then [locales](#) (optional) then [package](#) (optional) then [decimalSeparator](#) (optional) then [listSeparator](#) (optional) then [modelSearchPath](#) (optional) then [effectiveLocale](#) (optional))

Data Type Mapping for Response

When you call the method `runSpecification(specification, parameterValues, options)`, the response that is returned contains numeric data type identifiers. By contrast, the IBM Cognos Framework Manager user interface identifies data types by name.

An example of an excerpt of the response is as follows:

```
<ResponseRoot>
<folder ID="99:30" _ref="[gosales_goretailers]"
  name="gosales_goretailers"
  _path="[gosales_goretailers]">
  <querySubject
ID="100:115" _ref="[gosales_goretailers].[Orders]"
  name="Orders"
  _path="[gosales_goretailers].[Orders]" datatype="0">
  <queryItem
ID="101:192"
  _ref="[gosales_goretailers].[Orders].[Order
number]"
  name="Order number"
  _path="[gosales_goretailers].[Orders].[Order
number]"
  usage="7" datatype="11"/>
  ...
```

The following table cross-references the numeric identifiers from the `getMetadata` response with the corresponding names of the data types in the IBM Cognos Framework Manager user interface:

getMetadata response numeric identifier	Cognos Framework Manager user interface
0	unknown
1	unsupported
10	int16
11	int32
12	int64
20	decimal

Table 409. Cross-reference between getMetadata response numeric identifiers and Cognos Framework Manager user interface (continued)

getMetadata response numeric identifier	Cognos Framework Manager user interface
21	numeric
30	float
31	float32
32	float64
40	binary
41	binaryLength16
50	date
51	time
52	dateTime
53	timeInterval
60	character
61	characterLength16
62	characterLength32
70	blob
71	textBlob
72	blobArray
80	databaseKey

Chapter 36. Event specification reference

An event specification instance is used to query events in either the `monitorService` or the `eventManagementService`.

The `monitorService` will retrieve running or pending events. The `eventManagementService` will retrieve scheduled events. All exceptions are indicated in this reference chapter where applicable.

The schema file for the event specification is described in `eventSpecification.xsd`. The file is located in the `installation_location\webapps\p2pd\WEB-INF\classes\eventSpecification.xsd` directory.

For each event specification element, this section provides

- the name and description of the element
- information about attributes that apply to the element, including each attribute's name, description, optionality, legal values, and default value, if applicable
- content model information, consisting of a list of valid child elements presented as an element model group
- a list of valid parent elements

detailSeverity

Filters event records to return based on detail severity value.

Content model

Content type is string.

The possible values of this element are restricted to the following.

any

Returns event records with detail severity set to any.

debug

Returns event records with detail severity set to debug.

info

Returns event records with detail severity set to info.

warn

Returns event records with detail severity set to warn.

error

Returns event records with detail severity set to error.

fatal

Returns event records with detail severity set to fatal.

Parent elements

[filters](#)

dispatcherID

Filters event records to return based on the ID of the dispatcher on which the task is run. This element is only valid when you are retrieving information for the `monitorService`.

Content model

Content type is string.

Parent elements

[filters](#)

display

Specifies the number of event records to return to the client.

Content model

[start](#) (*optional*) then [end](#) (*optional*)

Parent elements

[queryEventSpecification](#)

end

Specifies the end index of the event records to return to the client.

The value must be greater than or equal to 1.

Note: If the `display` element is not explicitly defined in the query specification, the index of the last record returned will be calculated using the `PagingOptionEnum.maximumObjects` option. This option defaults to 10 in the `monitorService` and 20 in the `eventManagementService`. `PagingOptionEnum.maximumObjects` is an option in the options argument passed to the `runSpecification` method.

Content model

Content type is integer.

Parent elements

[display](#)

endTime

Filters event records to return based on end time. This element is optional for the `monitorService`.

The `startTime` and `endTime` delimit the time range of event records considered in a request. These elements define the total number of event records under consideration.

Content model

Content type is `dateTime`.

Parent elements

[filters](#) , [summary](#)

eventID

Specifies the event ID of the data to return.

This element must contain exactly 45 ASCII letters or digits. The letters can be uppercase or lowercase.

Content model

Content type is string.

The possible values of this element are restricted to the following.

Pattern: `[a-zA-Z0-9]{45}`

Parent elements

[scope](#)

filters

Filters to the scope of the query. This element is not required for restart scope requests : `restartEventID` and `restartParentEventID`.

Content model

`startTime` (*optional*) then `endTime` (*optional*) then `owner` (*optional*) then `user` (*optional*) then `scheduleTrigger` (*optional*) then `scheduleType` (*optional*) then `objectClass` (*optional*) then `priority` (*optional*) then `status` (*optional*) then `dispatcherID` (*optional*) then `detailSeverity` (*optional*)

Parent elements

[queryEventSpecification](#)

groupBy

Content model

Content type is string.

Parent elements

[summary](#)

objectClass

Filters event records to return based on objects defined in the object model.

Objects that can be run include Reports, Agents, Jobs, Analysis, etc, ...

Content model

Content type is string.

Parent elements

[filters](#)

order

Specifies the order of the sorted result.

Content model

Content type is string.

The possible values of this element are restricted to the following.

ascending

The order is ascending.

descending

the order is descending.

Parent elements

[sort](#) , [subSort](#)

owner

Filters event records to return based on the user that owns the task.

Content model

Content type is string.

Parent elements

[filters](#)

parentEventID

Limits the scope of query to events whose parent has this ID. Use this element to return information about child steps for jobs and agents.

Must contain exactly 45 ASCII letters or digits. The letters can be uppercase or lowercase.

Content model

Content type is string.

The possible values of this element are restricted to the following.

Pattern: `[a-zA-Z0-9]{45}`

Parent elements

[scope](#)

priority

Filters event records based on the priority of the task.

Content model

Content type is string.

The possible values of this element are restricted to the following.

any

Returns event records with priority set to any.

1

Returns event records with priority set to 1.

2

Returns event records with priority set to 2.

3

Returns event records with priority set to 3.

4

Returns event records with priority set to 4.

5

Returns event records with priority set to 5.

Parent elements

[filters](#)

queryEventSpecification

Root element for an event specification instance used to query events in either the `monitorService` or the `eventManagementService`.

Content model

[sort](#) (*optional*) then [subSort](#) (*optional*) then [scope](#) (*optional*) then [display](#) (*optional*) then [filters](#) then [summary](#) (*optional*)

restartEventID

Limits the scope of query to events with this ID. Use this element to return information about whether the specific event is restartable or not.

Must contain exactly 45 ASCII letters or digits. The letters can be uppercase or lowercase.

Content model

Content type is string.

The possible values of this element are restricted to the following.

Pattern: `[a-zA-Z0-9]{45}`

Parent elements

[scope](#)

restartParentEventID

Limits the scope of query to events whose parent has this ID. Use this element to return information about whether there are restartable events for this given parent id.

Must contain exactly 45 ASCII letters or digits; the letters can be uppercase or lowercase.

Content model

Content type is string.

The possible values of this element are restricted to the following.

Pattern: `[a-zA-Z0-9]{45}`

Parent elements

[scope](#)

scheduleTrigger

Filters event records based on schedule trigger name that was used to initiate the execution of this task.

Content model

Content type is string.

Parent elements

[filters](#)

scheduleType

Filters event records based on type of schedule to initiate this execution.

Content model

Content type is string.

The possible values of this element are restricted to the following.

any

Returns event records with schedule type set to any.

by day

Returns event records with schedule type set to by day.

by week

Returns event records with schedule type set to by week.

by month

Returns event records with schedule type set to by month.

by year

Returns event records with schedule type set to `by year`.

by trigger

Returns event records with schedule type set to `by trigger`.

manual

Returns event records with schedule type set to `manual`.

Parent elements

[filters](#)

scope

Restricts the events under consideration to events associated with objects at the specified `searchPaths` and IDs. The default value, if not specified is the entire content `"/content"`.

Content model

[searchPaths](#) or [eventID](#) or [parentEventID](#) or [restartEventID](#) or [restartParentEventID](#)

Parent elements

[queryEventSpecification](#)

searchPath

Limits the scope of query to the specified search path.

Content model

Content type is string.

Parent elements

[searchPaths](#)

searchPaths

Contains one or more `searchPath` elements.

Content model

[searchPath](#) (*one or more*)

Parent elements

[scope](#)

sort

Sorts the results returned from the service.

Content model

[order](#) then [sortItem](#)

Parent elements

[queryEventSpecification](#)

sortItem

Specifies the item by which to sort the results.

Content model

Content type is string.

The possible values of this element are restricted to the following.

startTime

The results are sorted by `startTime`.

status

scheduleType

The results are sorted by `scheduleType`.

priority

The results are sorted by `priority`.

Parent elements

[sort](#)

sortItem

Specifies the item to sort by.

Content model

Content type is string.

The possible values of this element are restricted to the following.

historyDetails.detailTime

Apply the sorting to the `historyDetail(s) detailTime` property.

Parent elements

[subSort](#)

start

Specifies the start index of the event records to return to the client.

The value must be greater than or equal to 1, and less than or equal to the end element.

Note: If the display element is not explicitly defined in the query specification, the index of the first record returned will be 1.

Content model

Content type is integer.

Parent elements

[display](#)

startTime

Filters event records to return based on start time. This element is optional for the `monitorService`.

The `startTime` and `endTime` delimit the time range of event records considered in a request. These elements define the total number of event records under consideration.

Content model

Content type is `dateTime`.

Parent elements

[filters](#) , [summary](#)

status

Filters event records to return based on status.

Filters event records to return based on status.

Content model

Content type is `string`.

The possible values of this element are restricted to the following.

all

Returns all event records with status of `cancelled`, and `scheduled` for `eventManagementService`.

cancelled

This enumeration value is only valid when you are retrieving information for the `eventManagementService`.

executing

This enumeration value is only valid when you are retrieving information for the `monitorService`.

failed

This enumeration value is only valid when you are retrieving information for the `contentManagerService`. Only history objects can have their status set to `failed`.

pending

This enumeration value is only valid when you are retrieving information for the `monitorService`.

purged

This enumeration value is only valid when you are retrieving information for the `contentManagerService`. Only history objects can have their status set to `purged`.

scheduled

This enumeration value is only valid when you are retrieving information for the `eventManagementService`.

succeeded

This enumeration value is only valid when you are retrieving information for the `contentManagerService`. Only history objects can have their status set to succeeded.

suspended

This enumeration value is only valid when you are retrieving information for the `monitorService`.

terminated

This enumeration value is only valid when you are retrieving information for the `contentManagerService`. Only history objects can have their status set to terminated.

inactive

This enumeration value is only valid when you are retrieving information for the `monitorService`.

Parent elements

[filters](#)

subSort

Applies a sub-level sort to the results returned from the service.

Content model

[order](#) then [sortItem](#)

Parent elements

[queryEventSpecification](#)

summary

Content model

[startTime](#) then [endTime](#) then [groupBy](#) (*optional*)

Parent elements

[queryEventSpecification](#)

user

Filters event records to return based on name of user running the task.

Content model

Content type is string.

Parent elements

[filters](#)

Chapter 37. Using selection context

A selection context instance is a dynamically created XML document that represents the values selected in a source resource. The selection context also contains information about the values that are dependent on the values selected.

The XML schema that defines a selection context instance is `SelectionContext.xsd`. For the `SelectionContext.xsd` schema reference, see the *Selection context reference*.

Viewing a selection context in Reporting

You can view a selection context instance for a selected cell in a report using Reporting.

Procedure

1. Open and run the report.
2. Right-click on the cell for which you want to view the selection context instance.
3. Select **Go to -> Related Links**.

The list of available links appears.

4. Click **View passed source values**.

The list of passed source values appears.

5. Click **More**.

The selection context instance appears.

Selection context example

The following graphic illustrates the report used to create the sample selection context instance that follows.

Return quantity		2004	2005	2006	2007
Personal Accessories	<u>Navigation</u>	6,088	4,576	5,727	7,316
	<u>Knives</u>	14,858	18,963	25,230	10,986
	<u>Watches</u>	7,423	10,173	11,107	5,814
	<u>Binoculars</u>	2,432	3,383	3,016	2,218
	<u>Eyewear</u>	29,795	38,610	49,169	30,016
Camping Equipment	<u>Tents</u>	10,012	10,914	9,301	9,279
	<u>Lanterns</u>	12,297	11,208	12,656	29,429
	<u>Sleeping Bags</u>	7,545	8,281	11,987	3,814
	<u>Cooking Gear</u>	26,443	27,167	55,335	32,786
	<u>Packs</u>	4,669	8,686	8,338	4,296
Golf Equipment	<u>Golf Accessories</u>	4,312	9,544	13,256	5,597
	<u>Woods</u>	926	956	850	527
	<u>Putters</u>	3,050	2,971	5,658	1,669
	<u>Irons</u>	562	1,128	1,474	767
Outdoor Protection	<u>First Aid</u>	2,703	2,699	1,755	919
	<u>Insect Repellents</u>	143,639	23,565	20,522	1,589
	<u>Sunscreen</u>	76,269	42,693	8,948	4,267
Mountaineering Equipment	<u>Safety</u>		2,726	4,990	3,157
	<u>Tools</u>		9,064	12,799	7,720
	<u>Rope</u>		835	2,483	512

Figure 21. Report for sample selection context instance

This sample XML is the selection context that corresponds to the highlighted cell.

```

<s:selection
  rModel="MP_0" xmlns:s="http://developer.cognos.com/schemas/selection/1/"
  rSelectedCells="C_0">
  <s:metadataCells>
    <s:metadataCell xml:id="M_0" rQueryName="QN_0" rName="N_0" rExpression="QI_0"
  rType="T_0" rUsage="U_0"/>
    <s:metadataCell xml:id="M_1" rName="N_1" rExpression="QI_1"
  rType="T_0" rUsage="U_1"/>
    <s:metadataCell xml:id="M_2" rName="N_2" rExpression="QI_2"
  rType="T_0" rUsage="U_1"/>
    <s:metadataCell xml:id="M_3" rName="N_3" rType="T_1"/>
    <s:metadataCell xml:id="M_4" rName="N_3" rType="T_1"/>
    <s:metadataCell xml:id="M_5" rName="N_4" rExpression="QI_3"
  rType="T_0" rUsage="U_1"/>
    <s:metadataCell xml:id="M_6" rName="N_3" rType="T_1"/>
    <s:metadataCell xml:id="M_7" rName="N_3" rType="T_1"/>
  </s:metadataCells>
  <s:cells>
    <s:cell xml:id="C_0" rMetadataCell="M_0" rValue="V_0">
      <s:dependentCells>
        <s:cell xml:id="C_7" rMetadataCell="M_7" rValue="V_7"/>
      </s:dependentCells>
      <s:definingCells>
        <s:axis rCell="C_1"/>
        <s:axis rCell="C_5"/>
      </s:definingCells>
    </s:cell>
    <s:cell xml:id="C_1" rMetadataCell="M_1" rValue="V_1">
      <s:dependentCells>
        <s:cell xml:id="C_4" rMetadataCell="M_4" rValue="V_4"/>
      </s:dependentCells>
      <s:definingCells>
        <s:axis rCell="C_2"/>
      </s:definingCells>
    </s:cell>
  </s:cells>

```

```

<s:cell xml:id="C_2" rMetadataCell="M_2" rValue="V_2">
  <s:dependentCells>
    <s:cell xml:id="C_3" rMetadataCell="M_3" rValue="V_3"/>
  </s:dependentCells>
</s:cell>
<s:cell xml:id="C_5" rMetadataCell="M_5" rValue="V_5">
  <s:dependentCells>
    <s:cell xml:id="C_6" rMetadataCell="M_6" rValue="V_6"/>
  </s:dependentCells>
</s:cell>
</s:cells>
<s:strings>
  <s:s xml:id="QN_0">Query1</s:s>
  <s:s xml:id="N_0">Return quantity</s:s>
  <s:s xml:id="N_1">Product type</s:s>
  <s:s xml:id="N_2">Product line</s:s>
  <s:s xml:id="N_3">_memberCaption</s:s>
  <s:s xml:id="N_4">Year</s:s>
  <s:s xml:id="T_0">memberUniqueName</s:s>
  <s:s xml:id="T_1">_memberCaption</s:s>
  <s:s xml:id="V_0">[Returned items (analysis)].[Returned items].[Return quantity]
</s:s>
  <s:s xml:id="V_1">[Product forecast (analysis)].[Product].[Product].[Product
type]-&gt;[all].[991].[953]</s:s>
  <s:s xml:id="V_2">[Product forecast (analysis)].[Product].[Product].[Product
line]-&gt;[all].[991]</s:s>
  <s:s xml:id="V_3">Camping Equipment</s:s>
  <s:s xml:id="V_4">Sleeping Bags</s:s>
  <s:s xml:id="V_5">[Product forecast (analysis)].[Time dimension].
[Time dimension].[Year]-&gt;[all].[2005]</s:s>
  <s:s xml:id="V_6">2005</s:s>
  <s:s xml:id="V_7">8281</s:s>
  <s:s xml:id="QI_0">
    [Returned items (analysis)].[Returned items].[Return quantity]
  </s:s>
  <s:s xml:id="QI_1">
    [Product forecast (analysis)].[Product].[Product].[Product type]
  </s:s>
  <s:s xml:id="QI_2">
    [Product forecast (analysis)].[Product].[Product].[Product line]
  </s:s>
  <s:s xml:id="QI_3">
    [Product forecast (analysis)].[Time dimension].[Time dimension].
[Year]</s:s>
  <s:s xml:id="U_0">measure</s:s>
  <s:s xml:id="U_1">nonMeasure</s:s>
  <s:s xml:id="MP_0">/content/folder[@name=&apos;Samples&apos;];
  /folder[@name=&apos;Models&apos;]/package[@name=&apos;G0
Sales (analysis)&apos;]/model[@name=&apos;model&apos;]</s:s>
</s:strings>
</s:selection>

```

Each selection context instance has three main parts. The first section, `<metadataCells>`, defines the metadata associated with the selected and dependent cells. The second section, `<cells>`, defines the selected cells, including the dependent and defining cells. The third section, `<strings>`, contains the strings that are re-used throughout the selection context.

Selection context for a multidimensional model

A selection context instance for a multidimensional model would contain more metadata for each of the selected cells. The selection context would also define more cells in order to accurately reflect the context of the selection. These cells are used to distinguish between the MUNs that identify the members referenced by the selection context from the member attribute values seen in the report.

Chapter 38. Selection context reference

This section provides reference information about each selection context element.

For each selection context element, this section provides

- the name and description of the element
- information about attributes that apply to the element, including each attribute's name, description, optionality, legal values, and default value, if applicable
- content model information, consisting of a list of valid child elements presented as an element model group
- a list of valid parent elements

axis

Specifies the axis for the defining cells.

Attributes

ordinal

Identifies the defining axis using the following values:

0

row

1

column

2

section

≥ 3

global

If the axis does not specify an ordinal value, the value can be determined by adding the number of preceding sibling axis elements between the current element and the nearest preceding sibling element that specifies an ordinal value to the specified ordinal value of that element (if no such element exists, the value 0 is used).

Each axis in an axis list must have a unique calculated ordinal value.

Usage: optional Type: int

rCell

This attribute must be included in every axis element.

Usage: required Type: IDREF

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

extension (*optional*) then any (*any number*)

Parent elements

[definingCells](#)

cell

Specifies the cells in the selection.

Attributes

depth

Reserved.

Usage: optional Type: int

rDrillActions

Reserved.

Usage: optional Type: IDREFS

rMetadataCell

This attribute must be included in every cell.

Usage: required Type: IDREF

rParentUniqueName

Reserved.

Usage: optional Type: IDREF

rValue

This attribute must be specified on every [cell](#) element and it must contain at least one xs:IDREF.

Usage: required Type: IDREFS

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

[definingCells](#) (*optional*) then [dependentCells](#) (*optional*) then [extension](#) (*optional*) then any (*any number*)

Parent elements

[cells](#) , [dependentCells](#)

cells

Specifies the cells in the selection.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([cell](#) or [extension](#) or any) (one or more)

Parent elements

[selection](#)

definingCells

Specifies the defining cells for the cell. A defining cell is a cell that provides values that are used to filter the data set to provide the value for the cell.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([axis](#) or [extension](#) or any) (one or more)

Parent elements

[cell](#)

dependentCells

Specifies the dependent cells for the cell. A dependent cell is a cell whose value is solely dependent on the cell.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

([cell](#) or [extension](#) or any) (one or more)

Parent elements

[cell](#)

extension

This element is an extensibility mechanism to allow elements from future versions of the target namespace to be added to the element while maintaining compatibility with the current version of the target namespace schema.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

any (*any number*)

Parent elements

[axis](#) , [cell](#) , [cells](#) , [definingCells](#) , [dependentCells](#) , [metadataCell](#) , [metadataCells](#) , [s](#) , [selection](#) , [sf](#) , [strings](#) , [value](#) , [values](#)

metadataCell

Specifies the metadata describing cells in the selection.

Attributes

depth

If this attribute is specified, the `rDimensionUniqueName` attribute must be specified.

Usage: optional Type: int

rDataItemName

Specifies the identity of the string that is the name of the data item that defines the cell values.

This attribute must be included in every metadata cell.

Usage: required Type: IDREF

rDimensionUniqueName

This attribute is only used with dimensionally-based models. This attribute may be included in any metadata cell.

Usage: optional Type: IDREF

rExpression

This attribute may be included in any metadata cell.

Usage: optional Type: IDREF

rHierarchyUniqueName

This attribute is only used with dimensionally-based models. This attribute may be included in any metadata cell.

If this attribute is specified, the `rDimensionUniqueName` attribute must be specified.

Usage: optional Type: IDREF

rLevelUniqueName

This attribute is only used with dimensionally-based models. This attribute may be included in any metadata cell.

If this attribute is specified, the following attributes must be specified: `rDimensionUniqueName` and `rHierarchyUniqueName`.

Usage: optional Type: IDREF

rQueryName

This attribute must be included in every metadata cell.

Usage: required Type: IDREF

rType

This attribute must be included in every metadata cell.

Usage: required Type: IDREF

rUsage

This attribute may be included in any metadata cell.

Usage: optional Type: IDREF

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

extension (*optional*) then any (*any number*)

Parent elements

metadataCells

metadataCells

Specifies the metadata describing cells in the selection.

Attributes**Adding Other Attributes**

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

(metadataCell or extension or any) (*one or more*)

Parent elements

selection

S

Specifies the strings in the selection.

Attributes**Adding Other Attributes**

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

Mixed content.

sf (*any number*) then extension (*optional*) then any (*any number*)

Parent elements

strings

selection

The root element of a selection context.

Attributes

rModel

This attribute must be included in any selection.

Usage: required Type: IDREF

rSelectedCells

This attribute must be included in any selection and it must contain at least one xs:IDREF

Usage: required Type: IDREFS

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

metadataCells then cells then strings then values (*optional*) then extension (*optional*) then any (*any number*)

sf

Reserved.

Attributes

rsf

This attribute must be included in every string fragment reference.

Usage: required Type: IDREF

Adding Other Attributes

anyAttribute indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and processContents parameters, respectively.

Content model

extension (*optional*) then any (*any number*)

Parent elements

s

strings

Specifies the strings used in the selection.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the namespace and `processContents` parameters, respectively.

Content model

(`s` or `extension` or any) (*one or more*)

Parent elements

[selection](#)

value

Reserved.

Attributes

inclusive

Specifies whether the value should be considered part of the selected value space.

Usage: optional Default: false Type: boolean

maxInclusive

If this attribute is specified, the `rMaxValue` attribute must be specified.

If this attribute is specified, the `rValue` attribute must not be specified.

Usage: optional Default: false Type: boolean

minInclusive

If this attribute is specified, the `rMinValue` attribute must be specified.

If this attribute is specified, `rValue` attribute must not be specified.

Usage: optional Default: false Type: boolean

rMaxValue

This attribute may be included in every value.

If this attribute is specified, the `rValue` attribute must not be specified.

Usage: optional Type: IDREF

rMinValue

This attribute may be included in every value.

If this attribute is specified, the `rValue` attribute must not be specified.

Usage: optional Type: IDREF

rNestedValues

This attribute may be included in every value. If this attribute is specified, it must contain at least one `xs:IDREF`.

If this attribute is specified, the following conditions must be met:

- the `rValue` attribute must be specified.
- the `inclusive` attribute must be specified and must have the value `true`.
- the following attributes must not be specified: `rMaxInclusive`, `rMaxValue`, `rMinInclusive` and `rMinValue`.

Usage: optional Type: IDREFS

rValue

If this attribute is specified, the following attributes must not be specified: `rMaxInclusive`, `rMaxValue`, `rMinInclusive` and `rMinValue`.

Usage: optional Type: IDREF

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

extension (*optional*) then any (*any number*)

Parent elements

values

values

Reserved.

Attributes

Adding Other Attributes

`anyAttribute` indicates that any attribute within the specified namespace(s) is permitted. Applicable namespace(s) and processing considerations are specified by the `namespace` and `processContents` parameters, respectively.

Content model

(value or extension or any) (*one or more*)

Parent elements

selection

Chapter 39. Repository Service API reference

You can navigate and retrieve resources stored in an external repository using the Repository Service REST API.

The API can be used with either through the Dispatcher Service or the Web server gateway. Responses are retrieved in the ATOM syndication format.

The REST API syntax is

`http://webservername:portnumber/ibmcognos/bi/v1/dispatch/repository/parameters...`
(gateway request)

`http://webservername:portnumber/p2pd/servlet/dispatch/ext/repository/parameters...` (Dispatcher Service request)

List report repositories

Returns a list of repositories that contain report versions for a given report.

Syntax

`rid/report_id`

Parameter

report_id

The storeID of the report.

Sample request (Dispatcher Service)

```
GET http://localhost:9300/p2pd/servlet/dispatch/ext/repository/rid/iA93243798ABF4F27A26B3C303AFD53C8
```

Sample response

```
HTTP/1.1 200 OK
Date: Tue, 28 Mar 2012 12:00:00 GMT
Transfer-Encoding: chunked
Content-Type: application/atom+xml;type=feed;charset="utf-8"

<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8</id>
  <title type="text">iA93243798ABF4F27A26B3C303AFD53C8</title>
  <author>
    <name>NONE</name>
  </author>
  <updated>2012-03-28T21:31:25.697Z</updated>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/repository/rid/iA93243798ABF4F27A26B3C303AFD53C8" rel="self" type="application/atom+xml">
  </link>
  <entry>
    <title type="text">iC13077127FA04E0EB79289714F3117F4</title>
    <id>tag:ibm.com,2012:iC13077127FA04E0EB79289714F3117F4</id>
    <updated>2012-03-21T22:37:51.602Z</updated>
    <summary type="text">iC13077127FA04E0EB79289714F3117F4</summary>
    <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/iA93243798ABF4F27A26B3C303AFD53C8" rel="alternate" type="application/atom+xml">
    </link>
    <content type="application/atom+xml" src="http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/iA93243798ABF4F27A26B3C303AFD53C8">
    </content>
```

```
</entry>
</feed>
```

List report versions

Returns a list of report versions for a given report.

Syntax

```
sid/repository_id/rid/report_id?ps=ps&si=si
```

Parameters

repository_id

The storeID of the data source connection for the repository.

report_id

The storeID of the report.

ps

The number of results to include per page of output. Optional. Default: 15.

si

The starting index in a collection when paging through results. Optional. Default: 0.

Sample request (Dispatcher Service)

```
GET http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid/
/iC13077127FA04E0EB79289714F3117F4/rid/iA93243798ABF4F27A26B3C303AFD53C8?ps=2
```

Sample response

```
HTTP/1.1 200 Ok
Date: Tue, 28 Mar 2012 12:00:00 GMT
Transfer-Encoding: chunked
Content-Type: application/atom+xml;type=feed;charset="utf-8"

<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8</id>
  <title type="text">iA93243798ABF4F27A26B3C303AFD53C8</title>
  <author>
    <name>NONE</name>
  </author>
  <updated>2012-04-02T20:36:54.379Z</updated>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/
/iA93243798ABF4F27A26B3C303AFD53C8?ps=2&si=0" rel="self"
type="application/atom+xml">
  </link>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/
/iA93243798ABF4F27A26B3C303AFD53C8?ps=2&si=0" rel="first"
type="application/atom+xml"> </link>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/
/iA93243798ABF4F27A26B3C303AFD53C8?ps=2&si=2" rel="next"
type="application/atom+xml">
  </link>
  <entry>
    <title type="text">iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22:37:51.602Z</title>
    <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22%3A37%3A51.602Z</id>
    <updated>2012-03-21T22:37:51.602Z</updated>
    <summary type="text">2012-03-21T22:37:51.602Z</summary>
    <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/
/iA93243798ABF4F27A26B3C303AFD53C8/vid/
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z"
rel="alternate" type="application/atom+xml">
```

```

</link>
<content type="application/atom+xml"
src="http://localhost:9300/p2pd/servlet/dispatch/ext
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid/iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22:37:51.602Z">
</content>
</entry>
<entry>
<title type="text">
iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:32:27.281Z
</title>
<id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22%3A32%3A27.281Z</id>
<updated>2012-03-21T22:32:27.281Z</updated>
<summary type="text">2012-03-21T22:32:27.281Z</summary>
<link href="http://localhost:9300/p2pd/servlet/dispatch
/ext/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:32:27.281Z"
rel="alternate" type="application/atom+xml">
</link>
<content type="application/atom+xml"
src="http://localhost:9300/p2pd/servlet/dispatch/ext/repository
/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid/iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22:32:27.281Z"> </content>
</entry>
</feed>

```

List report version outputs

Returns a list of report outputs for a given report version in a repository.

The response can contain the `format`, `locale`, and `busKey` properties for an output. See the output class for a description of these properties.

Syntax

`sid/repository_id/rid/report_id/vid/version_id?ps=ps&si=si`

Parameters

repository_id

The storeID of the data source connection for the repository.

report_id

The storeID of the report.

version_id

The ID of the report version.

ps

The number of results to include per page of output. Optional. Default: 15.

si

The starting index in a collection when paging through results. Optional. Default: 0.

Sample request (Dispatcher service)

```

GET
http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid
/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z

```

Sample response

```

HTTP/1.1 200 Ok
Date: Tue, 28 Mar 2012 12:00:00 GMT

```

Transfer-Encoding: chunked
Content-Type: application/atom+xml;type=feed; charset="utf-8"

```
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22%3A37%3A51.602Z</id>
  <title type="text">iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22:37:51.602Z</title>
  <author>
    <name>NONE</name>
  </author>
  <updated>2012-04-02T20:40:15.999Z</updated>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z"
rel="self" type="application/atom+xml">
  </link>
  <link href="http://localhost:9300/p2pd/servlet/dispatch/ext
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/vid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z"
rel="first" type="application/atom+xml">
  </link>
  <entry>
    <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8
_2012-03-21T22%3A37%3A51.602Z_iA63CD3C8C0DA4BE989F4C311CA7426D6_en-us</id>
    <title type="text">PDF</title>
    <updated>2012-03-21T22:37:51.602Z</updated>
    <link href="http://localhost:9300/p2pd/servlet/dispatch/ext
/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/oid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z
_iA63CD3C8C0DA4BE989F4C311CA7426D6_en-us/content"
rel="alternate" type="application/pdf">
    </link>
    <content type="application/pdf"
src="http://localhost:9300/p2pd/servlet/dispatch/ext/repository
/sid/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/oid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z
_iA63CD3C8C0DA4BE989F4C311CA7426D6_en-us/content">
    </content>
    <format xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text">PDF</format>
    <locale xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text">en-us</locale>
    <burstKey xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text"></burstKey>
  </entry>
  <entry>
    <id>tag:ibm.com,2012:iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22%3A37%3A51
.602Z_iBA59E68C2BA84A71902AA4D88BB6E640_en-us</id>
    <title type="text">HTML</title>
    <updated>2012-03-21T22:37:51.602Z</updated>
    <link href="http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid
/iC13077127FA04E0EB79289714F3117F4/rid/iA93243798ABF4F27A26B3C303AFD53C8/oid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z
_iBA59E68C2BA84A71902AA4D88BB6E640_en-us/content" rel="alternate"
type="text/html; charset=utf-8">
    </link>
    <content type="text/html; charset=utf-8"
src="http://localhost:9300/p2pd/servlet/dispatch/ext/repository/sid
/iC13077127FA04E0EB79289714F3117F4/rid/iA93243798ABF4F27A26B3C303AFD53C8
/oid/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z
_iBA59E68C2BA84A71902AA4D88BB6E640_en-us/content">
    </content>
    <format xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text">HTML</format>
    <locale xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text">en-us</locale>
    <burstKey xmlns="http://www.ibm.com/xmlns/prod/cognos/repositoryServiceAtom
/201204" type="text"></burstKey>
  </entry>
</feed>
```


Retrieve output content

Returns the requested report output.

Retrieved HTML output may contain relative URLs that link to external resources such as style sheets or images. To ensure that these relative URLs are correctly resolved, use a gateway request when retrieving HTML output. The URLs may not be resolved correctly when using a Dispatcher Service request. However, responses to report chart elements are retrieved correctly using either of the two calling methods.

Syntax

```
sid/repository_id/rid/report_id/oid/output_id_id/content
```

Parameters

repository_id

The storeID of the data source connection for the repository.

report_id

The storeID of the report.

output_id

The ID of the report output. In Content Manager, this is represented as storeID. In other repositories, output_id is the ID for the specific provider (for example, FileNet ID).

Sample request (Dispatcher service)

```
GET
http://localhost/ibmcognos/bi/v1/disp/repository/sid
/iC13077127FA04E0EB79289714F3117F4/rid
/iA93243798ABF4F27A26B3C303AFD53C8/oid
/iA93243798ABF4F27A26B3C303AFD53C8_2012-03-21T22:37:51.602Z
_iA63CD3C8C0DA4BE989F4C311CA7426D6_en-us/content
```

Sample response

```
HTTP/1.1 200 Ok
Accept-Ranges bytes
Cache-Control private
Content-Disposition inline; filename="Simple Chart Query.pdf"
Content-Language en-us
Content-Type application/pdf
Date Thu, 22 Mar 2012 17:20:23 GMT
Expires Thu, 21 Mar 2013 13:20:23 EDT

binary PDF data
```

External references in HTML output

HTML output may contain relative URLs to external resources, such as skins and styles. In these cases, the request is redirected based on the incoming URI. When accessing archived HTML output the gateway should be used instead of the dispatcher because the dispatcher may not be able to resolve the relative locations. References to report chart elements are retrieved correctly in either case.

Sample request for external references

```
GET
http://localhost/ibmcognos/cgibin/
cognos.cgi/repository/sid/iC13077127FA04E0EB79289714F3117F4/rid/
iA93243798ABF4F27A26B3C303AFD53C8/o
id/skins/corporate/viewer/QSRVSelection.css
```

Sample response for external references

```
HTTP/1.1 302 Found
Location: http://localhost/ibmcognos/skins/corporate/viewer/QSRVSelection.css
```

Sample response for external references from the dispatcher

```
HTTP/1.1 302 Found
Location: http://localhost:9300/p2pd/skins/corporate/viewer/QSRVSelection.css
```

A request through the dispatcher may result in a "Not Found" (404) response if the dispatcher is unable to resolve the URI.

Retrieve default output

Provides a reference to the requested report's default output in the content store.

Requests to list a report's repositories ([“List report repositories” on page 1811](#)) provide a link to the content store location as well as optionally providing a reference to any archive repositories that have been enabled for the report.

Given the storeID of a report, a request for default report output (instead of an archived version) results in a redirect response to the appropriate location in the content store.

Syntax

```
/repository/sid/cm/rid/report_id/oid/default/content[/mht/content]
```

Parameters

report_id

The storeID of the report.

Sample request (Dispatcher service)

```
GET
http://localhost/ibmcognos/cgibin/
cognos.cgi/repository/sid/cm/rid/iA93243798ABF4F27A26B3C303AFD53C8/oid
/default/content
```

Sample response

```
HTTP/1.1 302 Found
Location: http://localhost/ibmcognos/cgibin/
cognos.cgi/repository/sid/cm/rid/iBA79F920BB0A44638A6CC9D528DBDC15/
oid/i8E01B7C5C0E747EDA18ADE9D348
45A2A/content
Date Thu, 22 Mar 2012 17:20:23 GMT
```

Appendix A. New in lineage specification

The following changes have been made to the lineage specification.

Changes are categorized into the following sections:

- **Added** means the listed items have been added to the report specification.
- **Deprecated** means the listed items are deprecated and will be removed in a future version of the product. These items are still functional but we recommend that you remove them from all of your report specifications.
- **Obsolete** means that the listed items are obsolete and will be removed in a future version of the product.
- **Internal** means the listed items should not be added to your report specifications.

Schema version 10.1.0

The following changes were made in schema version 10.1.0

Added

- [V5QuerySet](#) element.

Deprecated

No deprecated items.

Obsolete

No obsolete items.

Internal

No internal-only items.

Appendix B. New in Version 11.1.2

This appendix provides information about the new features in Version 11.1.2 of the product.

Support for Jobs and Notebooks

Support for jobs and Jupyter notebooks have been added to the IBM Cognos Software Development Kit.

This change affects:

- [bibus](#) » [productCapability](#) class
- [bibus](#) » [userCapabilityEnum](#) » [canUseJob](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseNotebook](#) value

New Capability Cookie

A new expanded capability cookie ([userCapabilitiesEx](#)) has been added, which extends the number of supported cookie bits.

The capability cookie is used to store the set of global capabilities granted to the session during logon. The cookie does not store object capabilities.

The cookie value is determined by assigning each capability a specific bit in a bit array. Bits are assigned right to left, starting with bit 0. The bit for a capability is assigned the value 1 if the user has the global capability, and 0 otherwise.

The bit array is converted to an array of 32-bit unsigned integer values, which are then hex encoded, with leading zeroes removed. The resulting words are concatenated and separated by semicolons (;). The resulting string is then signed to allow Cognos Analytics to detect tampering.

The bit index for each capability is documented with the [bibus](#) » [userCapabilityEnum](#) enumeration set.

For information on cookie handling, refer to [“Querying Capabilities Using the Capability Cookie Bit Index”](#) on page 52.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) enumeration set

Appendix C. New in Version 10.2.2

This appendix provides information about the new features in Version 10.2.2 of the product.

For additional information about changes in Version 10.2.2, see *IBM Cognos Analytics New Features*.

Connecting to a datasource using namespace credentials

You can now connect to a datasource using credentials retrieved from a namespace you are already authenticated against, if the namespace and the datasource use the same authentication mechanism.

This change affects

- [bibus](#) » [retrieveCredential](#) method

Data Stores renamed to Dynamic Cubes in Cognos Administration

On the **Status** tab in IBM Cognos Administration, **Data Stores** has been renamed to **Dynamic Cubes**.

This change affects

- [“Dynamic Cubes”](#) on page 1732
- [“Dynamic Cubes”](#) on page 1734

My data sets

Individual users can import data from a CSV, XLS or XLSX file that is stored on their computers and create a stand-alone package that is based on that data. Administrators control access to this functionality through the My Data Sets capability.

This change affects

- [bibus](#) » [model](#) » [dataSetBase](#) property
- [bibus](#) » [dataSourceCapabilityEnum](#) » <http://developer.cognos.com/ceba/constants/dataSourceCapabilityEnum#lobData> enumeration value
- [bibus](#) » [installedComponentEnum](#) » [myDataSets](#) enumeration value
- [bibus](#) » [dataSet](#) class
- [bibus](#) » [dataSetFolder](#) class
- [canUseMyDataSets](#) capability
- [canUseMyDataSetsAdministration](#) capability
- [“My Data Sets Administration”](#) on page 1623
- [“My Data Sets”](#) on page 1653
- [“Data Sets”](#) on page 1726
- [“Data Sets”](#) on page 1732

Report output format restriction

In IBM Cognos Analytics, to manage system resources, administrators can now restrict the ability of users to run reports in the CVS, PDF, Microsoft Excel, and XML report output formats. These report output

formats are now separately secured functions that control the format options that users can see and run in the user interface.

This change affects

- [“Generate CSV Output” on page 1645](#)
- [“Generate PDF Output” on page 1646](#)
- [“Generate XLS Output” on page 1647](#)
- [“Generate XML Output” on page 1647](#)
- [canGenerateCSVOutput](#) capability
- [canGeneratePDFOutput](#) capability
- [canGenerateXLSOutput](#) capability
- [canGenerateXMLOutput](#) capability

Data source connection information

You can now test a JDBC data source connection and obtain information about the database and the JDBC driver.

This change affects

- [bibus](#) » [testDataSourceConnectionWithInfo](#) method

Dynamic cube messages

You can now retrieve warning and error messages for dynamic cubes.

This change affects

- [bibus](#) » [getCubeMessages](#) method
- [bibus](#) » [rolapCubeMessages](#) class
- [bibus](#) » [rolapMessage](#) class
- [bibus](#) » [asynchDetailROLAPMessages](#)

Dynamic cube metrics

You can now retrieve metrics for dynamic cubes.

This change affects

- [bibus](#) » [getCubeMetrics](#) method
- [bibus](#) » [asynchDetailROLAPMetrics](#) class
- [bibus](#) » [rolapCubeMetrics](#) class
- [bibus](#) » [rolapMetric](#) class
- [bibus](#) » [rolapMetricDateTime](#)
- [bibus](#) » [rolapMetricNumber](#)
- [bibus](#) » [rolapMetricString](#)

New options to manage dynamic cubes

You can now pause a dynamic cube and incrementally update the cube data.

This change affects:

- [bibus](#) » [incrementallyLoadCubes\(cubeNames, parameterValues, options\)](#) method
- [bibus](#) » [pauseCubes\(cubeNames, parameterValues, options\)](#) method
- [bibus](#) » [queryTaskOptionEnum](#) » <http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#incrementallyLoadROLAPCube> enumeration value
- [bibus](#) » [queryTaskOptionEnum](#) » <http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#incrementallyLoadROLAPCubeContext> enumeration value
- [bibus](#) » [queryTaskOptionEnum](#) » <http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#pauseROLAPCubes> enumeration value
- [bibus](#) » [queryTaskOptionEnum](#) » <http://developer.cognos.com/ceba/constants/queryTaskOptionEnum#pauseROLAPCubesContext> enumeration value
- [bibus](#) » [rolapDataSourceStateEnum](#) » [paused](#) enumeration value
- [bibus](#) » [rolapDataSourceStateEnum](#) » [pausing](#) enumeration value

Full tenant impersonation capability for system administrators

System administrators can impersonate a single tenant to view and interact with the content from the tenant perspective.

System administrators can impersonate tenants from IBM Cognos Connection and IBM Cognos Administration, or by using the software development kit.

This change affects:

- [bibus](#) » [identity](#) class
- [bibus](#) » [tenancy](#) class
- [bibus](#) » [getIdentity](#) method

Changes to IBM Cognos Analytics SOAP action for services

This release introduces changes to support interoperability between IBM Cognos components.

As a result of these changes, the SOAPAction HTTP header field for a number of services has changed. If you depend on these fields in your custom applications, you must update your applications accordingly.

When creating SDK applications, you must use the dll files (for C# applications) or jar files (for Java applications) that are included with the version of IBM Cognos Analytics that your applications are targeting.

This change affects:

- [agentService](#) service
- [batchReportService](#) service
- [contentManagerService](#) service
- [dataIntegrationService](#) service
- [dataMovementService](#) service
- [deliveryService](#) service
- [dimensionManagementService](#) service

- [dispatcher](#) service
- [eventManagementService](#) service
- [idVizService](#) service
- [indexSearchService](#) service
- [indexUpdateService](#) service
- [jobService](#) service
- [metadataService](#) service
- [migrationService](#) service
- [mobileService](#) service
- [monitorService](#) service
- [planningAdministrationConsoleService](#) service
- [planningRuntimeService](#) service
- [planningTaskService](#) service
- [powerPlayService](#) service
- [queryService](#) service
- [relationalMetadataService](#) service
- [reportService](#) service
- [repositoryService](#) service
- [saCAMService](#) service
- [systemService](#) service

WebSphere Liberty Profile replaces Apache Tomcat

IBM Cognos Analytics now installs a WebSphere Liberty Profile as the application server. Apache Tomcat is no longer installed with IBM Cognos Analytics.

This change affect the Java Server Page (JSP) samples. For more information, see [Java Server Page Samples](#)

New dynamic cube configuration properties

Five new properties have been added to the `baseROLAPCubeConfiguration` and `rolapCubeConfiguration` classes. Three of these properties replace advanced settings in previous releases.

The following table shows advanced settings that have been replaced by `rolapCubeConfiguration` properties

Advanced setting	<code>rolapCubeConfiguration</code> property
<code>qsMaxAggregateLoadThreads</code>	<code>maxAggregateLoadThreads</code>
<code>qsMaxCubeLoadThreads</code>	<code>maxHierarchyLoadThreads</code>
<code>qsMeasuresThreshold</code>	<code>measuresThreshold</code>

The following properties have also been added to the `rolapCubeConfiguration` class:

- `automaticAggregateOptimizationEnabled`
- `postInMemoryTriggerName`.

This change affects:

- [bibus](#) » [rolapCubeConfiguration](#) » [automaticAggregateOptimizationEnabled](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [maxAggregateLoadThreads](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [maxHierarchyLoadThreads](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [measuresThreshold](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [postInMemoryTriggerName](#) property
- [bibus](#) » [automaticAggregateOptimizationEnum](#) enumeration set

Reporting user interface profiles

The IBM Cognos Analytics - Reporting user interface can now be customized using profiles maintained by Library Administrators.

Reporting user interface profiles can only be enabled by Library Administrators in the IBM Cognos Administration user interface and not by using the IBM Cognos Software Development Kit. See the *IBM Cognos Analytics Administration and Security Guide* for more information.

This change affects:

- [bibus](#) » [catalogFolder](#) class
- [bibus](#) » [userInterfaceProfile](#) class
- “Library” on page 1670
- “User Interface Profiles” on page 1672
- “Cognos Workspace Do More Profiles” on page 1672
- “Reporting Profiles” on page 1673
- “User Interface Profiles” on page 1723
- “User Interface Profiles” on page 1721

Deprecation of Reporting profiles

Reporting profiles have been deprecated and will be removed in a future release of this product.

Customer should use [Reporting user interface profiles](#) instead.

This change affects:

- [bibus](#) » [reportStudioOption](#) class
- [bibus](#) » [reportStudioOptionSearchPathSingleObject](#) class
- [bibus](#) » [uiProfile](#) class
- [bibus](#) » [uiProfileFolder](#) class
- [bibus](#) » [reportStudioOptionEnum](#) enumeration set
- “Express - deprecated” on page 1679
- “Reporting Profiles - deprecated” on page 1678
- “User Interface Profiles - deprecated” on page 1678
- “Professional - deprecated” on page 1680

Dynamically rename excel sheet in a multi-tab excel output

The RSVP.EXCEL.PAGEGROUP_WSNAME_ITEMVALUE advanced setting was added to support this feature.

This change affects:

- RSVP.EXCEL.PAGEGROUP_WSNAME_ITEMVALUE advanced setting in [“Report service and batch report service advanced settings” on page 1597](#)

Support for delegated tenant administration

Tenant administration tasks can now be delegated to members of the Tenant Administrators role.

This change affects:

- [bibus](#) » [asynchDetailEventRecord](#) » [tenantID](#) property
- [bibus](#) » [copyOptions](#) » [keepTenancy](#) property
- [bibus](#) » [tenant](#) » [tenantMembers](#) property
- [bibus](#) » [queryTenantMembership](#) method
- [“Tenant Administrators” on page 1709](#)

Support for new advanced settings

Additional advanced settings have been added.

This change affects:

The following advanced settings in [“Report service and batch report service advanced settings” on page 1597](#) were added.

- BDS.split.maxKeysPerChunk
- RSVP.BURST_DISTRIBUTION
- RSVP.BURST_QUERY_PREFETCH
- RSVP.CONCURRENTQUERY.ENABLEDFORINTERACTIVEOUTPUT
- RSVP.CONCURRENTQUERY.MAXNUMHELPERSPERREPORT
- RSVP.CONCURRENTQUERY.NUMHELPERSPERPROCESS
- RSVP.CSV.REPEAT_XTAB_LABELS
- RSVP.DRILL.DynamicFilterUsesBusinessKey
- RSVP.EXCEL.EXCEL_2007_LARGE_WORKSHEET
- RSVP.EXCEL.EXCEL_2007_OUTPUT_FRAGMENT_SIZE
- RSVP.EXCEL.EXCEL_2007_WORKSHEET_MAXIMUM_ROWS
- RSVP.EXCEL.PAGEGROUP_WSNAME_ITEMVALUE
- RSVP.EXCEL.XLS2007_PRINT_MEDIA
- RSVP.PARAMSCACHEDISABLED
- RSVP.PRINT.POSTSCRIPT
- RSVP.PROMPT.CASTNUMERICSEARCHKEYTOSTRING
- RSVP.PROMPT.EFFECTIVEPROMPTINFO.IGNORE
- RSVP.PROMPT.RECONCILIATION
- RSVP.PROMPT.RECONCILIATION.CHUNKSIZE

- `RSVP.PROMPT.RECONCILIATION.CHUNKSIZE`

Documentation Updates

Several miscellaneous bug fixes were applied in this release.

This change affects:

- [bibus » baseClass » usage property](#)
- [“IBM Cognos Viewer” on page 1447](#)
- [“Catalog - removed” on page 1669](#)
- [“Library” on page 1670](#)
- [“Visualizations” on page 1671](#)
- [“Catalog Administrators - removed” on page 1693](#)
- [“Library Administrators” on page 1699](#)
- [“Tenants” on page 1683](#)
- [“Multitenancy” on page 1719](#)
- [“Tenants” on page 1720](#)
- [“Library” on page 1720](#)
- [“Visualizations” on page 1721](#)
- [Chapter 16, “Enumeration sets,” on page 1241](#)

Appendix D. New in Version 10.2.1

This appendix provides information about the new features in Version 10.2.1 of the product.

For additional information about changes in Version 10.2.1, see IBM Cognos Analytics *New Features*.

External object store for report archiving

You can now configure the Content Manager to store report output to a local drive or network share by defining an external content store. When a report is run and saved, the default behavior is to save the output to the content store. This feature allows an administrator to configure an external object store for report output. This reduces the size of the content store and can provide performance improvements for Content Manager.

This change affects:

- [bibus](#) » [graphic](#) » [dataPathInfo](#) property
- [bibus](#) » [output](#) » [dataPathInfo](#) property
- [bibus](#) » [configuration](#) » [unixRepositoryURI](#) property
- [bibus](#) » [configuration](#) » [windowsRepositoryURI](#) property

Support for archiving a namespace or namespaceFolder

The [bibus](#) » [account](#), [bibus](#) » [namespace](#), and [bibus](#) » [namespaceFolder](#) classes have been extended to include properties for repository rules. This change provides a simplified approach for administrators archiving multiple accounts within the same namespace.

This change affects:

- [bibus](#) » [account](#) » [repositoryRules](#) property
- [bibus](#) » [content](#) » [repositoryRules](#) property
- [bibus](#) » [folder](#) » [repositoryRules](#) property
- [bibus](#) » [namespace](#) » [repositoryRules](#) property
- [bibus](#) » [namespaceFolder](#) » [repositoryRules](#) property
- [bibus](#) » [package](#) » [repositoryRules](#) property

Enhanced tenant administration functionality

This release introduces changes related to tenant administration enhancements. New functionality includes the ability to include or exclude public content when deploying tenants, and the ability for administrators to obtain content store utilization information about individual tenants.

See the IBM Cognos Analytics *Administration and Security Guide* for more information about tenant administration.

This change affects:

- [standaloneCAM](#) » [terminateSessions\(search\)](#) method
- [bibus](#) » [contentStoreUtilizationConfiguration](#) class
- [bibus](#) » [tenant](#) class

- [bibus](#) » [tenants](#) class
- [bibus](#) » [directory](#) » [tenants](#) property
- [bibus](#) » [configuration](#) » [userProfile](#) property
- [bibus](#) » [contentTaskOptionEnum](#) » [createContentStoreUtilizationInfo](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [tenantsContentConflictResolution](#) value
- [standaloneCAM](#) method set
- “Tenants” on page 1683
- [bibus](#) » [baseClass](#) » [tenantID](#) property

Performance enhancements

Changes were made to affinity properties for the graphicsService in an effort to improve report rendering performance.

This change affects:

- [bibus](#) » [configuration](#) » [gsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [gsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [gsNonAffineConnections](#) property
- [bibus](#) » [graphicsService](#) » [gsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [gsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [gsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [gsPeakNonAffineConnections](#) property
- [bibus](#) » [graphicsService](#) » [gsPeakNonAffineConnections](#) property

Removal of statistics service

Parts of the IBM Cognos Software Development Kit have been changed as a result of removal of IBM Cognos Statistics.

The following items have been removed:

- [bibus](#) » [statisticsService](#) class
- [bibus](#) » [dispatcher](#) » [statisticsService](#) property
- [bibus](#) » [installedComponentEnum](#) » [statisticsService](#) value
- [Statistics](#) Authors and [Statistics](#) object definitions
- All properties with a prefix of "sts" in the [bibus](#) » [configuration](#) class, the [bibus](#) » [configurationFolder](#) class, and the [bibus](#) » [dispatcher](#) class.

Update your applications if they use any of the affected items.

Administrative changes for Cognos Analytics Mobile Reports

Reserved.

This change affects:

- [bibus](#) » [configuration](#) » [mobileConfiguration](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUseMobileAdministration](#) value
- “Mobile Administrators” on page 1702
- “Mobile Administration” on page 1623

Changes to IBM Cognos Analytics SOAP action for services

This release introduces changes to support interoperability between IBM Cognos components.

As a result of these changes, the SOAPAction HTTP header field for a number of services has changed. If you depend on these fields in your custom applications, you must update your applications accordingly.

When creating SDK applications, you must use the dll files (for C# applications) or jar files (for Java applications) that are included with the version of IBM Cognos Analytics that your applications are targeting.

This change affects:

- [agentService](#) service
- [batchReportService](#) service
- [contentManagerService](#) service
- [dataIntegrationService](#) service
- [dataMovementService](#) service
- [deliveryService](#) service
- [dimensionManagementService](#) service
- [dispatcher](#) service
- [eventManagementService](#) service
- [idVizService](#) service
- [indexSearchService](#) service
- [indexUpdateService](#) service
- [jobService](#) service
- [metadataService](#) service
- [migrationService](#) service
- [mobileService](#) service
- [monitorService](#) service
- [planningAdministrationConsoleService](#) service
- [planningRuntimeService](#) service
- [planningTaskService](#) service
- [powerPlayService](#) service
- [queryService](#) service
- [relationalMetadataService](#) service
- [reportService](#) service
- [repositoryService](#) service
- [saCAMService](#) service
- [systemService](#) service

Visualization support

New content store container objects and types facilitate the storage and management of elements to support visualizations in reports.

A visualization is a visual representation of data. For example, a map, a network diagram, or a bar graph. A visualization object in the content store is a module that allows authors to use visualizations in their reports. Visualization elements contained within this module, such as code and graphics items such as

icons, are stored using the `bibus » visualization` class. Instances of this class are organized within `bibus » catalogFolder` objects contained by a single `bibus » catalog` object in the `bibus » root` folder of the content store. A new role, Catalog Administrator, allows precise control over who is able to manage this content.

This change affects:

- `bibus » catalog` class
- `bibus » catalogFolder` class
- `bibus » visualization` class
- `bibus » root » catalog` property
- “Catalog - removed” on page 1669
- “Visualizations” on page 1671
- “Catalog Administrators - removed” on page 1693

New configuration options for burst distribution

New configuration options allow administrators more control of certain aspects of bursting reports. These options administrators to improve the performance of these burst operations.

For the majority of configurations, the default values are optimal. If any performance issues are encountered, careful adjustment of these options can make more efficient use of resources when bursting reports.

For more information, see the IBM Cognos *Administration and Security Guide*.

This change affects:

- `bibus » runOptionEnum » burstDistribution` value
- `bibus » runOptionEnum » burstKeyLimit` value
- `bibus » runOptionEnum » burstQueryPrefetch` value

New standalone IBM Cognos Access Manager (CAM) service

This previously reserved feature, introduced in Version 10.2.0, is now available for use.

This feature introduces a new service to support IBM Cognos Access Manager (CAM) functions.

This change affects:

- `bibus » dispatcher » saCAMService` property
- `bibus » installedComponentEnum » saCAMService` value
- `bibus » CAMPassport » authOrigin` property
- `bibus » configuration » saCAMAuditLevel` property
- `bibus » configurationFolder » saCAMAuditLevel` property
- `bibus » dispatcher » saCAMAuditLevel` property
- `bibus » saCAMService » saCAMAuditLevel` property
- `saCAMService` service

Developer description

This new service allows other IBM Cognos components to perform authentication and authorization operations without having a content manager service instance available. This can provide easier administration of services and reduce the amount of resources required for security operations.

Repository Service API supports retrieval of default report output

You are now able to retrieve the current version of report output from Content Manager with the Repository Service REST API. In previous releases, the API was limited to only accessing archived versions of report output. The API now treats the content store as a repository, by using `/repository/sid/cm` in the syntax of the URI.

Requests to list a report's repositories provide a link to the content store location as well as optionally providing a reference to any archive repositories that have been enabled for the report.

Given the `storeID` of a report, a request for default report output (instead of an archived version) results in a redirect response to the appropriate location in the content store.

Refer to the Repository Service API reference chapter for more information.

End of line characters removed for MHT and XLWA output

When the `bibus » outputEncapsulationEnum` enumeration set is set to `none` and `MHT` or `XLWA` is specified, the end of line character sequence, Carriage Return (#13) and Line Feed (#10), is normalized into a single Line Feed (#10) character within the encapsulated XML document, leaving only the Line Feed character. As many Microsoft® Windows applications cannot process these formats without the full end of line character sequence, the output is treated as invalid by consuming applications.

Refer to the referenced troubleshooting topic for information on how to correct this condition in your applications.

This change affects:

- [“End of line characters removed for MHT and XLWA output” on page 110](#)

Appendix E. New in Version 10.2.0

This appendix provides information about the new features in Version 10.2.0 of the product.

Relational metadata service

This previously reserved feature, introduced in Version 10.1.0 as “[Relational Metadata Service](#)” on page 1858, is now available for use.

This new service has been introduced to extract relational metadata from data sources. A new `canImportRelationalMetadata` capability allows administrators the ability to determine which users can access this new functionality.

This change affects:

- `bibus` » `relationalMetadataServiceSpecification` class
- `bibus` » `dispatcher` » `relationalMetadataService` property
- `bibus` » `configuration` » `rmadsConnections` property
- `bibus` » `configurationFolder` » `rmadsConnections` property
- `bibus` » `dispatcher` » `rmadsConnections` property
- `bibus` » `relationalMetadataService` » `rmadsConnections` property
- `bibus` » `configuration` » `rmadsPeakConnections` property
- `bibus` » `configurationFolder` » `rmadsPeakConnections` property
- `bibus` » `dispatcher` » `rmadsPeakConnections` property
- `bibus` » `relationalMetadataService` » `rmadsPeakConnections` property
- `relationalMetadataService` service

Support for bi-directional languages

New user preferences have been added to support bi-directional (BiDi) languages in IBM Cognos Analytics.

Bi-directional language support adds the ability to display text in both right-to-left (RTL) and left-to-right (LTR) directions. While English, French, and German read from left-to-right, languages such as Hebrew or Arabic read from right-to-left.

This change affects:

- `bibus` » `baseTextDirectionEnum` enumeration set
- `bibus` » `biDirectionalOptionEnum` enumeration set
- `bibus` » `baseTextDirectionEnum` » `Auto` value
- `bibus` » `biDirectionalOptionEnum` » `baseTextDirection` value
- `bibus` » `biDirectionalOptionEnum` » `biDirectionalFeaturesEnabled` value
- `bibus` » `baseTextDirectionEnum` » `LTR` value
- `bibus` » `baseTextDirectionEnum` » `RTL` value

Graduated dashboard capabilities

New capabilities have been added to restrict usage of IBM Cognos Workspace.

These new capabilities provide administrators with more granular control over the consumption, interaction, or the ability to author dashboard content. “Authors” on page 1693, “Query Users” on page 1706, and “Report Administrators” on page 1707 are granted these new capabilities by default.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseAdvancedDashboardFeatures](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseInteractiveDashboardFeatures](#) value
- “Use Advanced Dashboard Features” on page 1641
- “Use Interactive Dashboard Features” on page 1642
- “Executive Dashboard” on page 1640

IBM Cognos Dynamic Cubes Administration

This previously reserved feature, introduced in Version 10.1.1 as “IBM Cognos ROLAP Administration” on page 1847, is now available for use.

ROLAP data sources are supported in IBM Cognos Analytics. The IBM Cognos Software Development Kit provides administration and configuration functionalities to cube and server administrators.

Cube administrators can issue cube management commands, such as starting, stopping, and restarting cubes; retrieving cube states; refreshing member and data caches; and applying cube security. Before commands can be issued, a ROLAP cube must be modeled using IBM Cognos Cube Designer, published as a data source in Content Manager, and configured and assigned to a specific [queryService](#).

Server administrators can manage ROLAP cubes and configure cube properties on a [queryService](#) to achieve optimal query performance and cube usage.

See the *IBM Cognos Analytics Dynamic Query Guide* and the *IBM Cognos Analytics Dynamic Cubes User Guide* for more information.

This change affects:

- [rolapCubeAdministration](#) » [clearCubeWorkloadLog\(cubeNames, parameterValues, options\)](#) method
- [bibus](#) » [aliasLocation](#) class
- [bibus](#) » [aliasRoot](#) class
- [bibus](#) » [configuration](#) » [aliasLocations](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [enableDynamicMemberCaching](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [memberCacheLimit](#) property
- [bibus](#) » [configuration](#) » [qsROLAPMemberCacheAliasRoot](#) property
- [bibus](#) » [configurationFolder](#) » [qsROLAPMemberCacheAliasRoot](#) property
- [bibus](#) » [dispatcher](#) » [qsROLAPMemberCacheAliasRoot](#) property
- [bibus](#) » [queryService](#) » [qsROLAPMemberCacheAliasRoot](#) property

IBM Cognos Dynamic Cubes Aggregate Advisor configuration

IBM Cognos Dynamic Cubes Aggregate Advisor recommends cube aggregation, based on available cube definitions and optional query workload, to improve query performance. New configuration options allow cube administrators the ability to manage aggregate data, such as logging, caching, and setting memory constraints.

See the *IBM Cognos Dynamic Query Analyzer User Guide* for more information.

This change affects:

- [bibus](#) » [queryTaskOptionEnum](#) » [clearROLAPCubeWorkloadLog](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [clearROLAPCubeWorkloadLogContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [startROLAPCubesAndSourceCubes](#) value
- [bibus](#) » [rolapCubeConfiguration](#) » [aggregateCacheSize](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [aggregateLoggingEnabled](#) property
- [bibus](#) » [rolapDataSource](#) » [aggregates](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [disableExternalAggregates](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [multiDimensionalQuerySizeLimit](#) property
- [bibus](#) » [rolapCubeConfiguration](#) » [predicateMemberReferenceThreshold](#) property
- [bibus](#) » [configuration](#) » [qsMultiDimensionalQuerySizeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [qsMultiDimensionalQuerySizeLimit](#) property
- [bibus](#) » [dispatcher](#) » [qsMultiDimensionalQuerySizeLimit](#) property
- [bibus](#) » [queryService](#) » [qsMultiDimensionalQuerySizeLimit](#) property

New queryService configuration options

New [bibus](#) » [queryService](#) class properties have been added to provide administrators with more options for optimizing JVM settings and performance.

This change affects:

- [bibus](#) » [gcPolicyEnum](#) enumeration set
- [bibus](#) » [gcPolicyEnum](#) » [Balanced](#) value
- [bibus](#) » [gcPolicyEnum](#) » [Custom](#) value
- [bibus](#) » [gcPolicyEnum](#) » [Generational](#) value
- [bibus](#) » [configuration](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [configurationFolder](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [dispatcher](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [queryService](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [configuration](#) » [qsDisableVerboseGCLogging](#) property
- [bibus](#) » [configurationFolder](#) » [qsDisableVerboseGCLogging](#) property
- [bibus](#) » [dispatcher](#) » [qsDisableVerboseGCLogging](#) property
- [bibus](#) » [queryService](#) » [qsDisableVerboseGCLogging](#) property
- [bibus](#) » [configuration](#) » [qsGCPolicy](#) property
- [bibus](#) » [configurationFolder](#) » [qsGCPolicy](#) property
- [bibus](#) » [dispatcher](#) » [qsGCPolicy](#) property
- [bibus](#) » [queryService](#) » [qsGCPolicy](#) property
- [bibus](#) » [configuration](#) » [qsGenerateCommentsInNativeSQL](#) property
- [bibus](#) » [configurationFolder](#) » [qsGenerateCommentsInNativeSQL](#) property
- [bibus](#) » [dispatcher](#) » [qsGenerateCommentsInNativeSQL](#) property
- [bibus](#) » [queryService](#) » [qsGenerateCommentsInNativeSQL](#) property
- [bibus](#) » [configuration](#) » [qsInitialJVMNurserySize](#) property
- [bibus](#) » [configurationFolder](#) » [qsInitialJVMNurserySize](#) property
- [bibus](#) » [dispatcher](#) » [qsInitialJVMNurserySize](#) property

- [bibus](#) » [queryService](#) » [qsInitialJVMMNurserySize](#) property
- [bibus](#) » [configuration](#) » [qsJVMMNurserySizeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [qsJVMMNurserySizeLimit](#) property
- [bibus](#) » [dispatcher](#) » [qsJVMMNurserySizeLimit](#) property
- [bibus](#) » [queryService](#) » [qsJVMMNurserySizeLimit](#) property
- [bibus](#) » [configuration](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [configurationFolder](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [dispatcher](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [queryService](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [configuration](#) » [qsVerboseGCLogLimit](#) property
- [bibus](#) » [configurationFolder](#) » [qsVerboseGCLogLimit](#) property
- [bibus](#) » [dispatcher](#) » [qsVerboseGCLogLimit](#) property
- [bibus](#) » [queryService](#) » [qsVerboseGCLogLimit](#) property

New standalone IBM Cognos Access Manager (CAM) service

Reserved.

This change affects:

- [bibus](#) » [saCAMService](#) class
- [bibus](#) » [dispatcher](#) » [saCAMService](#) property
- [bibus](#) » [installedComponentEnum](#) » [saCAMService](#) value
- [bibus](#) » [CAMPassport](#) » [authOrigin](#) property
- [bibus](#) » [configuration](#) » [saCAMAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [saCAMAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [saCAMAuditLevel](#) property
- [bibus](#) » [saCAMService](#) » [saCAMAuditLevel](#) property
- [saCAMService](#) service

Support for multi-tenancy

Multi-tenancy provides the capability to support multiple customers or organizations (tenants) using a single deployment of an application. In a multi-tenancy configuration, each tenant can access only the data that they are authorized to use. Applications that support multi-tenancy are called multi-tenant applications. Multi-tenant applications minimize the extra costs associated with these environments.

Administrators can associate a [bibus](#) » [baseClass](#) » [tenantID](#) property with a user account in the content store and manage its content. At creation time, each object in the content store is populated with a tenant ID value that is based on the tenant ID associated with the current user's session. The IBM Cognos Software Development Kit provides methods to list and delete tenants and its descendants.

See the *IBM Cognos Analytics Administration and Security Guide* for more information about multi-tenancy support.

This change affects:

- [content](#) » [deleteTenants\(tenantIDs\)](#) method
- [content](#) » [listTenants\(options\)](#) method
- [bibus](#) » [deploymentOptionStringArray](#) class
- [bibus](#) » [listTenantsOptions](#) class

- [bibus](#) » [tenantInfo](#) class
- [bibus](#) » [deploymentOptionEnum](#) » [entireContentStoreReplace](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [tenants](#) value
- [content](#) method set
- [bibus](#) » [baseClass](#) » [tenantID](#) property
- [bibus](#) » [updateOptions](#) » [updateTenantIDRecursive](#) property

Deprecation of IBM Cognos PowerPlay capabilities and objects

The [bibus](#) » [userCapabilityEnum](#) » [canOpenPowerPlayInReportStudio](#) and [bibus](#) » [userCapabilityEnum](#) » [canOpenPowerPlayInAnalysisStudio](#) capabilities, and “[Open PowerPlay Reports with Analysis Studio](#)” on page 1630 and “[Open PowerPlay Reports with Reporting](#)” on page 1660 objects, are now deprecated.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canOpenPowerPlayInAnalysisStudio](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canOpenPowerPlayInReportStudio](#) value
- “[Open PowerPlay Reports with Analysis Studio](#)” on page 1630
- “[Open PowerPlay Reports with Reporting](#)” on page 1660

Support for interactive discovery and visualization

Reserved.

This change affects:

- [bibus](#) » [configuration](#) class
- [bibus](#) » [configurationFolder](#) class
- [bibus](#) » [idVizService](#) class
- [bibus](#) » [dispatcher](#) » [idVizService](#) property
- [bibus](#) » [installedComponentEnum](#) » [idVizService](#) value
- [bibus](#) » [configuration](#) » [idVizAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [idVizAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [idVizAuditLevel](#) property
- [bibus](#) » [idVizService](#) » [idVizAuditLevel](#) property
- [idVizService](#) service

Support for SAP BW hierarchy variables

SAP hierarchy variables support has been added to IBM Cognos Analytics to enhance report creation capabilities and improve product performance. While *hierarchy node variables* were supported in earlier releases, with *hierarchy variables*, users can select specific hierarchies when running reports that contain dimensions with multiple hierarchies.

A new data type enumeration has been added to associate dimensions belonging to a particular hierarchy.

This change affects:

- [bibus](#) » [parameterDataTypeEnum](#) » [hierarchyUniqueName](#) value

Changes to IBM Cognos Analytics SOAP action for services

This release introduces changes to support interoperability between IBM Cognos components.

As a result of these changes, the SOAPAction HTTP header field for a number of services has changed. If you depend on these fields in your custom applications, you must update your applications accordingly.

When creating SDK applications, you must use the dll files (for C# applications) or jar files (for Java applications) that are included with the version of IBM Cognos Analytics that your applications are targeting.

This change affects:

- [agentService](#) service
- [batchReportService](#) service
- [contentManagerService](#) service
- [dataIntegrationService](#) service
- [dataMovementService](#) service
- [deliveryService](#) service
- [dimensionManagementService](#) service
- [dispatcher](#) service
- [eventManagementService](#) service
- [indexSearchService](#) service
- [indexUpdateService](#) service
- [jobService](#) service
- [metadataService](#) service
- [migrationService](#) service
- [mobileService](#) service
- [monitorService](#) service
- [planningAdministrationConsoleService](#) service
- [planningRuntimeService](#) service
- [planningTaskService](#) service
- [powerPlayService](#) service
- [queryService](#) service
- [relationalMetadataService](#) service
- [reportService](#) service
- [systemService](#) service

Repository service

Reserved.

This change affects:

- [bibus](#) » [repositoryService](#) class
- [bibus](#) » [dispatcher](#) » [repositoryService](#) property
- [bibus](#) » [installedComponentEnum](#) » [repositoryService](#) value
- [bibus](#) » [configuration](#) » [reposAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [reposAuditLevel](#) property

- [bibus](#) » [dispatcher](#) » [reposAuditLevel](#) property
- [bibus](#) » [repositoryService](#) » [reposAuditLevel](#) property
- [bibus](#) » [configuration](#) » [reposCacheObjTTL](#) property
- [bibus](#) » [configurationFolder](#) » [reposCacheObjTTL](#) property
- [bibus](#) » [dispatcher](#) » [reposCacheObjTTL](#) property
- [bibus](#) » [repositoryService](#) » [reposCacheObjTTL](#) property
- [bibus](#) » [configuration](#) » [reposNumObjDisk](#) property
- [bibus](#) » [configurationFolder](#) » [reposNumObjDisk](#) property
- [bibus](#) » [dispatcher](#) » [reposNumObjDisk](#) property
- [bibus](#) » [repositoryService](#) » [reposNumObjDisk](#) property
- [bibus](#) » [configuration](#) » [reposNumObjMem](#) property
- [bibus](#) » [configurationFolder](#) » [reposNumObjMem](#) property
- [bibus](#) » [dispatcher](#) » [reposNumObjMem](#) property
- [bibus](#) » [repositoryService](#) » [reposNumObjMem](#) property
- [repositoryService](#) service

Excel 2007 Data output format

A new Excel 2007 Data format is available when generating report output. With this format users can generate native Microsoft® Excel 2007 data containing list report data. Users can then move data into Microsoft® Excel for further manipulation. By minimizing data formatting, similar to CSV output, this option provides product performance improvements.

This change affects:

- [bibus](#) » [outputFormatEnum](#) » [xlsxData](#) value

IBM Business Process Server integration

IBM Cognos Workspace can now integrate with IBM Business Process Manager for improved decision-making.

Administrators can use the new [bibus](#) » [configuration](#) » [bpmRestURI](#) property to specify the REST URI of the Business Process Manager server.

This change affects:

- [bibus](#) » [configurationData](#) » [bpmRestURI](#) property
- [bibus](#) » [configurationDataEnum](#) » [bpmRestURI](#) value
- [bibus](#) » [configuration](#) » [bpmRestURI](#) property

Restricting access to Cognos Insight in IBM Cognos Analytics

A new capability has been added to restrict access to Cognos Insight. Assigning a user this capability grants them access to Cognos Insight from the Welcome Page and IBM Cognos Connection.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseCognosInsight](#) value
- “Cognos Insight Users” on page 1694
- “Cognos Insight” on page 1631

Restricting access to Cognos Analytics Mobile Reports Service in IBM Cognos Analytics

A new capability has been added to restrict access to IBM Cognos Analytics Mobile Reports. Granting users, groups, or roles access to the capability will allow them to access IBM Cognos Analytics services from Cognos Analytics Mobile Reports.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseMobileService](#) value
- “Mobile Users” on [page 1702](#)
- “Mobile” on [page 1652](#)

Changes to .NET Framework support

The .NET Framework libraries included with the IBM Cognos Software Development Kit now support .NET Framework 4.0, while earlier versions are no longer supported. For more information about supported software environments, visit the IBM Cognos Customer Center (<http://www.ibm.com/support/docview.wss?uid=swg27037784>).

SDK applications that reference `cognosdotnet` DLL assemblies and namespace must be updated. See “Changes to .NET Framework support” on [page 121](#) for information on upgrading SDK applications to use .NET Framework 4.0.

Updated default settings for Report Service and Batch Report Service

Affinity connection settings in [bibus](#) » [configuration](#) class, [bibus](#) » [configurationFolder](#) class and [bibus](#) » [dispatcher](#) class have been increased to provide improved memory usage and product performance. Updated settings apply to new installations only and do not impact previous customer settings in older product installations.

The updated default settings for [bibus](#) » [reportService](#) class are the following:

- `rsNonAffineConnections` = 8
- `rsAffineConnections` = 2
- `rsPeakNonAffineConnections` = 8
- `rsPeakAffineConnections` = 2

The new default settings for [bibus](#) » [batchReportService](#) class are the following:

- `brsNonAffineConnections` = 4
- `brsAffineConnections` = 2
- `brsPeakNonAffineConnections` = 4
- `brsPeakAffineConnections` = 2

This change affects:

- [bibus](#) » [batchReportService](#) » [brsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [brsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsAffineConnections](#) property
- [bibus](#) » [batchReportService](#) » [brsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsNonAffineConnections](#) property

- [bibus](#) » [configurationFolder](#) » [brsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsNonAffineConnections](#) property
- [bibus](#) » [batchReportService](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [batchReportService](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsNonAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsPeakNonAffineConnections](#) property

New Repository Service (REST) API

A new Repository Service REST API allows for navigation and retrieval only of resources stored in an external repository. The response format is the Atom Syndication Format as defined in RFC42871. For more information, see the Repository Service API reference chapter.

This change affects:

- [bibus](#) » [account](#) » [repositoryRules](#) property
- [bibus](#) » [content](#) » [repositoryRules](#) property
- [bibus](#) » [folder](#) » [repositoryRules](#) property
- [bibus](#) » [namespace](#) » [repositoryRules](#) property
- [bibus](#) » [namespaceFolder](#) » [repositoryRules](#) property
- [bibus](#) » [package](#) » [repositoryRules](#) property

ReportNET API is obsolete

The CognosReportNetService service and its methods are obsolete and have been removed.

Appendix F. New in Version 10.1.1

This appendix provides information about the new features in Version 10.1.1 of the product.

Support for IBM Cognos Content Archival

This previously reserved feature, introduced in Version 10.1.0 as [“IBM Cognos Content Manager/Enterprise Content Management Integration”](#) on page 1866, is now available for use.

IBM Cognos Content Archival introduces support for archiving versioned report output in the content store to an external repository, such as IBM FileNet Content Manager. This can improve system performance by reducing the size of the content store, and can help customers adhere to regulatory requirements. An external repository can be configured as a data source connection by an IBM Cognos administrator. Content in a package or folder enabled for archival by the administrator is saved to the external repository by running an IBM Cognos Content Manager archive task.

This service is not accessible with the SOAP toolkits. However, a REST API is available that allows for navigation and retrieval of resources stored in an external repository.

This change affects:

- [bibus](#) » [repositoryRule](#) class
- [bibus](#) » [account](#) » [repositoryRules](#) property
- [bibus](#) » [content](#) » [repositoryRules](#) property
- [bibus](#) » [folder](#) » [repositoryRules](#) property
- [bibus](#) » [namespace](#) » [repositoryRules](#) property
- [bibus](#) » [namespaceFolder](#) » [repositoryRules](#) property
- [bibus](#) » [package](#) » [repositoryRules](#) property
- [bibus](#) » [dataSourceCapabilityEnum](#) enumeration set
- [bibus](#) » [userCapabilityEnum](#) » [canUpdateRepositoryRules](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseRepository](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canViewContentInRepository](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentDoNotWriteToRepository](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentUpdateContextObjects](#) value
- [bibus](#) » [systemMetricEnum](#) » [failedExternalizedDocumentsPercent](#) value
- [bibus](#) » [systemMetricEnum](#) » [numberOfFailedExternalizedDocuments](#) value
- [bibus](#) » [systemMetricEnum](#) » [numberOfSuccessfulExternalizedDocuments](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [preserveStoreIDs](#) value
- [bibus](#) » [dataSourceCapabilityEnum](#) » [repository](#) value
- [bibus](#) » [installedComponentEnum](#) » [repositoryIntegration](#) value
- [bibus](#) » [systemMetricEnum](#) » [successfulExternalizedDocumentsPercent](#) value
- [“Manage repository connections”](#) on page 1644
- [“View external documents”](#) on page 1645
- [“External Repositories”](#) on page 1644

JMX Proxy Server Scalability

This previously reserved feature, introduced in Version 10.1.0 as “[JMX Proxy Server Scalability](#)” on page 1871 is now available for use.

New configuration parameters are now available to provide improved management for a Java Management Extensions (JMX) proxy server in the IBM Cognos Analytics server environment.

See the *IBM Cognos Analytics Administration and Security Guide* for more information.

This change affects:

- [bibus](#) » [configuration](#) » [jmxProxyHostDispatchers](#) property
- [bibus](#) » [configuration](#) » [activeJMXProxyURI](#) property

Changes to IBM Cognos Analytics SOAP action for services

This release introduces changes to support interoperability between IBM Cognos components.

As a result of these changes, the SOAPAction HTTP header field for a number of services has changed. If you depend on these fields in your custom applications, you must update your applications accordingly.

Applications must be compiled on the same version of the IBM Cognos Software Development Kit as that used by the IBM Cognos Analytics environment that you intend to deploy your applications in.

This change affects:

- [agentService](#) service
- [contentManagerService](#) service
- [eventManagementService](#) service
- [jobService](#) service
- [monitorService](#) service
- [queryService](#) service

Run with Owner Capabilities

This previously reserved feature, introduced in Version 10.1.0 as “[Run with Owner Capabilities](#)” on page 1871 is now available for use. This feature allows users to run a report using specific capabilities granted to the owner of the report, even if they do not have these capabilities themselves.

If a report, for example, uses HTML items or user-defined SQL, the [canUseHTML](#) value or the [canUseUserDefinedSQL](#) value capabilities are required to run the report. These capabilities are normally granted to a limited set of users, because the potential to create reports with malicious intent using either HTML or SQL is high. Setting the new [runWithOwnerCapabilities](#) property to `true` allows consumers of a report to run it using specific capabilities of the owner, even if they do not have the required capabilities themselves. This differs from [bibus](#) » [baseReport](#) » [runAsOwner](#) property in that it includes only the owner's capabilities and excludes other aspects of the owner's security context, such as access to data sources.

During normal report execution, permissions and capabilities used to run a task are that of the caller making the run request.

When [bibus](#) » [baseReport](#) » [runAsOwner](#) property is set to `true`, users other than the owner can run a task that requires access to privileged data. The task runs using the security credentials of the owner, not those of the user running the task.

When [bibus](#) » [authoredReport](#) » [runWithOwnerCapabilities](#) property is `true`, users other than the owner can run a task that requires additional capabilities. The task runs using the capabilities of the owner, not those of the user running the task.

The security context used for data access and capabilities used in all of these instances is summarized in the following table:

<i>Table 411. Security context for data access and capability checks</i>			
runAsOwner value	runWithOwnerCapabilities value	Security context used for data access	Security context used for capability checks*
false	false	Session	Session
false	true	Session	Owner
true	—	Owner	Owner

* Defined in the User Capability Details section of the [bibus » userCapabilityEnum](#) enumeration set.

In all cases, the normal permissions defined in Content Manager for the target object still apply.

This new property provides more specific control by allowing administrators to restrict the use of capabilities to only those users who really need them.

Existing applications do not have to change as a result of this new property.

This change affects:

- [bibus » authoredReport » runWithOwnerCapabilities](#) property

HttpOnly Attribute Support for the CAM_Passport Cookie

Support has been added to allow administrators to set the `HttpOnly` attribute of the `CAM_Passport` cookie.

Setting the `HttpOnly` attribute prevents a web browser from being able to read or manipulate the `CAM_Passport` cookie that identifies a user's session to the server. With this attribute set, the browser uses the cookie to send HTTP requests back to the server but does not allow scripts to access the contents of the cookie. Malicious scripts are often inserted into the browser using a Cross Site Scripting (XSS) attack on the same server or web application that the user has authenticated to. If the `HttpOnly` attribute is set, the browser does not allow access to the cookie, preventing such a script from stealing a user's identity.

Administrators who want to enable this feature should ensure that users have a browser that supports the `HttpOnly` attribute.

This change affects:

- [bibus » configuration » cookieCAMPassportHttpOnly](#) property

IBM Cognos ROLAP Administration

Reserved.

This change affects:

- [rolapCubeAdministration » getCubeState\(cubeNames, parameterValues, options\)](#) method
- [rolapCubeAdministration » refreshCubeDataCache\(cubeNames, parameterValues, options\)](#) method
- [rolapCubeAdministration » refreshCubeMemberCache\(cubeNames, parameterValues, options\)](#) method
- [rolapCubeAdministration » refreshCubeSecurity\(cubeNames, parameterValues, options\)](#) method
- [rolapCubeAdministration » restartCubes\(cubeNames, parameterValues, options\)](#) method
- [rolapCubeAdministration » startCubes\(cubeNames, parameterValues, options\)](#) method

- [rolapCubeAdministration](#) » [stopCubes\(cubeNames, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailROLAPDataSourceState](#) class
- [bibus](#) » [baseROLAPCubeConfiguration](#) class
- [bibus](#) » [baseROLAPDataSource](#) class
- [bibus](#) » [rolapCubeConfiguration](#) class
- [bibus](#) » [rolapDataSource](#) class
- [bibus](#) » [rolapVirtualCubeConfiguration](#) class
- [bibus](#) » [rolapVirtualDataSource](#) class
- [bibus](#) » [namespace](#) » [items](#) property
- [bibus](#) » [configuration](#) » [qsROLAPCubeConfigurations](#) property
- [bibus](#) » [configurationFolder](#) » [qsROLAPCubeConfigurations](#) property
- [bibus](#) » [dispatcher](#) » [qsROLAPCubeConfigurations](#) property
- [bibus](#) » [queryService](#) » [qsROLAPCubeConfigurations](#) property
- [bibus](#) » [rolapDataSourceStateEnum](#) enumeration set
- [bibus](#) » [rolapOptionEnum](#) enumeration set
- [bibus](#) » [queryTaskOptionEnum](#) » [getROLAPCubeState](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [getROLAPCubeStateContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeDataCache](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeDataCacheContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeMemberCache](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeMemberCacheContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeSecurity](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [refreshROLAPCubeSecurityContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [restartROLAPCubes](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [restartROLAPCubesContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [startROLAPCubes](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [startROLAPCubesContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [stopROLAPCubes](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [stopROLAPCubesContext](#) value
- [bibus](#) » [queryTaskOptionEnum](#) » [stopROLAPCubesImmediately](#) value
- [rolapCubeAdministration](#) method set
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [dataCacheSizeLimit](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [dataSourceName](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [disabled](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [disableResultSetCache](#) property
- [bibus](#) » [configuration](#) » [qsAdditionalJVMArguments](#) property
- [bibus](#) » [configurationFolder](#) » [qsAdditionalJVMArguments](#) property
- [bibus](#) » [dispatcher](#) » [qsAdditionalJVMArguments](#) property
- [bibus](#) » [queryService](#) » [qsAdditionalJVMArguments](#) property
- [bibus](#) » [configuration](#) » [qsInitialJVMHeapSize](#) property
- [bibus](#) » [configurationFolder](#) » [qsInitialJVMHeapSize](#) property
- [bibus](#) » [dispatcher](#) » [qsInitialJVMHeapSize](#) property
- [bibus](#) » [queryService](#) » [qsInitialJVMHeapSize](#) property

- [bibus](#) » [configuration](#) » [qsJVMHeapSizeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [qsJVMHeapSizeLimit](#) property
- [bibus](#) » [dispatcher](#) » [qsJVMHeapSizeLimit](#) property
- [bibus](#) » [queryService](#) » [qsJVMHeapSizeLimit](#) property
- [bibus](#) » [configuration](#) » [qsManualCubeStart](#) property
- [bibus](#) » [configurationFolder](#) » [qsManualCubeStart](#) property
- [bibus](#) » [dispatcher](#) » [qsManualCubeStart](#) property
- [bibus](#) » [queryService](#) » [qsManualCubeStart](#) property
- [bibus](#) » [configuration](#) » [qsResultSetCacheQueryTimeThreshold](#) property
- [bibus](#) » [configurationFolder](#) » [qsResultSetCacheQueryTimeThreshold](#) property
- [bibus](#) » [dispatcher](#) » [qsResultSetCacheQueryTimeThreshold](#) property
- [bibus](#) » [queryService](#) » [qsResultSetCacheQueryTimeThreshold](#) property
- [bibus](#) » [configuration](#) » [qsROLAPCubeAdministrationCommandTimeout](#) property
- [bibus](#) » [configurationFolder](#) » [qsROLAPCubeAdministrationCommandTimeout](#) property
- [bibus](#) » [dispatcher](#) » [qsROLAPCubeAdministrationCommandTimeout](#) property
- [bibus](#) » [queryService](#) » [qsROLAPCubeAdministrationCommandTimeout](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [resultSetCacheSizeLimit](#) property
- [bibus](#) » [baseROLAPCubeConfiguration](#) » [startupTriggerName](#) property
- [queryService](#) service

Deprecation of [qsDisableQueryPlanCache](#)

The [bibus](#) » [configuration](#) » [qsDisableQueryPlanCache](#), [bibus](#) » [configurationFolder](#) » [qsDisableQueryPlanCache](#), [bibus](#) » [dispatcher](#) » [qsDisableQueryPlanCache](#), and [bibus](#) » [queryService](#) » [qsDisableQueryPlanCache](#) properties are now deprecated and will be removed in a future version of the product. Support for the properties is maintained for this release; however, you should update your applications by discontinuing use of these properties.

This change affects:

- [bibus](#) » [configuration](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [configurationFolder](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [dispatcher](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [queryService](#) » [qsDisableQueryPlanCache](#) property

Personal Data Source Signons

The “[personalDataSourceSignon\(dataSourceName\)](#)” on [page 1570](#) search path function has been revised. Using the [personalDataSourceSignon](#) function does *not* return [dataSourceCredential](#) objects when a [dataSourceSignon](#) is not found for a matching [dataSource](#) and [dataSourceConnection](#). This change in [personalDataSourceSignon](#) behavior maintains consistency with the [dataSourceSignon](#) function.

Maximum process configuration values for the statistics service are not used

Client applications cannot set configuration properties for the [statisticsService](#) class related to the maximum number of processes during peak and non-peak times.

It is possible to set these values, however they are not used by the service.

Changes to monitor service advanced settings

Administrators must specify a value equal to or greater than the minimum specified for the following `monitorService` advanced settings:

1. `connection.write.maxConnections`
2. `connection.read.maxConnections`

If a value less than the minimum of 5 was specified, the setting had no effect and the service uses the default value. This information was not documented in previous releases.

Improving the utilization of system resources by using the `asynch » release(conversation)` method

Every asynchronous conversation reserves server-side resources to handle client requests. These reserved resources cannot be used for other purposes while the associated asynchronous conversation exists. Although asynchronous conversations are terminated after a period of inactivity, more effective use of server-side resources can be realized when clients call `asynch » release(conversation)` when they have no further use for an asynchronous conversation. This allows these reserved resources to be used for other purposes.

The `asynch » release(conversation)` method can be called whenever the conversation status is `complete`. If the status is `working` or `stillWorking`, clients can free up server-side resources by calling `asynch » cancel(conversation)`. If the conversation status is `conversationComplete`, the server-side components have already made all resources associated with the asynchronous conversation available for other purposes and no further action is required by the client to terminate the asynchronous conversation.

IBM Cognos Viewer is an example of an SDK client that uses `asynch » release(conversation)` to make the most effective use of server-side resources possible in your IBM Cognos installation.

This change affects:

- `asynch » release(conversation)` method

Changes to `runSpecification()` method for `eventManagementService` and `monitorService`

Capability and caller requirements have been added to the `asynch » runSpecification(specification, parameterValues, options)` method. Conditions apply when a `asynch » runSpecification(specification, parameterValues, options)` method is called against either `eventManagementService` or `monitorService` to manage interactive and scheduled tasks.

Appendix G. New in Version 10.1.0

This appendix provides information about the new features in Version 10.1.0 of the product.

External Data

A new capability and associated object now limits using external data as a source for content authors.

External data refers to data not included in a package managed by an administrator. For example, a resource on a user's local file system.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseExternalData](#) value
- [“Allow External Data”](#) on page 1657

Personal Data Source Credentials

Individual users can now save their data source credentials without administrator intervention. Previously, an administrator needed to create a credential for a data source connection and specify which users and groups could access the credential. The ability to save a credential allows subsequent requests to run tasks that rely on the same data source without requiring the user to re-authenticate. This can also be advantageous for running batch processes.

A new capability, [bibus](#) » [userCapabilityEnum](#) » [canUsePersonalDataSourceCredentials](#) value, has been introduced to allow administrators to determine which users can store their personal data source credentials. By default, users are not allowed to store personal data source credentials in the content store.

A new Content Manager search path function, [“personalDataSourceSignon\(dataSourceName\)”](#) on page 1570, has also been added. This new function returns personal data source credential objects associated with the queried data source for the current user.

This change affects:

- [bibus](#) » [dataSourceCredential](#) class
- [bibus](#) » [account](#) » [dataSourceCredentials](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUsePersonalDataSourceCredentials](#) value
- [“Manage own data source signons”](#) on page 1650
- [“personalDataSourceSignon\(dataSourceName\)”](#) on page 1570

Flexible Scheduling

New schedule types and additional properties provide increased flexibility and options. For example, users and administrators can now set a task scheduled to run on the first Monday of each month to run hourly on that day, in a specified recurrence window within that day, such as between 6 AM and 5 PM.

This provides users with more granular scheduling options that can reduce the overhead of interactive tasks. This can allow administrators to better predict schedule load.

New capabilities also allow administrators to control the scheduling options that are available to users.

See [“Using Intraday Scheduling”](#) on page 71 for more information.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByDay](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByHour](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByMinute](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByMonth](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByTrigger](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByWeek](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingByYear](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [dailyWithIntradayRecurrence](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [monthlyAbsoluteWithIntradayRecurrence](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [monthlyRelativeWithIntradayRecurrence](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [weeklyWithIntradayRecurrence](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [yearlyAbsoluteWithIntradayRecurrence](#) value
- [bibus](#) » [scheduleTypeEnum](#) » [yearlyRelativeWithIntradayRecurrence](#) value
- [“Schedule by day”](#) on page 1662
- [“Schedule by hour”](#) on page 1663
- [“Schedule by minute”](#) on page 1663
- [“Schedule by month”](#) on page 1664
- [“Schedule by trigger”](#) on page 1664
- [“Schedule by week”](#) on page 1665
- [“Schedule by year”](#) on page 1665
- [bibus](#) » [schedule](#) » [intradayRecurrenceEnd](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceInterval](#) property
- [bibus](#) » [schedule](#) » [intradayRecurrenceStart](#) property

Scheduling – Better Administrative Oversight

An additional property was added to the [bibus](#) » [asynchDetailEventRecord](#) class to provide more detailed information about currently running tasks. This allows administrators to identify which server process is running an interactive or background task.

This change affects:

- [bibus](#) » [asynchDetailEventRecord](#) » [cancelledBy](#) property
- [bibus](#) » [asynchDetailEventRecord](#) » [suspendedBy](#) property
- [bibus](#) » [asynchDetailEventRecord](#) » [processID](#) property

Delaying Scheduled Tasks

You can now delay a scheduled task by a specified time duration or until a specific time. This allows both administrators and users to postpone tasks from running at their scheduled times in the event that another dependent task has not yet completed, or due to unforeseen circumstances.

This change affects:

- [event](#) » [delayEventsFor\(eventIDs, for\)](#) method
- [event](#) » [delayEventsUntil\(eventIDs, until\)](#) method
- [event](#) method set

- [bibus](#) » [asynchDetailEventRecord](#) » [scheduledStartTime](#) property

New for Report Specification Schema Version 7.0

Version 7.0 of the Report Specification schema includes new charting functionality. All of the new chart element names are prefaced with "v2_". Many of these elements are updated versions of older chart elements and have the same functionality and name, except for the "v2_" prefix. Reports that contain charts that are authored with legacy chart elements will still run and produce the same results as in the previous release.

The new functionality of the "v2_" elements includes

- a schema that includes a bullet chart, defined by the `v2_bulletChart` element.
- customization of the legend position with the `v2_legendPosition` element.
- customization of how text appears in the legend using the attributes on the `v2_legend` element.
- defining the colors in a chart based on the data values using the `v2_pointColorByValue` element.
- control of the shape of bubbles in a bubble chart using the `changeShapeByCategory` attribute on the `v2_bubbleChart` element.
- control of pie charts, allowing you to customize data labels, pull out pie slices, summarize slices, position pie slices, and customize the appearance of pie charts using the attributes and child elements of the `v2_pieChart` element.
- customization of axis lines using the `v2_axis` element and its child elements.

Support for Axis 1.4

The WSDL file now supports Axis 1.4, which is now used by the default Java toolkit.

A version of the toolkit that uses Axis 1.1 is provided for backward compatibility for Java applications that were created in a previous version of the IBM Cognos Software Development Kit. This toolkit supports all the features included in this release. The Axis 1.1 toolkit is located in *installation_location/sdk-compat/java/lib*.

If you have SDK applications created in a previous version of the IBM Cognos Software Development Kit, you can either use the Axis 1.1 toolkit or upgrade the applications to use the Axis 1.4 toolkit. See [“Upgrading Java Applications for Axis 1.4”](#) on page 122 for information on upgrading Java application to use the Axis 1.4 toolkit.

SDK customers who do not use one of the SOAP toolkits provided with the IBM Cognos Software Development Kit must note the following changes to the WSDL file.

The changes are:

- The elements in some types have been resequenced.
- The attribute `required` has been removed.
- The compositor `xs:sequence` is now used instead of `xs:all`.
- Many elements now have the `minOccurs` attribute specified with value 0.

New Data Integration Service Task Option

A new `dataIntegrationService` task option, the `bibus` » `dataIntegrationTaskOptionEnum` » `clearAuditLogs` value, allows administrators to specify whether audit history information is cleared when tasks are run. When this option is enabled, the execution of metrics-related tasks will delete IBM Cognos Metric Studio audit records, such as usage logging, object updates, and metric value updates.

Clearing data store metric history and calendar data (`bibus` » `dataIntegrationTaskOptionEnum` » `clearHistory` value and `bibus` » `dataIntegrationTaskOptionEnum` » `clearCalendar` value task options) reinitializes the metric package database. After performing these tasks, clear the audit logs to ensure

that data is current and relevant. Additionally, clearing audit logs on a regular basis prevents the audit table from becoming too large.

This change affects:

- [bibus](#) » [dataIntegrationTaskOptionEnum](#) » [clearAuditLogs](#) value

Human Task Service

The human task service creates and manages human tasks. A human task, such as report approval, can be assigned to individuals or groups manually or automatically through the use of other services.

This change affects:

- [bibus](#) » [humanTaskService](#) class
- [bibus](#) » [dispatcher](#) » [humanTaskService](#) property
- [bibus](#) » [installedComponentEnum](#) » [humanTaskService](#) value
- [bibus](#) » [configuration](#) » [htsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [htsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [htsAuditLevel](#) property
- [bibus](#) » [humanTaskService](#) » [htsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [htsCompletedTaskLifetime](#) property
- [bibus](#) » [configurationFolder](#) » [htsCompletedTaskLifetime](#) property
- [bibus](#) » [dispatcher](#) » [htsCompletedTaskLifetime](#) property
- [bibus](#) » [humanTaskService](#) » [htsCompletedTaskLifetime](#) property

Variable Support for Data Movement Tasks

You can now share the specification of a [bibus](#) » [dataMovementTask](#) object by creating one or more [bibus](#) » [dataMovementTaskAlias](#) objects that refer to the [bibus](#) » [dataMovementTask](#).

Each [bibus](#) » [dataMovementTaskAlias](#) has a set of property values (such as information related to scheduling) or parameters that are distinct from the values associated with the [bibus](#) » [dataMovementTask](#) object referenced. This allows the specification for the task to be reused instead of having to copy the task to run with a different set of properties.

Parameter values that are specified in the [parameters](#) property and the [bibus](#) » [dataMovementTaskAlias](#) allow the use of variable values when running a data movement task. These values were previously only available from the IBM Cognos Data Manager command-line interface.

This change affects:

- [bibus](#) » [baseDataMovementTask](#) class
- [bibus](#) » [dataMovementTask](#) class
- [bibus](#) » [dataMovementTaskAlias](#) class
- [bibus](#) » [dashboard](#) » [items](#) property
- [bibus](#) » [folder](#) » [items](#) property
- [bibus](#) » [package](#) » [items](#) property
- [bibus](#) » [eventRecord](#) » [runnable](#) property
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#) property
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#) property
- [bibus](#) » [dataMovementTask](#) » [ownerPassport](#) property

- [bibus](#) » [dataMovementTask](#) » [runAsOwner](#) property

Support for Lineage Requests

The [reportService](#) and [batchReportService](#) services now support report lineage requests on layout expressions that use data items. This feature extends the capability of obtaining lineage information.

For newly saved reports, users can now obtain lineage information on layout expressions that contain data items by issuing a [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) lineage request against [reportService](#) or [batchReportService](#). The [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method must include [bibus](#) » [metadataServiceLineageSpecification](#) class as a specification, and must contain the search path. For unsaved report data, such as during an interactive session, users can now obtain lineage information on report data items by issuing a [secondary report](#) » [lineage\(conversation, parameterValues, options\)](#) request. In both cases, users must forward the response from [reportService](#) (or [batchReportService](#)) to [metadataService](#) for further lineage processing.

SDK users should continue to request lineage information for package data from [metadataService](#) using the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method.

See [Chapter 25, “Using lineage specifications,”](#) on [page 1509](#) for more information.

This change affects:

- [report](#) » [lineage\(conversation, parameterValues, options\)](#) method
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailAsynchSpecification](#) class
- [bibus](#) » [metadataServiceLineageSpecification](#) class
- [report](#) method set

Query Service

The query service has been introduced to support the new IBM Cognos Analytics dynamic query mode, delivering improved query performance for some OLAP data sources.

For more information on supported query modes, see the *Query Modes* feature or the *IBM Cognos IBM Cognos Analytics Administration and Security Guide*.

This change affects:

- [bibus](#) » [queryService](#) class
- [bibus](#) » [dispatcher](#) » [queryService](#) property
- [bibus](#) » [installedComponentEnum](#) » [queryService](#) value
- [bibus](#) » [configuration](#) » [qsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [qsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [qsAuditLevel](#) property
- [bibus](#) » [queryService](#) » [qsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [configurationFolder](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [dispatcher](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [queryService](#) » [qsDiagnosticsEnabled](#) property
- [bibus](#) » [configuration](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [configurationFolder](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [dispatcher](#) » [qsDisableQueryPlanCache](#) property

- [bibus](#) » [queryService](#) » [qsDisableQueryPlanCache](#) property
- [bibus](#) » [configuration](#) » [qsDumpModelToFile](#) property
- [bibus](#) » [configurationFolder](#) » [qsDumpModelToFile](#) property
- [bibus](#) » [dispatcher](#) » [qsDumpModelToFile](#) property
- [bibus](#) » [queryService](#) » [qsDumpModelToFile](#) property
- [bibus](#) » [configuration](#) » [qsIdleConnectionTimeout](#) property
- [bibus](#) » [configurationFolder](#) » [qsIdleConnectionTimeout](#) property
- [bibus](#) » [dispatcher](#) » [qsIdleConnectionTimeout](#) property
- [bibus](#) » [queryService](#) » [qsIdleConnectionTimeout](#) property
- [bibus](#) » [configuration](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [configurationFolder](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [dispatcher](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [queryService](#) » [qsMetricsEnabled](#) property
- [bibus](#) » [configuration](#) » [qsQueryExecutionTrace](#) property
- [bibus](#) » [configurationFolder](#) » [qsQueryExecutionTrace](#) property
- [bibus](#) » [dispatcher](#) » [qsQueryExecutionTrace](#) property
- [bibus](#) » [queryService](#) » [qsQueryExecutionTrace](#) property
- [bibus](#) » [configuration](#) » [qsQueryPlanningTrace](#) property
- [bibus](#) » [configurationFolder](#) » [qsQueryPlanningTrace](#) property
- [bibus](#) » [dispatcher](#) » [qsQueryPlanningTrace](#) property
- [bibus](#) » [queryService](#) » [qsQueryPlanningTrace](#) property

Adaptive Analytics Service

The adaptive analytics service has been integrated into the IBM Cognos Analytics architecture to perform tasks for Analyst Add-in for IBM Cognos Analytic Applications clients.

A specification class and a configuration class have been added to support this service. Configuration properties have also been added to the classes [bibus](#) » [configuration](#), [bibus](#) » [configurationFolder](#), and [bibus](#) » [dispatcher](#).

This change affects:

- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [adaptiveAnalyticsService](#) class
- [bibus](#) » [dispatcher](#) » [adaptiveAnalyticsService](#) property
- [bibus](#) » [installedComponentEnum](#) » [adaptiveAnalyticsService](#) value
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasAffineConnections](#) property
- [bibus](#) » [configuration](#) » [aasAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [aasAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [aasAffineConnections](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasAuditLevel](#) property
- [bibus](#) » [configuration](#) » [aasAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [aasAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [aasAuditLevel](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [aasExecutionTimeLimit](#) property

- [bibus](#) » [configurationFolder](#) » [aasExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [aasExecutionTimeLimit](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [aasMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [aasMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [aasMaximumProcesses](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [aasNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [aasNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [aasNonAffineConnections](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [aasPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [aasPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [aasPeakAffineConnections](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [aasPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [aasPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [aasPeakMaximumProcesses](#) property
- [bibus](#) » [adaptiveAnalyticsService](#) » [aasPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [aasPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [aasPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [aasPeakNonAffineConnections](#) property

Storing Temporary Output Objects Outside the Content Store

The [reportService](#), [batchReportService](#), and [powerPlayService](#) services can now store temporary output objects, such as interactive report data, in the server file system. Previously, temporary output objects were always stored in the content store. This feature improves overall system throughput and performance by reducing the number of content store queries. Applications that frequently run reports interactively will likely benefit most from this change.

Administrators can change the default location using the new [bibus](#) » [configuration](#) » [temporaryObjectLocation](#) property, or using IBM Cognos Administration. IBM Cognos dispatcher deletes the temporary objects after the duration specified in the [bibus](#) » [configuration](#) » [temporaryObjectLifetime](#) property. It is important to specify a duration that ensures interactive report data remain accessible, yet not excessively long such that disk space becomes an issue. The default duration is four hours.

As objects stored in the file system are not available using the SDK, applications should continue to store temporary output objects in the content store by not specifying the [bibus](#) » [runOptionEnum](#) » [outputLocation](#) value option for [reportService](#) objects or the [bibus](#) » [powerPlay8OptionEnum](#) » [outputLocation](#) value option for [powerPlayService](#) objects. This uses the default storage location (content store), thereby allowing SDK applications to retrieve temporary outputs, as in previous releases.

For information on configuration and encryption options, refer to the *IBM Cognos Analytics Installation and Configuration Guide*.

This change affects:

- [bibus](#) » [powerPlay8OptionAnyURI](#) class
- [bibus](#) » [temporaryObjectLocationEnum](#) enumeration set
- [bibus](#) » [powerPlay8OptionEnum](#) » [outputLocation](#) value

- [bibus](#) » [runOptionEnum](#) » [outputLocation](#) value
- [bibus](#) » [configuration](#) » [temporaryObjectLifetime](#) property
- [bibus](#) » [configuration](#) » [temporaryObjectLocation](#) property

IBM Cognos Go! Dashboard Upgrade to IBM Cognos Workspace

Upgrading to the new IBM Cognos Workspace will allow users with the [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewer](#) capability to edit dashboard content. Previously, users required both [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewer](#) and [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewerFileManagement](#) capabilities to modify content. The capability [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewerFileManagement](#) is no longer required and has been deprecated.

Administrators can set whether the Workspace Start Page is invoked when the component is started. By default, the Start Page is displayed at startup; however, users can change this behavior by altering their user preferences in their dashboard.

This change affects:

- [report](#) » [add\(parentPath, object, options\)](#) method
- [report](#) » [update\(object, options\)](#) method
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#) method
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [genericOption](#) class
- [bibus](#) » [genericOptionBoolean](#) class
- [bibus](#) » [dashboardOptionEnum](#) enumeration set
- [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewerFileManagement](#) value
- “Use the Edit Features” on page 1643

IBM Cognos TM1 Packages

Support has been added for packages produced by IBM Cognos TM1.

Users can now invoke TM1 packages from IBM Cognos Connection. This feature allows users to identify TM1 packages, for use with Reporting and Analysis Studio.

This change affects:

- [bibus](#) » [installedComponentEnum](#) » [tm1Studio](#) value
- [bibus](#) » [uiComponentEnum](#) » [tm1Studio](#) value

Relational Metadata Service

Reserved.

This change affects:

- [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [relationalMetadataService](#) class
- [bibus](#) » [dispatcher](#) » [relationalMetadataService](#) property

- [bibus](#) » [userCapabilityEnum](#) » [canImportRelationalMetadata](#) value
- [bibus](#) » [installedComponentEnum](#) » [relationalMetadataService](#) value
- [“Import relational metadata”](#) on page 1649
- [bibus](#) » [configuration](#) » [rmdsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rmdsAffineConnections](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rmdsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [rmdsAuditLevel](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [rmdsExecutionTimeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [rmdsExecutionTimeLimit](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [rmdsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rmdsNonAffineConnections](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rmdsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rmdsPeakAffineConnections](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rmdsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rmdsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rmdsPeakNonAffineConnections](#) property
- [bibus](#) » [relationalMetadataService](#) » [rmdsPeakNonAffineConnections](#) property

Accessibility

A new option was added to control the use of accessibility features in IBM Cognos Analytics.

To enable accessibility functionality on all reports by default, set the [bibus](#) » [systemOptionEnum](#) » [accessibilityFeatures](#) value on the [bibus](#) » [configuration](#) » [serviceDefaultOptions](#) property to `true`. Because options and parameter values can be specified in various places, when you set the [bibus](#) » [systemOptionEnum](#) » [accessibilityFeatures](#) value to `true`, some reports may still run without accessibility features. To force all reports to run with accessibility features enabled, set the [bibus](#) » [systemOptionEnum](#) » [accessibilityFeatures](#) value on both the [bibus](#) » [configuration](#) » [serviceDefaultOptions](#) property and the [bibus](#) » [configuration](#) » [overrideOptions](#) property to `true`.

For more information, see [“How IBM Cognos Determines Search Order When Building a Request”](#) on page 68 and [“Constraining Option Values and Setting Service Default Values”](#) on page 69.

This change affects:

- [bibus](#) » [configuration](#) » [overrideOptions](#) property
- [bibus](#) » [configuration](#) » [serviceDefaultOptions](#) property
- [bibus](#) » [systemOptionEnum](#) enumeration set
- [bibus](#) » [systemOptionEnum](#) » [accessibilityFeatures](#) value

- [bibus » account » useAccessibilityFeatures](#) property
- [“Constraining Option Values and Setting Service Default Values”](#) on page 69
- [“Running an Object Using a Method Call”](#) on page 68
- [“Running an Object Using the Schedule”](#) on page 68

Graphics Service

The graphics service has been introduced to produce graphics (charts) on behalf of the [reportService](#) service.

Charts that are used in reports can be generated in raster, vector, Microsoft® Excel XML, or PDF format.

This change affects:

- [asynch » run\(objectPath, parameterValues, options\)](#) method
- [asynch » runSpecification\(specification, parameterValues, options\)](#) method
- [bibus » graphicsService](#) class
- [bibus » dispatcher » graphicsService](#) property
- [bibus » installedComponentEnum » graphicsService](#) value
- [bibus » configuration » gsAuditLevel](#) property
- [bibus » configurationFolder » gsAuditLevel](#) property
- [bibus » dispatcher » gsAuditLevel](#) property
- [bibus » graphicsService » gsAuditLevel](#) property
- [bibus » configuration » gsExecutionTimeLimit](#) property
- [bibus » configurationFolder » gsExecutionTimeLimit](#) property
- [bibus » dispatcher » gsExecutionTimeLimit](#) property
- [bibus » graphicsService » gsExecutionTimeLimit](#) property
- [bibus » configuration » gsNonAffineConnections](#) property
- [bibus » configurationFolder » gsNonAffineConnections](#) property
- [bibus » dispatcher » gsNonAffineConnections](#) property
- [bibus » graphicsService » gsNonAffineConnections](#) property
- [bibus » configuration » gsPeakNonAffineConnections](#) property
- [bibus » configurationFolder » gsPeakNonAffineConnections](#) property
- [bibus » dispatcher » gsPeakNonAffineConnections](#) property
- [bibus » graphicsService » gsPeakNonAffineConnections](#) property
- [bibus » configuration » gsQueueLimit](#) property
- [bibus » configurationFolder » gsQueueLimit](#) property
- [bibus » dispatcher » gsQueueLimit](#) property
- [bibus » graphicsService » gsQueueLimit](#) property

New PowerPlay Service Methods

The [powerPlayService](#) now supports additional methods.

The methods [report » collectParameterValues\(objectPath, parameterValues, options\)](#) – obsolete and [report » collectParameterValuesSpecification\(specification, parameterValues, options\)](#) – obsolete were moved from the [report](#) method set to the [parameter](#) method set. This change does not affect consumers of the [batchReportService](#) or [reportService](#) services.

This change affects:

- [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) method
- [report](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) – obsolete method
- [report](#) » [getPromptValues\(conversation, parameterValues, options\)](#) – obsolete method
- [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValues\(objectPath, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [getParameters\(objectPath, parameterValues, options\)](#) method
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#) method
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) method
- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) method
- [validate](#) » [validate\(objectPath, parameterValues, options\)](#) method
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) method set
- [promptPaging](#) method set
- [powerPlayService](#) service

Supporting Fail-over for Tree Prompts

The affinity of the [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) method was changed to high to support server fail-over.

This change affects:

- [promptPaging](#) » [getPromptValues\(conversation, parameterValues, options\)](#) method

Content Cleanup Content Manager Task

Authorized users can now create Content Manager tasks to delete unwanted history and report outputs from the content store. Users can specify both the set of objects affected by the cleanup task, and also the rules that determine whether to delete an object.

The new [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentAuditLevel](#) value determines the amount of task history information that the cleanup task records. By default, the history information includes the total number of successfully and unsuccessfully deleted objects per container object (such as a folder or package).

This change affects:

- [bibus](#) » [contentTaskOptionAuditLevelEnum](#) class
- [bibus](#) » [contentTaskOptionRetentionRuleArray](#) class
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContent](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentAuditLevel](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentContext](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentRetentionRules](#) value

Content Pane Improvements

Improvements were made to the content pane in IBM Cognos Analytics Workspace.

These changes allow users to save shortcuts to their favourite objects and references to external resources.

This change affects:

- [bibus](#) » [favoritesFolder](#) class
- [bibus](#) » [mruFolder](#) class
- [bibus](#) » [resource](#) class
- [bibus](#) » [account](#) » [favorites](#) property
- [bibus](#) » [mruFolder](#) » [items](#) property
- [bibus](#) » [namespace](#) » [items](#) property
- [bibus](#) » [namespaceFolder](#) » [items](#) property
- [bibus](#) » [URL](#) » [contentType](#) property

Content Manager Cache Service

The Content Manager cache service was added to the IBM Cognos architecture to cache results of Content Manager queries. Repeated queries to the Content Manager are no longer necessary when the required data is stored in the cache. Implementing the cache service feature can reduce the load on the Content Manager, thereby improving overall system performance. Performance improvements increase with an extended user base using multiple components issuing frequent Content Manager requests, such as running reports and invoking IBM Cognos studios.

This change affects:

- [bibus](#) » [contentManagerCacheService](#) class
- [bibus](#) » [genericOption](#) class
- [bibus](#) » [genericOptionXMLEncodedXML](#) class
- [bibus](#) » [dispatcher](#) » [contentManagerCacheService](#) property
- [bibus](#) » [installedComponentEnum](#) » [contentManagerCacheService](#) value
- [bibus](#) » [configuration](#) » [cmcsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [cmcsAuditLevel](#) property
- [bibus](#) » [contentManagerCacheService](#) » [cmcsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [cmcsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [cmcsHeapLimit](#) property
- [bibus](#) » [configurationFolder](#) » [cmcsHeapLimit](#) property
- [bibus](#) » [contentManagerCacheService](#) » [cmcsHeapLimit](#) property
- [bibus](#) » [dispatcher](#) » [cmcsHeapLimit](#) property

Human Task

The agent infrastructure was extended to allow human tasks to be invoked as an agent task.

A human task is defined as a notification, request, or other user-defined action related to an issue. Human tasks can be assigned to users manually by another user, or automatically. Automatic human tasks are generated as a result of certain conditions being met when a report is run.

This change affects:

- [bibus](#) » [humanTask](#) class
- [bibus](#) » [authoredAgentDefinition](#) » [items](#) property
- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#) property
- [bibus](#) » [eventRecord](#) » [runnable](#) property
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#) property

Portal Skin Management

Administrators with the [bibus](#) » [userCapabilityEnum](#) » [canUsePortalAdministrationTool](#) value capability can use the new Style Management Utility to create customized portal skins (styles) and make them available to end users. As in previous releases, administrators can also manually create a custom style.

Several properties were added to the [bibus](#) » [portalSkin](#) class to improve the management of portal skins, and preserve predefined custom skins during IBM Cognos component upgrades.

This change affects:

- [bibus](#) » [portalSkin](#) » [base](#) property
- [bibus](#) » [portalSkin](#) » [published](#) property
- [bibus](#) » [portalSkin](#) » [specification](#) property

Multi-Instance IBM Cognos Connection

Support has been added to allow IBM Cognos Connection to reference content in other installations of IBM Cognos Analytics.

This makes multi-version coexistence possible. Customers can stage their upgrade process by allowing two versions of IBM Cognos Analytics to be accessed through a single portal, preventing disruptions to end users.

For example, users can access IBM Cognos Analytics Version 8.4 or IBM Cognos Analytics Version 8.3 content from within the IBM Cognos Connection Version 10.1.0 portal.

You can use this feature to provide access to content through IBM Cognos Connection from multiple installations of IBM Cognos Analytics Version 10.1.0 in a large enterprise.

This change affects:

- [bibus](#) » [launchable](#) class
- [bibus](#) » [content](#) » [items](#) property

Personal Packages

Users can now create personalized packages based on SAP BW or PowerCube data sources directly from IBM Cognos Connection. Using IBM Cognos Framework Manager to import, publish, and synchronize data is no longer required. Administrators can enable this functionality for specific data sources in their installations by setting the [capabilities](#) property.

Users must have the [bibus](#) » [userCapabilityEnum](#) » [canUseSelfServicePackageWizard](#) capability to create personal packages.

This change affects:

- [bibus](#) » [dataSource](#) » [capabilities](#) property
- [bibus](#) » [dataSourceCapabilityEnum](#) enumeration set

- [bibus](#) » [userCapabilityEnum](#) » [canUseSelfServicePackageWizard](#) value
- [bibus](#) » [dataSourceCapabilityEnum](#) » [personalPackages](#) value
- “Self Service Package Wizard” on [page 1667](#)

Launchable

The [bibus](#) » [launchable](#) class is a generic class that allows arbitrary content to be stored in the content store.

Annotation Service

The annotation service stores and manages comments or additional textual information that are added to an IBM Cognos object (such as a report element or a task) against the context of a report, report widget, or report element. These are made accessible through IBM Cognos Workspace.

The annotation service also manages comments against tasks accessible through **My Inbox** in IBM Cognos Connection. These comments persist across versions of a report.

This change affects:

- [bibus](#) » [annotationService](#) class
- [bibus](#) » [dispatcher](#) » [annotationService](#) property
- [bibus](#) » [installedComponentEnum](#) » [annotationService](#) value
- [bibus](#) » [annotationService](#) » [ansAnnotationLifetime](#) property
- [bibus](#) » [configuration](#) » [ansAnnotationLifetime](#) property
- [bibus](#) » [configurationFolder](#) » [ansAnnotationLifetime](#) property
- [bibus](#) » [dispatcher](#) » [ansAnnotationLifetime](#) property
- [bibus](#) » [annotationService](#) » [ansAuditLevel](#) property
- [bibus](#) » [configuration](#) » [ansAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [ansAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [ansAuditLevel](#) property

Visual Basic 6.0 Support

The IBM Cognos Software Development Kit no longer supports the Visual Basic 6.0 language. For a list of supported software environments, see the IBM Cognos Software Development Kit pages at IBM Cognos Customer Center. Please migrate your applications to one of the supported environments.

CAM Passport Changes

The structure of the `cam_passport` cookie has changed and its contents may change during a session. SDK applications that assume that the value of the `cam_passport` cookie is the same as the value of the [bibus](#) » [CAMPassport](#) » [id](#) property, will no longer work and will have to be modified.

See “Changes to the `cam_passport` cookie” on [page 123](#) for more information.

This change affects:

- [bibus](#) » [CAMSettings](#) class
- [bibus](#) » [CAMPassport](#) » [canCallLogon](#) property
- [bibus](#) » [CAMPassport](#) » [generation](#) property
- [bibus](#) » [CAMPassport](#) » [isAnonymous](#) property
- [bibus](#) » [CAMPassport](#) » [logEnabled](#) property

- [bibus](#) » [CAM](#) » [settings](#) property

Statistics Service

The statistics service has been introduced to support IBM Cognos Statistics.

Statistics allows IBM Cognos Analytics - Reporting users to include statistical objects, such as descriptive tables and histograms, in their reports.

The `canUseDescriptiveStatistics` capability allows administrators to determine who can use statistical objects when authoring reports in Reporting.

This change affects:

- [“Statistics” on page 1668](#)

Use Report Name for Output File Name

When you run a report in an output format such as PDF, delimited text (CSV), or in a Microsoft® Excel (XLS) spreadsheet, the report name is now used as the file name. This makes it easier for users to save the report output using the same name as the original report.

The report name is based on the value of the report specification element `reportName`. If a value cannot be determined, "New Report", with an appropriate extension, is used by default.

When the `report » query(objectPath, parameterValues, options)` method is called, the `defaultName` property is now used to update the `reportName` element in the report specification.

The element `reportName` is in Version 7.0 of the report specification schema.

This change affects:

- `report » query(objectPath, parameterValues, options)` method

Query Modes

In addition to the compatible query mode supported in earlier releases, IBM Cognos Analytics now supports dynamic query mode, which offers improved query performance for some OLAP data sources.

When a model is published, the selected query mode is reflected by the `bibus » queryOptionEnum » mode` value option in the `bibus » model » options` property of the published model instance. This option defines how IBM Cognos Analytics processes queries using the model. The default query mode is compatible query mode.

When a model is republished and the query mode is changed, existing reports might need to be updated to use the newly published model to leverage the new query mode, depending on whether the version of the model that the report references is still available in the content store.

For more information on dynamic query mode and its supported data sources, see the IBM Cognos Analytics *Dynamic Query Guide*.

This change affects:

- `dataSource » testDataSourceConnection(connectionString, credentials)` method
- `bibus » genericOption` class
- `bibus » genericOptionAnyURI` class
- `bibus » session » items` property
- `bibus » model » options` property
- `bibus » queryModeEnum` enumeration set

- [bibus](#) » [queryOptionEnum](#) enumeration set
- [bibus](#) » [authoredReport](#) » [queryMode](#) property
- [queryService](#) service

Production of Excel 2000 Format Report Outputs is Obsolete

The [bibus](#) » [outputFormatEnum](#) » [XLS](#) value and [bibus](#) » [outputFormatEnum](#) » [singleXLS](#) value output formats are deprecated in this release. These formats were previously represented in the portal as **Excel 2000** and **Excel 2000 Single Sheet**, respectively.

Run requests specifying either of these formats will fail indicating that the output format is not supported. For backwards compatibility, reports specifying these formats in the content store are still viewable.

During upgrade or deployment, properties where [bibus](#) » [outputFormatEnum](#) can be specified are checked for either of these values and automatically changed to [bibus](#) » [outputFormatEnum](#) » [XLWA](#) value. See “Changes to the Microsoft® Excel Output Formats” on page 124 for more information.

This change affects:

- [bibus](#) » [outputFormatEnum](#) » [singleXLS](#) value
- [bibus](#) » [outputFormatEnum](#) » [XLS](#) value

Support for .NET 1.1 Removed

The .NET Framework libraries included with the IBM Cognos Software Development Kit no longer support the .NET Framework, version 1.1. The `cognosdotnet.dll` and `cognosdotnetassembly.dll` assemblies are no longer included.

To create .NET applications using the .NET Framework toolkit, use the .NET Framework, version 2.0 or higher.

IBM Cognos Content Manager/Enterprise Content Management Integration

Reserved.

This change affects:

- [bibus](#) » [repositoryRule](#) class
- [bibus](#) » [account](#) » [repositoryRules](#) property
- [bibus](#) » [content](#) » [repositoryRules](#) property
- [bibus](#) » [folder](#) » [repositoryRules](#) property
- [bibus](#) » [namespace](#) » [repositoryRules](#) property
- [bibus](#) » [namespaceFolder](#) » [repositoryRules](#) property
- [bibus](#) » [package](#) » [repositoryRules](#) property
- [bibus](#) » [dataSourceCapabilityEnum](#) enumeration set
- [bibus](#) » [userCapabilityEnum](#) » [canUpdateRepositoryRules](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseRepository](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canViewContentInRepository](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentDoNotWriteToRepository](#) value
- [bibus](#) » [contentTaskOptionEnum](#) » [cleanupContentUpdateContextObjects](#) value
- [bibus](#) » [systemMetricEnum](#) » [failedExternalizedDocumentsPercent](#) value

- [bibus](#) » [systemMetricEnum](#) » [numberOfFailedExternalizedDocuments](#) value
- [bibus](#) » [systemMetricEnum](#) » [numberOfSuccessfulExternalizedDocuments](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [preserveStoreIDs](#) value
- [bibus](#) » [dataSourceCapabilityEnum](#) » [repository](#) value
- [bibus](#) » [installedComponentEnum](#) » [repositoryIntegration](#) value
- [bibus](#) » [systemMetricEnum](#) » [successfulExternalizedDocumentsPercent](#) value
- “[Manage repository connections](#)” on page 1644
- “[View external documents](#)” on page 1645
- “[External Repositories](#)” on page 1644

Object Updates

The initial value of the [iconURI](#) property for several content store objects have been updated.

This change affects:

- “[Execute Indexed Search](#)” on page 1640
- “[Executive Dashboard](#)” on page 1640
- “[Glossary](#)” on page 1648

Package Capabilities Overwritten on Default Import/Export

User capability policies in the content store can now be replaced with the user capability policies defined in corresponding objects when importing a deployment archive. This change has been introduced to make deployment behaviour more consistent with existing handling of object security policies during import and export operations. In previous releases, when you specified that object policies should be included when importing a deployment, user capability policies were not included. This improvement also avoids the need for administrators to update user capability policies manually.

By default, identified object and user capability policies are not imported during the deployment. Administrators can change this behaviour by setting [bibus](#) » [deploymentOptionEnum](#) » [objectPolicies](#) value to `true`.

This change affects:

- [bibus](#) » [deploymentOptionEnum](#) » [objectPolicies](#) value

IBM Cognos TM1 and IBM Cognos Workspace Integration

You can now use content from IBM Cognos TM1 a dashboard object in IBM Cognos Workspace.

A new enumeration ([dataSourceCapabilityEnum](#) » [atomContentProvider](#) value) allows an administrator to configure IBM Cognos TM1 as an atom feed content provider.

This change affects:

- [bibus](#) » [dataSourceCapabilityEnum](#) enumeration set
- [bibus](#) » [dataSourceCapabilityEnum](#) » [atomContentProvider](#) value

Collaboration Tool Integration

IBM Cognos and IBM Cognos Workspace can now integrate with IBM Connections for collaborative decision-making.

The [bibus » configuration » collaborationDiscoveryURI](#) property has been added to specify the discovery URI of the IBM Connections server. New capabilities and security objects have also been added to provide administrators with control over which users, roles, and groups can access this functionality.

This change affects:

- [bibus » userCapabilityEnum » canCollaborate](#) value
- [bibus » userCapabilityEnum » canLaunchCollaborationTools](#) value
- [bibus » userCapabilityEnum » canUseCollaborationFeatures](#) value
- “Allow collaboration features” on page 1635
- “Launch collaboration tools” on page 1636
- “Collaborate” on page 1635
- [bibus » configuration » collaborationDiscoveryURI](#) property

New Email Configuration Parameters

Configuration parameters can now limit the size of email attachments and email messages. These changes allow administrators to configure their systems to avoid potential problems with abnormally large items.

The configuration parameters [asMaximumEMailAttachmentSize](#), [brsMaximumEMailAttachmentSize](#), [ppsMaximumEMailAttachmentSize](#), and [rsMaximumEMailAttachmentSize](#) were added to the [agentService](#) service, [batchReportService](#) service, [powerPlayService](#) service, and the [reportService](#) service respectively to limit the size of attachments these services can send to the [deliveryService](#) service. If very large email attachments are causing memory errors on the server that hosts the [deliveryService](#) service, the values of these configuration parameters may need to be adjusted. If the uncompressed size of an email attachment exceeds the specified configuration parameter, an error message is generated in the email message in place of the attachment indicating that the size of the attachment has exceeded the configured limit.

The new configuration parameter [dsMaximumEMailSize](#), added to the [deliveryService](#) service, limits the total size of an email. Your mail server configuration determines the value for this configuration parameter. If the uncompressed size of the email exceeds the specified configuration parameter value, the [deliveryService](#) service removes the largest attachments from the message until the total size of the message is lower than the specified configuration parameter value. If an attachment is removed, an error message is created and added as a plain text attachment to the email indicating that the message size has exceeded the configured limit. If the uncompressed size of the email body (for example, an HTML report) exceeds the configured limit, the error message replaces the content of the body.

This change affects:

- [bibus » agentService » asMaximumEMailAttachmentSize](#) property
- [bibus » configuration » asMaximumEMailAttachmentSize](#) property
- [bibus » configurationFolder » asMaximumEMailAttachmentSize](#) property
- [bibus » dispatcher » asMaximumEMailAttachmentSize](#) property
- [bibus » batchReportService » brsMaximumEMailAttachmentSize](#) property
- [bibus » configuration » brsMaximumEMailAttachmentSize](#) property
- [bibus » configurationFolder » brsMaximumEMailAttachmentSize](#) property
- [bibus » dispatcher » brsMaximumEMailAttachmentSize](#) property
- [bibus » configuration » dsMaximumEMailSize](#) property
- [bibus » configurationFolder » dsMaximumEMailSize](#) property
- [bibus » deliveryService » dsMaximumEMailSize](#) property
- [bibus » dispatcher » dsMaximumEMailSize](#) property

- [bibus](#) » [configuration](#) » [ppsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [configurationFolder](#) » [ppsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [dispatcher](#) » [ppsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [powerPlayService](#) » [ppsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [configuration](#) » [rsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [configurationFolder](#) » [rsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [dispatcher](#) » [rsMaximumEMailAttachmentSize](#) property
- [bibus](#) » [reportService](#) » [rsMaximumEMailAttachmentSize](#) property

IBM Cognos Active Report

This release includes support for a new report type that allows users to work offline with their data.

An IBM Cognos Active Report ([bibus](#) » [interactiveReport](#) class) allows users to work offline with a local copy of their data. They can explore and analyze data, make changes to their reports, and filter and sort data while disconnected from the network. When they re-connect, they can synchronize their changes with the live data source.

See the IBM Cognos Analytics - Reporting *User Guide* for more information.

This change affects:

- [report](#) » [add\(parentPath, object, options\)](#) method
- [report](#) » [update\(object, options\)](#) method
- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [collectParameterValuesSpecification\(specification, parameterValues, options\)](#) method
- [parameter](#) » [getParametersSpecification\(specification, parameterValues, options\)](#) method
- [validate](#) » [validate\(objectPath, parameterValues, options\)](#) method
- [validate](#) » [validateSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [interactiveReport](#) class
- [bibus](#) » [reportServiceInteractiveReportSpecification](#) class
- [bibus](#) » [specificationFormatEnum](#) » [interactiveReport](#) value

Query Service Administration Task

Query service administration tasks are now supported.

IBM Cognos Administration allows access to maintenance tasks administrators related to the query service, such as cache cleaning. Caching leverages previously executed query results for reuse and, when possible, avoids new queries to the database. Caching can improve performance when

- reports are re-run with small modifications
- analyses are performed within the same cube
- repetitive master-detail requests are performed for large reports

Because query service caches can become stale when the underlying data or metadata is refreshed, a clear cache task can be scheduled and targeted at a specific cube or relational data source according to the refresh rate of the underlying data.

See the *IBM Cognos IBM Cognos Analytics Dynamic Query Guide* and the *IBM Cognos IBM Cognos Analytics Administration and Security Guide* for more information.

This change affects:

- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [genericOptionStringArray](#) class
- [bibus](#) » [queryServiceTask](#) class
- [bibus](#) » [adminFolder](#) » [items](#) property
- [bibus](#) » [asynchDetailEventRecord](#) » [runnable](#) property
- [bibus](#) » [eventRecord](#) » [runnable](#) property
- [bibus](#) » [jobStepDefinition](#) » [stepObject](#) property
- [bibus](#) » [agentTaskDefinition](#) » [taskObject](#) property
- [bibus](#) » [queryTaskOptionEnum](#) enumeration set
- [bibus](#) » [userCapabilityEnum](#) » [canUseQueryServiceTool](#) value
- [“Query Service Administration” on page 1626](#)

Batch Report Service/Report Service Optimizations

Enhancements have been added to the [reportService](#) service and [batchReportService](#) service to optimize interaction between IBM Cognos Analytics and IBM Cognos Workspace.

These changes provide application developers with the opportunity to improve performance by providing a new run option ([bibus](#) » [runOptionEnum](#) » [returnOutputWhenAvailable](#) value) that can reduce the number of method calls required to retrieve report output.

By setting [returnOutputWhenAvailable](#) to `true`, your application does not have to call the [report](#) » [getOutput\(conversation, parameterValues, options\)](#) method to retrieve report output from the [reportService](#) service. Output, if available, can be obtained directly from the [bibus](#) » [asynchReply](#) » [details](#) property returned in the [wait\(conversation, parameterValues, options\)](#) » [result](#) return value of the [asynch](#) » [wait\(conversation, parameterValues, options\)](#) method instead.

Existing applications will continue to work as they did in previous releases and do not require source code changes as a result of this new option.

It is also now possible to override the default styles used to render a report by specifying a value in the [useStyleVersion](#).

This change affects:

- [bibus](#) » [asynchDetailAsynchSpecification](#) class
- [bibus](#) » [specificationOptionString](#) class
- [bibus](#) » [runOptionEnum](#) » [returnOutputWhenAvailable](#) value
- [bibus](#) » [specificationOptionEnum](#) » [useStyleVersion](#) value
- [“Optimizing the Asynchronous Conversation” on page 79](#)

Data Collection Options for Indexing Tasks

There are new data collection options for indexing tasks.

These new options provide greater flexibility for administrators in defining the scope of individual indexing tasks.

This change affects:

- [bibus](#) » [indexOptionEnum](#) » [indexModeledData](#) value
- [bibus](#) » [indexOptionEnum](#) » [indexObjectTypes](#) value
- [bibus](#) » [indexOptionEnum](#) » [indexReferencedData](#) value

JMX Proxy Server Scalability

Reserved.

This change affects:

- [bibus](#) » [configuration](#) » [jmxProxyHostDispatchers](#) property
- [bibus](#) » [configuration](#) » [activeJMXProxyURI](#) property

Updated Support for IBM Cognos Express

Support has been updated for IBM Cognos Express.

A new service, configurable through the [bibus](#) » [dataAdvisorService](#), has been added to provide support for IBM Cognos Express Data Advisor clients.

A new MIME type has also been added to the [bibus](#) » [launchable](#) » [launchableType](#) property to support **canvas** objects within the IBM Cognos Express Advisor user interface.

This change affects:

- [bibus](#) » [dataAdvisorService](#) class
- [bibus](#) » [dispatcher](#) » [dataAdvisorService](#) property
- [bibus](#) » [installedComponentEnum](#) » [dataAdvisorService](#) value
- [bibus](#) » [configuration](#) » [dasAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [dasAuditLevel](#) property
- [bibus](#) » [dataAdvisorService](#) » [dasAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [dasAuditLevel](#) property
- [bibus](#) » [launchable](#) » [launchableType](#) property

Run with Owner Capabilities

Reserved.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) enumeration set
- [bibus](#) » [authoredReport](#) » [runWithOwnerCapabilities](#) property
- [bibus](#) » [dataMovementTask](#) » [runWithOwnerCapabilities](#) property

Old Drill-through API Removed

The following deprecated methods that supported drill-through operations have been removed:

- `determineDrillThroughTargetParameterValues(objectPaths, sourceContext, sourceContextValues, targetParameterAssignments, targetParameters, parameterValues, options)`
- `getDrillThroughPaths(objectPath, parameterValues, options)`
- `getDrillThroughTargetParameterValues(objectPath, sourceContextValues, parameterValues, options)`

The unused drill-through method, `getDrillThroughDefaultParameterAssignments(objectPath, parameterValues, options)`, was also removed.

Developers can no longer use these methods in their applications. Refer to the "What's new" description in the affected methods to determine what methods should be used instead.

This change affects:

- [drillThrough](#) method set

Integration of Adaptive Analytics and Dashboards

To facilitate integration of Analyst Add-in for IBM Cognos Analytic Applications reports into IBM Cognos Workspace, Adaptive Analytics publishes reports with an option of type `bibus » genericOptionXMLEncodedXML` with the `name` value of `http://developer.cognos.com/ceba/constants/serviceProvider#AA_Info`.

Object Documentation Updates

Documentation related to the initial values of a number of content store object properties has been corrected.

This change affects:

- [“Anonymous”](#) on page 1690
- [“User Profile”](#) on page 1681
- [“Planning Contributor”](#) on page 1654
- [“My Folders”](#) on page 1692
- [“My Folders”](#) on page 1682
- [“All Authenticated Users”](#) on page 1689
- [“Everyone”](#) on page 1698
- [“Most Recently Used list”](#) on page 1691
- [“Most Recently Used list”](#) on page 1681
- [“Cognos”](#) on page 1687
- [“<Indeterminate>”](#) on page 1684
- [“11x17”](#) on page 1674
- [“A3”](#) on page 1674
- [“A4”](#) on page 1675
- [“B4 JIS”](#) on page 1676
- [“B5 JIS”](#) on page 1676
- [“Legal”](#) on page 1677
- [“Letter”](#) on page 1677
- [“Capabilities”](#) on page 1723
- [“Content Administration”](#) on page 1713
- [“Current Activities”](#) on page 1725
- [“Data Source Connections”](#) on page 1714
- [“Dispatchers and Services”](#) on page 1714
- [“Distribution Lists and Contacts”](#) on page 1715
- [“Pagelets”](#) on page 1712
- [“Configuration”](#) on page 1713
- [“Pages”](#) on page 1746

- [“Console” on page 1712](#)
- [“Index Search” on page 1717](#)
- [“Security” on page 1722](#)
- [“Status” on page 1725](#)
- [“Index” on page 1718](#)
- [“Past Activities” on page 1726](#)
- [“Portlets” on page 1715](#)
- [“PowerPlay” on page 1722](#)
- [“Printers” on page 1716](#)
- [“Schedules” on page 1727](#)
- [“Search” on page 1718](#)
- [“Storage” on page 1719](#)
- [“Styles” on page 1717](#)
- [“System” on page 1727](#)
- [“Upcoming Activities” on page 1728](#)
- [“User Interface Profiles” on page 1723](#)
- [“Users, Groups, and Roles” on page 1724](#)
- [“Administration” on page 1711](#)
- [“Connection” on page 1746](#)
- [“Business” on page 1757](#)
- [“Classic” on page 1757](#)
- [“Contemporary” on page 1758](#)
- [“Corporate” on page 1758](#)
- [“Styles” on page 1756](#)
- [“Modern” on page 1758](#)
- [“Presentation” on page 1759](#)
- [“Bookmarks Viewer” on page 1754](#)
- [“Capabilities” on page 1730](#)
- [“Content Administration” on page 1730](#)
- [“Current Activities” on page 1731](#)
- [“Data Source Connections” on page 1732](#)
- [“Dispatchers and Services” on page 1733](#)
- [“Distribution Lists and Contacts” on page 1734](#)
- [“Portlets” on page 1728](#)
- [“Portlets” on page 1747](#)
- [“Console” on page 1729](#)
- [“HTML Source” on page 1754](#)
- [“HTML Viewer” on page 1755](#)
- [“IBM Cognos Enhanced Search - command panel” on page 1740](#)
- [“IBM Cognos Enhanced Search - External Search Results” on page 1741](#)
- [“IBM Cognos Enhanced Search - main UI” on page 1741](#)
- [“IBM Cognos Enhanced Search - Refinement Viewer” on page 1742](#)
- [“IBM Cognos Enhanced Search - Results Viewer” on page 1742](#)

- [“IBM Cognos Extended Applications Portlet” on page 1751](#)
- [“IBM Cognos History Chart” on page 1752](#)
- [“IBM Cognos Metric List” on page 1753](#)
- [“IBM Cognos Navigator” on page 1749](#)
- [“IBM Cognos Search” on page 1749](#)
- [“IBM Cognos Viewer” on page 1750](#)
- [“Image Viewer” on page 1755](#)
- [“Index” on page 1743](#)
- [“Multi-page” on page 1748](#)
- [“Past Activities” on page 1735](#)
- [“Portlets” on page 1735](#)
- [“PowerPlay” on page 1736](#)
- [“Printers” on page 1736](#)
- [“cogadmin” on page 1729](#)
- [“Dashboard” on page 1747](#)
- [“IBM Cognos Content” on page 1748](#)
- [“IBM Cognos Extended Applications” on page 1750](#)
- [“IBM Cognos Go! Search” on page 1740](#)
- [“IBM Cognos Go! Search Admin” on page 1743](#)
- [“IBM Cognos Metric Studio” on page 1751](#)
- [“IBM Cognos Utility” on page 1753](#)
- [“Profiles Administration” on page 1737](#)
- [“RSS Viewer” on page 1756](#)
- [“Schedules” on page 1737](#)
- [“Search” on page 1744](#)
- [“Storage” on page 1744](#)
- [“Styles” on page 1738](#)
- [“System” on page 1738](#)
- [“Upcoming Activities” on page 1739](#)
- [“Users Groups and Roles” on page 1739](#)
- [“Adaptive Analytics Administrators” on page 1688](#)
- [“Adaptive Analytics Users” on page 1688](#)
- [“Analysis Users” on page 1689](#)
- [“Authors” on page 1693](#)
- [“Consumers” on page 1694](#)
- [“Controller Administrators” on page 1695](#)
- [“Controller Users” on page 1696](#)
- [“Data Manager Authors” on page 1696](#)
- [“Directory Administrators” on page 1697](#)
- [“Express Authors” on page 1698](#)
- [“Metrics Administrators” on page 1700](#)
- [“Metrics Authors” on page 1700](#)
- [“Metrics Users” on page 1701](#)

- [“Planning Contributor Users” on page 1703](#)
- [“Planning Rights Administrators” on page 1703](#)
- [“Portal Administrators” on page 1704](#)
- [“PowerPlay Administrators” on page 1705](#)
- [“PowerPlay Users” on page 1705](#)
- [“Query Users” on page 1706](#)
- [“Readers” on page 1707](#)
- [“Report Administrators” on page 1707](#)
- [“Server Administrators” on page 1708](#)
- [“System Administrators” on page 1708](#)
- [“Adaptive Analytics Administration” on page 1618](#)
- [“Administration tasks” on page 1618](#)
- [“Advanced” on page 1656](#)
- [“Allow External Data” on page 1657](#)
- [“Bursting” on page 1658](#)
- [“Distribution Lists and Contacts” on page 1621](#)
- [“Printers” on page 1625](#)
- [“Configure and manage the system” on page 1619](#)
- [“Context Menu” on page 1632](#)
- [“Controller Administration” on page 1620](#)
- [“Create” on page 1656](#)
- [“Create/Delete” on page 1659](#)
- [“Data Source Connections” on page 1620](#)
- [“HTML Items in Report” on page 1659](#)
- [“Metric Studio Administration” on page 1622](#)
- [“Edit View” on page 1651](#)
- [“Open PowerPlay Reports with Analysis Studio” on page 1630](#)
- [“Open PowerPlay Reports with Reporting” on page 1660](#)
- [“Planning Administration” on page 1624](#)
- [“PowerPlay Servers” on page 1625](#)
- [“Run activities and schedules” on page 1626](#)
- [“Run With Options” on page 1633](#)
- [“Scheduling Priority” on page 1665](#)
- [“Selection” on page 1634](#)
- [“Set capabilities and manage UI profiles” on page 1627](#)
- [“Styles and portlets” on page 1628](#)
- [“Toolbar” on page 1634](#)
- [“User Defined SQL” on page 1661](#)
- [“Users, Groups, and Roles” on page 1629](#)
- [“Adaptive Analytics” on page 1616](#)
- [“Administration” on page 1617](#)
- [“Analysis Studio” on page 1629](#)
- [“Data Manager” on page 1637](#)

- [“Cognos Viewer” on page 1631](#)
- [“Controller Studio” on page 1636](#)
- [“Detailed Errors” on page 1637](#)
- [“Drill Through Assistant” on page 1638](#)
- [“Event Studio” on page 1639](#)
- [“EVStudio” on page 1639](#)
- [“Execute Indexed Search” on page 1640](#)
- [“Executive Dashboard” on page 1640](#)
- [“Glossary” on page 1648](#)
- [“Hide Entries” on page 1648](#)
- [“Lineage” on page 1650](#)
- [“Metric Studio” on page 1651](#)
- [“Package Data Sources” on page 1653](#)
- [“PowerPlay Studio” on page 1655](#)
- [“Query Studio” on page 1655](#)
- [“Reporting” on page 1657](#)
- [“Scheduling” on page 1661](#)
- [“Set Entry-Specific Capabilities” on page 1667](#)
- [“Specification Execution” on page 1668](#)
- [“Statistics” on page 1668](#)
- [“Watch Rules” on page 1669](#)
- [“My Watch Items” on page 1692](#)
- [“My Watch Items” on page 1682](#)
- [“Express - deprecated” on page 1679](#)
- [“Reporting Profiles - deprecated” on page 1678](#)
- [“User Interface Profiles - deprecated” on page 1678](#)
- [“Professional - deprecated” on page 1680](#)

Object Updates

Some of the initial content store objects have been updated for this release.

This change affects:

- [“PowerPlay” on page 1722](#)
- [“IBM Cognos Extended Applications Portlet” on page 1751](#)
- [“IBM Cognos History Chart” on page 1752](#)
- [“IBM Cognos Metric List” on page 1753](#)
- [“IBM Cognos Navigator” on page 1749](#)
- [“IBM Cognos Search” on page 1749](#)
- [“PowerPlay” on page 1736](#)
- [“IBM Cognos Content” on page 1748](#)
- [“IBM Cognos Extended Applications” on page 1750](#)
- [“IBM Cognos Metric Studio” on page 1751](#)
- [“IBM Cognos Utility” on page 1753](#)

Previously Reserved Index Options Now Available for Client Use

Index options that were previously marked as "Reserved" are now available for client use.

This change affects:

- [bibus](#) » [indexOptionEnum](#) » [excludedObjects](#) value
- [bibus](#) » [indexOptionEnum](#) » [force](#) value
- [bibus](#) » [indexOptionEnum](#) » [includedObjects](#) value

Object Capabilities Properties

You can now retrieve several read-only properties related to object capabilities with only `traverse` permission on the full path to the containing object. Read permission on the object is no longer required.

This change allows users who do not have read permission on an object to use the package and perform such tasks as running reports.

This change affects:

- [bibus](#) » [content](#) » [effectiveUserCapabilities](#) property
- [bibus](#) » [folder](#) » [effectiveUserCapabilities](#) property
- [bibus](#) » [package](#) » [effectiveUserCapabilities](#) property
- [bibus](#) » [content](#) » [userCapabilities](#) property
- [bibus](#) » [folder](#) » [userCapabilities](#) property
- [bibus](#) » [package](#) » [userCapabilities](#) property

New Limitations for Setting Object Ownership

Changes in this release limit the ability to set the [bibus](#) » [baseClass](#) » [owner](#) property.

Content Manager now prevents the [bibus](#) » [baseClass](#) » [owner](#) property from being set to a value other than an account referenced by the passport used in the request.

System administrators can continue to assign ownership to any account.

Deployment functionality is unchanged.

If your application sets the [bibus](#) » [baseClass](#) » [owner](#) property to a value other than an account referenced by the passport used in the request, you need to make changes to avoid this scenario.

This change affects:

- [bibus](#) » [baseClass](#) » [owner](#) property

Administration Capabilities for Background Job Queue Maintenance

Methods that manage events that are not owned by the security context of the request now require the caller to have the [canUseMonitorActivityTool](#) value capability instead of the [canUseAdministrationPortal](#) value capability.

Applications that check for these capabilities when performing event management tasks must be updated to reflect this change.

This change affects:

- [event](#) » [cancelEvent\(eventID\)](#) method
- [event](#) » [cancelEvents\(eventIDs\)](#) method
- [event](#) » [holdEvent\(eventID\)](#) method
- [event](#) » [holdEvents\(eventIDs\)](#) method
- [event](#) » [releaseEvent\(eventID\)](#) method
- [event](#) » [releaseEvents\(eventIDs\)](#) method
- [event](#) » [scheduleEvent\(eventID\)](#) method
- [event](#) » [scheduleEvents\(eventIDs\)](#) method
- [event](#) » [trigger\(triggerName\)](#) method
- [event](#) » [updateEvents\(events\)](#) method

Changes to JSP Tags in IBM Cognos Portal Services

The JSP tag, <Cognos8Connect>, used in IBM Cognos Portal Services, is deprecated and will be removed in a future version of the product. It has been changed to <IBMCognosConnect>.

Backwards compatibility is maintained for the <Cognos8Connect> tag for this version of the product, however, you should upgrade your portal applications to use the new tag.

This change affects:

- [IBMCognosConnect](#)

Documentation Updates

Several miscellaneous bug fixes were applied in this release.

This change affects:

- [bibus](#) » [systemMetricThresholds](#) class
- [bibus](#) » [output](#) » [pages](#) property
- “<Known User>” on [page 1684](#)
- “My Folders” on [page 1686](#)
- “Most Recently Used List” on [page 1686](#)
- “<Indeterminate>” on [page 1691](#)
- “<Indeterminate>” on [page 1685](#)
- “My Watch Items” on [page 1687](#)
- [bibus](#) » [output](#) » [lastPage](#) property
- [bibus](#) » [baseAgentDefinition](#) » [ownerPassport](#) property
- [bibus](#) » [baseDataIntegrationTask](#) » [ownerPassport](#) property
- [bibus](#) » [baseDataMovementTask](#) » [ownerPassport](#) property
- [bibus](#) » [baseReport](#) » [ownerPassport](#) property
- [bibus](#) » [schedule](#) » [priority](#) property
- [bibus](#) » [baseParameter](#) » [promptType](#) property

Appendix H. New in Version 9.0.0

This appendix provides information about the new features in Version 9.0.0 of the product.

Support for IBM Cognos Express

Support for IBM Cognos Express has been added to the IBM Cognos Software Development Kit.

This change affects:

- [bibus](#) » [EVService](#) class
- [bibus](#) » [launchable](#) class
- [bibus](#) » [dispatcher](#) » [EVService](#) property
- [bibus](#) » [folder](#) » [items](#) property
- [bibus](#) » [package](#) » [items](#) property
- [bibus](#) » [shortcut](#) » [target](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUseEV](#) value
- [bibus](#) » [installedComponentEnum](#) » [EVService](#) value
- [bibus](#) » [installedComponentEnum](#) » [EVStudio](#) value
- [bibus](#) » [uiComponentEnum](#) » [EVStudio](#) value
- [“Public Folders”](#) on page 1759
- [“EVStudio”](#) on page 1639
- [bibus](#) » [configuration](#) » [evsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [evsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [evsAuditLevel](#) property
- [bibus](#) » [EVService](#) » [evsAuditLevel](#) property

Appendix I. New in Version 8.4 GA

This appendix provides information about the new features in Version 8.4 GA of the product.

Statistics

Support for statistics has been incorporated into Analysis Studio and Reporting.

This feature uses a third-party application to generate statistical output including charts and tables that can be used within Cognos Analytics applications.

A new capability and new security object was added to define the set of users who can use the statistics feature.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) » [canUseDescriptiveStatistics](#) value
- [“Statistics” on page 1668](#)

Capability Cookie Format Changed

The format of the capability cookie has been changed to reduce the size of the cookie value.

The capability cookie is used to store the set of global capabilities granted to the session during logon. The cookie does not store object capabilities.

The cookie value is now determined by assigning each capability a specific bit in a bit array. Bits are assigned right to left, starting with bit 0. The bit for a capability is assigned the value 1 if the user has the global capability, and 0 otherwise.

The bit array is converted to an array of 32-bit unsigned integer values, which are then hex encoded, with leading zeroes removed. The resulting words are concatenated and separated by semicolons (;). The resulting string is then signed to allow IBM Cognos 8 software to detect tampering.

The bit index for each capability is documented with the [bibus](#) » [userCapabilityEnum](#) enumeration set.

This change affects:

- [bibus](#) » [userCapabilityEnum](#) enumeration set

Lineage - Filter Support

Filter support has been added to the lineage specification. The following object types have been added.

- authoredReport Query Specification
 - `detailFilter` object type
 - `summaryFilter` object type
 - `slicerMemberSet` object type
- Framework Manager Model Schema Specification
 - `filter` object type
 - `slicerMemberSet` object type

The Source Mappings tables in [Chapter 25, “Using lineage specifications,” on page 1509](#) have been updated to reflect this change.

Documentation Updates

The following documentation has been added for this release to support lineage specifications:

- [Chapter 25, “Using lineage specifications,” on page 1509](#)
- [Chapter 26, “Lineage specification reference,” on page 1529](#)
- [Chapter 37, “Using selection context,” on page 1799](#)
- [Chapter 38, “Selection context reference,” on page 1803](#)

Lineage is the process of determining how objects in a [bibus » baseReport](#), [bibus » package](#), or [bibus » model](#) have been derived through transformations from other objects, back to source data. A lineage request is defined in the [bibus » metadataServiceLineageSpecification](#) class.

The `selectionContext` specification is available for application developers who wish to replace the default lineage information provided with a custom solution.

The following documentation has been added to support the use of the event specification:

- [Chapter 36, “Event specification reference,” on page 1787](#)
- An event specification instance is used to query events managed by either the [monitorService](#) service or the [eventManagementService](#) service.

Refer to the service-specific sections of the [asynch » runSpecification\(specification, parameterValues, options\)](#) method for more information.

Lineage Specification Changes

For performance reasons, the `datatype`, `precision`, `scale`, `regularAggregate`, and `semiAggregate` property types have been removed from the `dataItem` object type in the `authoredReport Query Specification`.

The code snippets and Source Mappings tables in [Chapter 25, “Using lineage specifications,” on page 1509](#) have been updated to reflect this change.

Bug Fixes

Several miscellaneous bug fixes were applied in this release.

This change affects:

- [“Adaptive Analytics” on page 1616](#)

Appendix J. New in Version 8.4

This appendix provides information about the new features in Version 8.4 of the product.

Hiding Objects in the Portal

Application developers should use the [`@shown= 'true'`] search predicate to select objects to show in the portal. The `bibus » uiClass » hidden` property is now used in an algorithm that calculates the value of the `bibus » uiClass » shown` property.

Other inputs to the algorithm are:

- the `showHiddenObjects` preference
- the `canUseShowHiddenObjectsPreference` capability
- the `owner` property of the object

The `canUseShowHiddenObjectsPreference` capability was added to allow administrators to limit user access to the user interface required to change the `showHiddenObjects` preference. It also suppresses the ability to set the `hidden` property on an object in its property page.

Changes in future releases related to determining whether an object should be shown in the portal will be reflected in the algorithm that calculates the value of the `bibus » uiClass » shown` property.

This change affects:

- `bibus » userCapabilityEnum » canUseShowHiddenObjectsPreference` value
- “Hide Entries” on page 1648
- `bibus » uiClass » hidden` property
- `bibus » uiClass » shown` property

Report Server `deliver()` method

The `report » deliver(conversation, parameterValues, options)` method was added to the `batchReportService` and `reportService` services to handle the delivery of report output to any destination.

This method replaces the existing methods `report » email(conversation, parameterValues, options)`, `report » print(conversation, parameterValues, options)`, `report » save(conversation, parameterValues, options)`, and `report » saveAs(conversation, parameterValues, options)`.

The destinations of the report output are specified using the options `archive`, `email`, `mobile`, `print`, `saveAs`, and `saveOutput`. Multiple destinations may be specified in the same request.

This change affects:

- `report » deliver(conversation, parameterValues, options)` method
- `bibus » deliveryChannelEnum » mobile` value
- `report` method set

Migration Service

The `migrationService` service was added to the IBM Cognos 8 architecture to perform migration tasks. A specification class and a configuration class have been added for this service. In addition, configuration properties have been added to the classes `bibus » configuration`, `bibus » configurationFolder` and `bibus » dispatcher`.

The [bibus » migrationTask](#) class defines tasks for the [migrationService](#). Instances of this class can be scheduled, and can also be used by agents (see the [bibus » agentTaskDefinition » taskObject](#) property) and jobs (see the [bibus » jobStepDefinition » stepObject](#) property).

This change affects:

- [bibus » asynchDetailMessages](#) class
- [bibus » historyDetailMigrationService](#) class
- [bibus » migrationCubeMapping](#) class
- [bibus » migrationMapping](#) class
- [bibus » migrationService](#) class
- [bibus » migrationServiceSpecification](#) class
- [bibus » migrationTask](#) class
- [bibus » migrationTaskOption](#) class
- [bibus » migrationTaskOptionMappingArray](#) class
- [bibus » migrationTaskOptionResolution](#) class
- [bibus » migrationTaskOptionSearchPathSingleObject](#) class
- [bibus » migrationTaskOptionSearchPathSingleObjectArray](#) class
- [bibus » adminFolder » items](#) property
- [bibus » dispatcher » migrationService](#) property
- [bibus » eventRecord » runnable](#) property
- [bibus » jobStepDefinition » stepObject](#) property
- [bibus » agentTaskDefinition » taskObject](#) property
- [bibus » migrationTaskOptionEnum](#) enumeration set
- [bibus » installedComponentEnum » migrationService](#) value
- [bibus » configuration » misAuditLevel](#) property
- [bibus » configurationFolder » misAuditLevel](#) property
- [bibus » dispatcher » misAuditLevel](#) property
- [bibus » migrationService » misAuditLevel](#) property
- [bibus » configuration » misConnections](#) property
- [bibus » configurationFolder » misConnections](#) property
- [bibus » dispatcher » misConnections](#) property
- [bibus » migrationService » misConnections](#) property
- [bibus » configuration » misPeakConnections](#) property
- [bibus » configurationFolder » misPeakConnections](#) property
- [bibus » dispatcher » misPeakConnections](#) property
- [bibus » migrationService » misPeakConnections](#) property
- [migrationService](#) service

On Demand Refresh of Prompt Cache

Instances of the [bibus » reportCache](#) class stored as a child of a [bibus » baseReport](#) object are now subject to Content Manager retention rules.

The content manager applies retention rules only when objects are added, so the [readonly](#) property [expirationTime](#) was added to the [bibus » reportCache](#) class (and numerous other classes) to allow applications to filter the set of objects subject to retention rules. This applies, for example, to [content »](#)

`query(searchPath, properties, sortBy, options)` calls when selecting by the expiration date and time of the object.

If you use this feature, you should set the retention rules for the prompt cache, typically based on the age of the object. The following query compares a `expirationTime` value to an arbitrary time to retrieve report objects which have not expired with respect to the given time.

```
<report search path>/reportCache[@expirationTime < '<time-value>']
```

The `bibus » runOptionEnum » promptCacheMode` value was added to allow administrators to control how the report service acts on the prompt cache when a report is run. Values for `promptCacheMode` are as follows:

- `none` - disables prompt cache processing. This is the default value for this option.
- `create` - deletes the existing prompt cache and then creates prompt cache entries for the specified locales. This replaces the deprecated `savePromptCache` run option with a value of `true`.
- `refresh` - creates a prompt cache if necessary and then adds prompt cache entries for the locales specified by the `outputLocale` run option if a matching prompt cache entry does not exist. Prompt cache entries for other locales are not affected.
- `update` - creates a prompt cache if necessary and then adds a prompt cache with entries for the locales specified by the `outputLocale` run option. Matching prompt cache entries are replaced. Prompt cache entries for other locales are not affected.

This change affects:

- `bibus » runOptionPromptCacheMode` class
- `bibus » baseReport » cache` property
- `bibus » promptCacheModeEnum` enumeration set
- `bibus » runOptionEnum » promptCacheMode` value
- `bibus » runOptionEnum » savePromptCache` value
- `bibus » agentOutputHotList » expirationTime` property
- `bibus » documentVersion » expirationTime` property
- `bibus » history » expirationTime` property
- `bibus » model » expirationTime` property
- `bibus » reportCache » expirationTime` property
- `bibus » reportVersion » expirationTime` property
- `bibus » shortcut » expirationTime` property

Object Capabilities

The ability to assign certain capabilities has been refined to support finer-grained configurations. In previous releases, the assignment of capabilities to users was controlled by granting or denying them globally. For example, a user entitled to use Reporting could use it to author a report based on any package for which the user had read access. In this release, the structure used to specify capability policies is similar to that of a security policy. Rather than specifying an access permission such as `execute` or `write`, the policy specifies a user capability such as `canUseQueryStudio` or `canUseUserDefinedSQL` on chosen objects.

The `bibus » package` class has been extended to support policies that determine the set of capabilities granted to users when they are using a particular `bibus » package`. To use a capability that can be specified on an object, the user must be granted the capability both globally and on the object.

The classes `bibus » content` and `bibus » folder` also support specifying capability policies to simplify capability administration. Capability policies for an object are acquired from an ancestor object unless specified in the object.

For example, administrators can allow a user to create reports, using Reporting, based on the Sales package. To do this, a member of the “[Directory Administrators](#)” on [page 1697](#) role needs to know that a user needs to create reports and then grant the user the global [canUseReportStudio](#) capability. A member of the “[Report Administrators](#)” on [page 1707](#) role then needs to know that the user needs to create sales reports and grant the specific [canUseReportStudio](#) capability on the Sales [bibus » package](#).

The “[Public Folders](#)” on [page 1759](#) predefined object has been updated to define a default capability policy for contained packages. The “[Everyone](#)” on [page 1698](#) group is granted all capabilities that can be specified on an object.

By default, the [canUseObjectCapabilities](#) capability is not granted to any role. A member of the “[Directory Administrators](#)” on [page 1697](#) role must grant the capability to some role before object capabilities can be assigned.

This change affects:

- [bibus » userCapabilityCache](#) class
- [bibus » userCapabilityPermission](#) class
- [bibus » userCapabilityPolicy](#) class
- [bibus » content » effectiveUserCapabilities](#) property
- [bibus » folder » effectiveUserCapabilities](#) property
- [bibus » package » effectiveUserCapabilities](#) property
- [bibus » content » userCapabilities](#) property
- [bibus » folder » userCapabilities](#) property
- [bibus » package » userCapabilities](#) property
- [bibus » content » userCapabilityPolicies](#) property
- [bibus » folder » userCapabilityPolicies](#) property
- [bibus » package » userCapabilityPolicies](#) property
- [bibus » userCapabilityEnum](#) enumeration set
- [bibus » userCapabilityEnum » canUseObjectCapabilities](#) value
- “[Public Folders](#)” on [page 1759](#)
- “[Set Entry-Specific Capabilities](#)” on [page 1667](#)
- [bibus » biBusHeader » userCapabilityCache](#) property

PowerPlay 8 Integration

IBM Cognos PowerPlay and the IBM Cognos 8 architecture have been integrated.

The [powerPlayService](#) service was added to the IBM Cognos 8 architecture to support PowerPlay reports. A set of specification classes and a configuration class have been added to support this new service. Configuration properties have also been added to the classes [bibus » configuration](#), [bibus » configurationFolder](#) and [bibus » dispatcher](#).

The [bibus » powerPlay8Report](#) class defines the information used to generate a PowerPlay 8 report.

A [bibus » powerPlay8ReportView](#) refers to an [bibus » authoredPowerPlay8Report](#), allowing that [bibus » authoredPowerPlay8Report](#) to be executed with different options and parameter values, or under the control of a different schedule.

Instances of the [bibus » basePowerPlay8Report](#) class may be used as part of an agent (see the [bibus » agentTaskDefinition » taskObject](#) property) or as part of a job (see the [bibus » jobStepDefinition » stepObject](#) property).

New capabilities and roles were added to define the set of users that use PowerPlay. The initial security policies of several objects were updated to reflect the addition of these new roles.

The `dataSource` method set was added to support using the `metadataService` service to test data source connections when the `batchReportService` and `reportService` services are not available. The capability rules controlling the use of the `dataSource » testDataSourceConnection(connectionString, credentials)` method were updated to allow users with the `canUsePowerPlay` capability to use these methods.

This change affects:

- `dataSource » testDataSourceConnection(connectionString, credentials)` method
- `report » testDataSourceConnection(connectionString, credentials)` – obsolete method
- `dataSource » testDataSourceConnection(connectionString, credentials)` method
- `dataSource » testDataSourceConnection(connectionString, credentials)` method
- `bibus » authoredPowerPlay8Report` class
- `bibus » basePowerPlay8Report` class
- `bibus » powerPlay8Option` class
- `bibus » powerPlay8OptionBoolean` class
- `bibus » powerPlay8OptionData` class
- `bibus » powerPlay8OptionLanguageArray` class
- `bibus » powerPlay8OptionOutputFormat` class
- `bibus » powerPlay8OptionSaveAs` class
- `bibus » powerPlay8OptionSearchPathSingleObject` class
- `bibus » powerPlay8OptionString` class
- `bibus » powerPlay8Report` class
- `bibus » powerPlay8ReportView` class
- `bibus » powerPlayService` class
- `bibus » powerPlayServiceReportSpecification` class
- `bibus » powerPlayServiceSpecification` class
- `bibus » folder » items` property
- `bibus » package » items` property
- `bibus » shortcutRSTask » link` property
- `bibus » dispatcher » powerPlayService` property
- `bibus » eventRecord » runnable` property
- `bibus » jobStepDefinition » stepObject` property
- `bibus » shortcut » target` property
- `bibus » agentTaskDefinition » taskObject` property
- `bibus » basePowerPlay8ReportActionEnum` enumeration set
- `bibus » powerPlay8DataEnum` enumeration set
- `bibus » powerPlay8OptionEnum` enumeration set
- `bibus » powerPlay8OutputFormatEnum` enumeration set
- `bibus » powerPlay8SaveAsEnum` enumeration set
- `bibus » userCapabilityEnum » canUsePowerPlay` value
- `bibus » userCapabilityEnum » canUsePowerPlayAdministration` value
- `bibus » installedComponentEnum » powerPlayService` value
- `bibus » installedComponentEnum » powerPlayStudio` value
- `bibus » uiComponentEnum » powerPlayStudio` value
- `bibus » systemMetricEnum » requestsPerSecond` value

- [bibus](#) » [systemMetricEnum](#) » [requestsPerSecondHighWaterMark](#) value
- [bibus](#) » [systemMetricEnum](#) » [requestsPerSecondLowWaterMark](#) value
- [bibus](#) » [systemMetricEnum](#) » [responseTime](#) value
- [dataSource](#) method set
- [“Administration”](#) on page 1615
- [“Public Folders”](#) on page 1759
- [“PowerPlay”](#) on page 1722
- [“PowerPlay”](#) on page 1736
- [“PowerPlay Administrators”](#) on page 1705
- [“PowerPlay Users”](#) on page 1705
- [“Administration tasks”](#) on page 1618
- [“PowerPlay Servers”](#) on page 1625
- [“Run With Options”](#) on page 1633
- [“Scheduling Priority”](#) on page 1665
- [“Administration”](#) on page 1617
- [“Cognos Viewer”](#) on page 1631
- [“PowerPlay Studio”](#) on page 1655
- [“Scheduling”](#) on page 1661
- [“Watch Rules”](#) on page 1669
- [bibus](#) » [content](#) » [powerPlay8Configuration](#) property
- [bibus](#) » [folder](#) » [powerPlay8Configuration](#) property
- [bibus](#) » [package](#) » [powerPlay8Configuration](#) property
- [bibus](#) » [configuration](#) » [ppsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [ppsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [ppsAffineConnections](#) property
- [bibus](#) » [powerPlayService](#) » [ppsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [ppsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [ppsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [ppsAuditLevel](#) property
- [bibus](#) » [powerPlayService](#) » [ppsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [ppsExecutionTimeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [ppsExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [ppsExecutionTimeLimit](#) property
- [bibus](#) » [powerPlayService](#) » [ppsExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [ppsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [ppsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [ppsNonAffineConnections](#) property
- [bibus](#) » [powerPlayService](#) » [ppsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [ppsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [ppsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [ppsPeakAffineConnections](#) property
- [bibus](#) » [powerPlayService](#) » [ppsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [ppsPeakNonAffineConnections](#) property

- [bibus](#) » [configurationFolder](#) » [ppsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [ppsPeakNonAffineConnections](#) property
- [bibus](#) » [powerPlayService](#) » [ppsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [ppsQueueLimit](#) property
- [bibus](#) » [configurationFolder](#) » [ppsQueueLimit](#) property
- [bibus](#) » [dispatcher](#) » [ppsQueueLimit](#) property
- [bibus](#) » [powerPlayService](#) » [ppsQueueLimit](#) property
- [batchReportService](#) service
- [metadataService](#) service
- [powerPlayService](#) service
- [reportService](#) service

pagedLayoutDataXML Report Output Format

Reserved.

This change affects:

- [bibus](#) » [outputFormatEnum](#) » [pagedLayoutDataXML](#) value

Drill-Through Improvements

Additional drill-through functionality has been incorporated to convert a selection context from one format to another. For example, a selection context in PowerPlay 8 filter format can be converted to the IBM Cognos 8 format. A selection context with a list of parameter values can also be converted to the IBM Cognos 8 format.

This change affects:

- [drillThrough](#) » [convertDrillThroughContext\(inputContext, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailSelectionContext](#) class
- [bibus](#) » [drillThroughOptionAnyURI](#) class
- [bibus](#) » [selectionContextFormatEnum](#) enumeration set
- [bibus](#) » [drillThroughOptionEnum](#) » [contextFormat](#) value
- [drillThrough](#) method set

Dynamic Filtering of Report Data

It is now possible to specify additional filter criteria to be applied during the execution of [bibus](#) » [report](#) objects by the batch report service and report service.

The additional filters are defined in an XML document and are specified in a request using the [editSpecification](#) option.

Drill-through requests can take advantage of this new functionality when the drill-through target is a [bibus](#) » [report](#) object. The methods [asynch](#) » [run\(objectPath, parameterValues, options\)](#) and [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) have been enhanced to support drill-through-related objects and specifications, respectively. These methods are now able to generate appropriate filters for the selection context of the drill-through source. The [determineDrillThroughTargetParameterValues\(objectPaths, sourceContext, sourceContextValues, targetParameterAssignments, targetParameters, parameterValues, options \)](#) and

`getDrillThroughDefaultParameterAssignments(objectPath, parameterValues, options)` methods are now deprecated.

The `drillThrough » findDrillThroughPaths(objectPath, parameterValues, options)` method has been added to locate the drill-through paths that match the specified selection context. This method provides more information than the `getDrillThroughPaths(objectPath, parameterValues, options)` method, which is now deprecated.

This change affects:

- [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailDrillThroughRequest](#) class
- [bibus](#) » [asynchDetailDrillThroughTarget](#) class
- [bibus](#) » [asynchDetailDrillThroughTargetURI](#) class
- [bibus](#) » [drillPath](#) class
- [bibus](#) » [drillThroughOptionXMLEncodedXML](#) class
- [bibus](#) » [reportServiceDrillThroughSpecification](#) class
- [bibus](#) » [specificationOption](#) class
- [bibus](#) » [specificationOptionXMLEncodedXML](#) class
- [bibus](#) » [drillThroughActionEnum](#) enumeration set
- [bibus](#) » [drillThroughRecipientEnum](#) enumeration set
- [bibus](#) » [specificationOptionEnum](#) enumeration set
- [bibus](#) » [selectionContextFormatEnum](#) » [cognos8](#) value
- [bibus](#) » [drillThroughOptionEnum](#) » [diagnostics](#) value
- [bibus](#) » [drillThroughOptionEnum](#) » [selectionContext](#) value
- [bibus](#) » [drillThroughOptionEnum](#) » [useEditSpecification](#) value
- [drillThrough](#) method set

Writing Task History Subsets

The `writeCompleteHistory` option was added to allow task authors to control whether a complete task history is written to the content store when a task runs successfully.

Specifying this option with a value of `false` improves throughput when running background tasks by reducing the number of history objects written to the content store when the task and its child tasks complete successfully. Features of the Past Activities portal pages that rely on the presence of task history logs will not be available for those tasks affected by this option.

If the `writeCompleteHistory` option is not used, if it is used and its value is `true`, or if the task or any of its child tasks do not complete successfully, the complete task history is written to the content store.

This change affects:

- [bibus](#) » [monitorOptionEnum](#) » [writeCompleteHistory](#) value

Supporting New Drill-through Targets

The [bibus](#) » [drillPath](#) class has been extended to support additional types of drill-through targets. In previous releases, only [bibus](#) » [analysis](#), [bibus](#) » [query](#), [bibus](#) » [report](#), and [bibus](#) » [reportView](#) objects were valid drill-through targets. PowerPlay Studio cubes and [bibus](#) » [powerPlay8Report](#) objects are now also valid drill-through targets.

New methods were added to the batch report service ([batchReportService](#)) and report service ([reportService](#)) to support the creation, querying and updating of [bibus](#) » [drillPath](#) objects in the content

store. These methods handle upgrading the [bibus » reportServiceDrillThroughSpecification](#) as well as updating references to content store objects that may no longer be valid if the containing [bibus » drillPath](#) object has been deployed.

This change affects:

- [drillThrough » addDrillPath\(parentPath, object, options\)](#) method
- [drillThrough » queryDrillPath\(objectPath, parameterValues, options\)](#) method
- [drillThrough » updateDrillPath\(object, options\)](#) method
- [bibus » asynchDetailDrillPathObject](#) class
- [bibus » reportServiceDrillThroughSpecification](#) class
- [bibus » reportServiceQueryDrillPathOption](#) class
- [bibus » reportServiceQueryDrillPathOptionBoolean](#) class
- [bibus » drillPath » deploymentReferences](#) property
- [bibus » drillPath » target](#) property
- [bibus » drillPath » targetOptions](#) property
- [bibus » drillPath » targetParameters](#) property
- [bibus » reportServiceQueryDrillPathOptionEnum](#) enumeration set
- [drillThrough](#) method set
- [bibus » drillPath » specification](#) property

Mobile Service

New options and classes were added to specify the recipients of content delivered using the [mobileService](#).

This change affects:

- [bibus » mobileOption](#) class
- [bibus » mobileOptionSearchPathMultipleObjectArray](#) class
- [bibus » mobileOptionEnum](#) enumeration set

Migration Capabilities

The capabilities [canOpenPowerPlayInAnalysisStudio](#) and [canOpenPowerPlayInReportStudio](#) were added to allow administrators to restrict the ability of users to open [bibus » powerPlayReport](#) objects in Analysis Studio and Reporting, respectively.

By default, these capabilities are assigned to users of Analysis Studio and Reporting.

This change affects:

- [bibus » userCapabilityEnum » canOpenPowerPlayInAnalysisStudio](#) value
- [bibus » userCapabilityEnum » canOpenPowerPlayInReportStudio](#) value
- [“Open PowerPlay Reports with Analysis Studio” on page 1630](#)
- [“Open PowerPlay Reports with Reporting” on page 1660](#)

Lineage Metadata

Support for lineage has been added to the [metadataService](#).

SDK users can use the lineage specification to query the lineage of objects in the content store. For more information, see [Chapter 25, “Using lineage specifications,” on page 1509](#).

This change affects:

- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailMIMEAttachment](#) class
- [bibus](#) » [metadataServiceLineageSpecification](#) class
- [bibus](#) » [metadataServiceSpecification](#) class
- [bibus](#) » [configurationData](#) » [metadataInformationURI](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUseLineage](#) value
- [bibus](#) » [configurationDataEnum](#) » [metadataInformationURI](#) value
- [“Lineage”](#) on page 1650
- [bibus](#) » [configuration](#) » [mdsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mdsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [mdsAffineConnections](#) property
- [bibus](#) » [metadataService](#) » [mdsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [mdsExecutionTimeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [mdsExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [mdsExecutionTimeLimit](#) property
- [bibus](#) » [metadataService](#) » [mdsExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [mdsMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [mdsMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [mdsMaximumProcesses](#) property
- [bibus](#) » [metadataService](#) » [mdsMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [mdsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mdsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [mdsNonAffineConnections](#) property
- [bibus](#) » [metadataService](#) » [mdsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [mdsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mdsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [mdsPeakAffineConnections](#) property
- [bibus](#) » [metadataService](#) » [mdsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [mdsPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [mdsPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [mdsPeakMaximumProcesses](#) property
- [bibus](#) » [metadataService](#) » [mdsPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [mdsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mdsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [mdsPeakNonAffineConnections](#) property
- [bibus](#) » [metadataService](#) » [mdsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [mdsQueueLimit](#) property
- [bibus](#) » [configurationFolder](#) » [mdsQueueLimit](#) property
- [bibus](#) » [dispatcher](#) » [mdsQueueLimit](#) property
- [bibus](#) » [metadataService](#) » [mdsQueueLimit](#) property

- [bibus](#) » [configuration](#) » [metadataInformationURI](#) property
- [metadataService](#) service

URL Validation Rules

New IBM Cognos Application Firewall validation rules were introduced to provide protection against potentially harmful code in URLs. A new advanced configuration setting, `RSVP.RENDER.VALIDATEURL`, was added to specify whether these rules are applied to values specified by the `cssURL` option or any URL values contained within a report specification. The default value for the `RSVP.RENDER.VALIDATEURL` advanced setting is `false`, which means that URL values for this option are not validated.

This change affects:

- [bibus](#) » [runOptionEnum](#) » [cssURL](#) value

Index Options

New index options were added to allow a subset of the objects in the content store to be included in or excluded from the index when the index update task is run. These options allow administrators to specify what content to index and allows the index system to efficiently perform incremental updates on an index.

This change affects:

- [bibus](#) » [indexOptionSearchPathMultipleObjectArray](#) class
- [bibus](#) » [indexOptionEnum](#) » [excludedObjects](#) value
- [bibus](#) » [indexOptionEnum](#) » [includedObjects](#) value

Adaptive Analytics Integration

New capabilities and roles were added to define the set of users who can use Adaptive Analytics. The initial security policies of several objects were updated to reflect the addition of these new roles.

This change affects:

- [bibus](#) » [installedComponentEnum](#) » [adaptiveAnalytics](#) value
- [bibus](#) » [uiComponentEnum](#) » [adaptiveAnalytics](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseAdaptiveAnalytics](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseAdaptiveAnalyticsAdministration](#) value
- [“Adaptive Analytics Administrators” on page 1688](#)
- [“Adaptive Analytics Users” on page 1688](#)
- [“Adaptive Analytics Administration” on page 1618](#)
- [“Adaptive Analytics” on page 1616](#)
- [“Administration” on page 1617](#)

Software Editions

Several properties were added to record the software edition of a service, as well as the edition of the software that is used to handle requests using a particular object.

Configuration information has been added to specify the web location of a gateway for a particular software edition.

This change affects:

- [bibus](#) » [biBusHeaderExtension1](#) class
- [bibus](#) » [softwareEdition](#) class
- [bibus](#) » [configuration](#) » [editions](#) property
- [bibus](#) » [portletProducer](#) » [routingHints](#) property
- [bibus](#) » [dispatcher](#) » [edition](#) property
- [bibus](#) » [biBusHeader](#) » [extension](#) property
- [bibus](#) » [portletProducer](#) » [routingServerGroup](#) property
- [bibus](#) » [baseAgentDefinition](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [baseDataIntegrationTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [baseDataMovementTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [baseReport](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [baseRSSTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [contentTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [exportDeployment](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [importDeployment](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [indexUpdateTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [jobDefinition](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [memo](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [package](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [planningMacroTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [planningTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [portalSkin](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [portletProducer](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [storedProcedureTask](#) » [routingServerGroupEdition](#) property
- [bibus](#) » [webServiceTask](#) » [routingServerGroupEdition](#) property

Dashboards

With the inclusion of executive dashboards in the IBM Cognos 8 UI, the following additions were made to incorporate dashboard support:

- the [bibus](#) » [dashboard](#) class to instantiate the new content store objects
- the global capabilities, [canUseDashboardViewerFileManagement](#) for dashboard designers and [canUseDashboardViewer](#) for dashboard users
- the [bibus](#) » [personalization](#) class to enable persistence of personalization information
- the [bibus](#) » [personalizationFolder](#) class to instantiate containers for user-saved [bibus](#) » [personalization](#) objects

Instances of the [bibus](#) » [personalization](#) class are used to persist the state of dashboards so that designers can pre-arrange a dashboard layout for users and users can return to a dashboard layout they have previously saved.

This change affects:

- [bibus](#) » [dashboard](#) class
- [bibus](#) » [personalization](#) class
- [bibus](#) » [personalizationFolder](#) class

- [bibus](#) » [folder](#) » [items](#) property
- [bibus](#) » [package](#) » [items](#) property
- [bibus](#) » [pageletFolder](#) » [items](#) property
- [bibus](#) » [account](#) » [personalizationFolder](#) property
- [bibus](#) » [shortcut](#) » [target](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewer](#) value
- [bibus](#) » [userCapabilityEnum](#) » [canUseDashboardViewerFileManagement](#) value
- [bibus](#) » [installedComponentEnum](#) » [dashboardViewer](#) value
- “Use the Edit Features” on [page 1643](#)
- “Executive Dashboard” on [page 1640](#)

Updating the Credential of Imported Schedules

The Content Manager advanced setting `CM.DeploymentUpdateScheduleCredential` was added so that when a deployment archive is imported into the content store, the `credential` property of all imported [bibus](#) » [schedule](#) objects can be changed to reference the [bibus](#) » [credential](#) contained in the [bibus](#) » [account](#) used to import the deployment.

To change the `credential` property, the `takeOwnership` option must be used during the [import](#) and the value of `CM.DeploymentUpdateScheduleCredential` must be `true`.

This change affects:

- [bibus](#) » [deploymentOptionEnum](#) » [takeOwnership](#) value

Package Data Sources

Reserved.

This change affects:

- [bibus](#) » [aliasLocation](#) class
- [bibus](#) » [metadataServiceModelInformationSpecification](#) class
- [bibus](#) » [configuration](#) » [aliasLocations](#) property
- [bibus](#) » [package](#) » [dataSources](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUsePackageDataSources](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [packageDataSourceConflictResolution](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [packageDataSourceSelect](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [packageDataSourceSignonSelect](#) value
- “Package Data Sources” on [page 1653](#)

Report Output Annotations

Support for annotations has been added. Annotations are additional information, such as questions or explanations, that are attached to report outputs or created and saved with agents. They can also be viewed by report consumers other than the one who made the annotation.

Annotations are stored in [bibus](#) » [annotation](#) objects and can be added to the content store using the [content](#) » [addAnnotations\(containerPath, objects, options\)](#) method.

This change affects:

- [content](#) » [addAnnotations\(containerPath, objects, options\)](#) method
- [bibus](#) » [annotation](#) class
- [bibus](#) » [annotationFolder](#) class
- [bibus](#) » [agentOutputHotList](#) » [annotationFolder](#) property
- [bibus](#) » [reportVersion](#) » [annotationFolder](#) property
- [bibus](#) » [agentOptionEnum](#) » [allowAnnotations](#) value
- [bibus](#) » [runOptionEnum](#) » [allowAnnotations](#) value
- [content](#) method set
- [bibus](#) » [agentOutputHotList](#) » [allowAnnotations](#) property
- [bibus](#) » [reportVersion](#) » [allowAnnotations](#) property

Dimension Management Service

Support for master dimension management was added to IBM Cognos 8. Dimensions can be added, modified, deleted, and retrieved by using the [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method to send a request in a [bibus](#) » [dimensionManagementServiceSpecification](#) object.

This change affects:

- [asynch](#) » [runSpecification\(specification, parameterValues, options\)](#) method
- [bibus](#) » [dimensionManagementService](#) class
- [bibus](#) » [dimensionManagementServiceSpecification](#) class
- [bibus](#) » [dispatcher](#) » [dimensionManagementService](#) property
- [bibus](#) » [installedComponentEnum](#) » [dimensionManagementService](#) value
- [bibus](#) » [configuration](#) » [dimsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dimsAffineConnections](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dimsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dimsAuditLevel](#) property
- [bibus](#) » [configurationFolder](#) » [dimsAuditLevel](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsAuditLevel](#) property
- [bibus](#) » [dispatcher](#) » [dimsAuditLevel](#) property
- [bibus](#) » [configuration](#) » [dimsExecutionTimeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [dimsExecutionTimeLimit](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [dimsExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [dimsMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [dimsMaximumProcesses](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [dimsMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [dimsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dimsNonAffineConnections](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dimsNonAffineConnections](#) property

- [bibus](#) » [configuration](#) » [dimsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dimsPeakAffineConnections](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dimsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dimsPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [dimsPeakMaximumProcesses](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [dimsPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [dimsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dimsPeakNonAffineConnections](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dimsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dimsQueueLimit](#) property
- [bibus](#) » [configurationFolder](#) » [dimsQueueLimit](#) property
- [bibus](#) » [dimensionManagementService](#) » [dimsQueueLimit](#) property
- [bibus](#) » [dispatcher](#) » [dimsQueueLimit](#) property
- [dimensionManagementService](#) service

Support for IBM® WebSphere® Business Glossary

Support for glossaries has been added. Glossaries allow users to map metadata to terminology that is meaningful to business users of IBM Cognos 8. Users viewing IBM Cognos 8 output are able to use the glossary to view descriptions of selected output items.

Users with the [canUseGlossary](#) capability can use a glossary if one is installed. The [glossaryURI](#) parameter specifies the URI of the glossary provider agent.

This change affects:

- [bibus](#) » [configurationData](#) » [glossaryURI](#) property
- [bibus](#) » [userCapabilityEnum](#) » [canUseGlossary](#) value
- [bibus](#) » [configurationDataEnum](#) » [glossaryURI](#) value
- “Glossary” on page 1648
- [bibus](#) » [configuration](#) » [glossaryURI](#) property

userInterface property of the uiProfile class

The [bibus](#) » [uiProfile](#) » [userInterface](#) property is now intrinsic. This means that it can be retrieved without having read permission on the object. Only traverse permission for all ancestors of the object is required.

This change affects:

- [bibus](#) » [uiProfile](#) » [userInterface](#) property

Bug Fixes

Several miscellaneous bug fixes were applied in this release.

This change affects:

- “Capability” on page 1616

- [“Most Recently Used list” on page 1691](#)
- [“Most Recently Used list” on page 1681](#)
- [“Content Administration” on page 1713](#)
- [“Business” on page 1757](#)
- [“Content Administration” on page 1730](#)
- [“Portlets” on page 1735](#)
- [“System” on page 1738](#)
- [“/” on page 1614](#)
- [“Administration tasks” on page 1618](#)
- [“Configure and manage the system” on page 1619](#)
- [“Data Source Connections” on page 1620](#)
- [“Metric Studio Administration” on page 1622](#)
- [“Planning Administration” on page 1624](#)
- [“Run activities and schedules” on page 1626](#)
- [“Run With Options” on page 1633](#)
- [“Set capabilities and manage UI profiles” on page 1627](#)
- [“Styles and portlets” on page 1628](#)
- [“Users, Groups, and Roles” on page 1629](#)
- [“My Watch Items” on page 1692](#)
- [“My Watch Items” on page 1682](#)

Appendix K. New in Version 8.3

This appendix provides information about the new features in Version 8.3 of the product.

Conditional Subscriptions

It is now possible to subscribe to output generated by a [bibus » baseReport](#).

When a user subscribes to a report, a subscription agent ([bibus » reportDataServiceAgentDefinition](#) class) is created in the user's [bibus » subscriptionFolder](#), which is a child of the user's [bibus » account](#) object. Like other agents, subscription agents support the conditional generation of events to reduce the number of times the agent tasks are run. For example, a user might request that the "Recent Orders" report be forwarded to the user's email Inbox only if an order with a total value greater than \$10MM USD is reported.

The capability [bibus » userCapabilityEnum » canUseConditionalSubscriptions](#) value can be used to limit the number of users that can subscribe to reports.

The output of a report must be saved in the content store if users can subscribe to the report. To subscribe to a report, a user must have `traverse` and `read` permissions to the report (to allow access to the report output).

If a report allows subscriptions, saved output is also saved as document content to the transient state folder in the content store. The subscription agents process the output in the transient folder. The policy of the report that produces the output is copied to the new document version.

This change affects:

- [bibus » agentDefinition](#) class
- [bibus » authoredAgentDefinition](#) class
- [bibus » periodical](#) class
- [bibus » reportDataServiceAgentDefinition](#) class
- [bibus » reportDataServiceSpecification](#) class
- [bibus » subscriptionFolder](#) class
- [bibus » subscriptionOption](#) class
- [bibus » subscriptionOptionSearchPathSingleObject](#) class
- [bibus » agentDefinition » eventKey](#) property
- [bibus » authoredAgentDefinition » eventKey](#) property
- [bibus » agentDefinition » items](#) property
- [bibus » authoredAgentDefinition » items](#) property
- [bibus » transientStateFolder » items](#) property
- [bibus » documentContent » pages](#) property
- [bibus » account » subscriptionFolder](#) property
- [bibus » agentDefinition » tasks](#) property
- [bibus » authoredAgentDefinition » tasks](#) property
- [bibus » subscriptionOptionEnum](#) enumeration set
- [bibus » userCapabilityEnum » canUseConditionalSubscriptions](#) value
- [bibus » scheduleTypeEnum » subscription](#) value
- [event](#) method set
- [“Watch Rules” on page 1669](#)

- [bibus](#) » [baseReport](#) » [allowSubscription](#) property
- [bibus](#) » [documentContent](#) » [burstID](#) property
- [bibus](#) » [output](#) » [burstID](#) property
- [bibus](#) » [documentContent](#) » [burstKey](#) property
- [bibus](#) » [output](#) » [burstKey](#) property
- [bibus](#) » [documentContent](#) » [context](#) property
- [bibus](#) » [documentContent](#) » [contextBlockCount](#) property
- [bibus](#) » [documentContent](#) » [format](#) property
- [bibus](#) » [documentContent](#) » [lastPage](#) property
- [bibus](#) » [configuration](#) » [periodicalDocumentVersionRetentionAge](#) property
- [bibus](#) » [configuration](#) » [periodicalDocumentVersionRetentionCount](#) property
- [bibus](#) » [agentDefinition](#) » [sequencing](#) property
- [bibus](#) » [authoredAgentDefinition](#) » [sequencing](#) property

Report Email Alerts

Users may now request that they be alerted when a [bibus](#) » [baseReport](#) object runs and saves new output. If the [bibus](#) » [baseReport](#) » [allowNotification](#) property is set to `true`, an alert will be sent to each user on the object's alert list whenever new output is available.

- Use the [delivery](#) » [queryNotification\(objectPath\)](#) method to determine whether a [bibus](#) » [baseReport](#) object allows alerts, and if it does, whether the current user is on the object's alert list.
- Use the [delivery](#) » [addNotification\(objectPath\)](#) method to add the current user to a [bibus](#) » [baseReport](#) object's alert list.
- Use the [delivery](#) » [deleteNotification\(objectPath\)](#) method to delete the current user from a [bibus](#) » [baseReport](#) object's alert list.
- Use the [delivery](#) » [deleteAllNotifications\(\)](#) method to delete the current user from the alert lists of all agents and reports.
- Use the [delivery](#) » [clearNotifications\(objectPath\)](#) method to delete all users from a [bibus](#) » [baseReport](#) object's alert list. The caller must have write permission on the [bibus](#) » [baseReport](#) object to use this method.

The [delivery](#) service ([deliveryService](#)) implements these methods.

The properties [bibus](#) » [baseReport](#) » [notificationListIsEmpty](#) and [bibus](#) » [baseAgentDefinition](#) » [notificationListIsEmpty](#) can be used to determine whether the notification list of the object is empty.

When the [delivery](#) service processes the alert list, it uses the email address specified in the [bibus](#) » [account](#) » [notificationEMail](#) property to send alerts to the user. If there is no email address specified in this property and the user running the task has the necessary access permissions to view the subscriber's account, the email address specified in the [bibus](#) » [account](#) » [email](#) property will be used.

This change affects:

- [delivery](#) » [addNotification\(objectPath\)](#) method
- [delivery](#) » [clearNotifications\(objectPath\)](#) method
- [delivery](#) » [deleteAllNotifications\(\)](#) method
- [delivery](#) » [deleteNotification\(objectPath\)](#) method
- [delivery](#) » [queryNotification\(objectPath\)](#) method
- [bibus](#) » [asynchDetailAgentNotificationStatus](#) class
- [bibus](#) » [baseReport](#) » [notificationList](#) property
- [bibus](#) » [agentNotificationStatusEnum](#) enumeration set

- [bibus](#) » [deliveryOptionEnum](#) » [notificationList](#) value
- [bibus](#) » [baseReport](#) » [allowNotification](#) property
- [bibus](#) » [account](#) » [notificationEMail](#) property
- [bibus](#) » [baseAgentDefinition](#) » [notificationListIsEmpty](#) property
- [bibus](#) » [baseReport](#) » [notificationListIsEmpty](#) property

Agent Default Portal Action

It is now possible to specify a default action for agents in the portal. The [bibus](#) » [baseAgentDefinition](#) » [defaultPortalAction](#) property was added to specify the action performed. This allows users to change the action (for example, run or open for editing, depending on the item) invoked when they click a hyperlinked name in the portal. The user clicking the name must have sufficient permissions and capabilities to perform the action.

The [bibus](#) » [baseAgentDefinition](#) » [mostRecentEventList](#) property was added to return information about the most recent event list for the agent.

This change affects:

- [bibus](#) » [baseAgentDefinition](#) » [mostRecentEventList](#) property
- [bibus](#) » [baseAgentDefinitionActionEnum](#) enumeration set
- [bibus](#) » [baseAgentDefinition](#) » [defaultPortalAction](#) property

Agent Task Event Filters

It is now possible to filter the events passed to an agent task ([bibus](#) » [agentTaskDefinition](#)). This adds the ability to execute a task based on specific conditions. Specifying a data item name in the [bibus](#) » [agentTaskDefinition](#) » [filterDataItemName](#) property requires the agent to filter the events processed by the agent task. Only events that evaluate to a value of `true` for the named data item are processed by the agent task. The data item must be defined in the agent's condition report (the [bibus](#) » [report](#) object contained by the [bibus](#) » [agentDefinition](#) object).

If the [bibus](#) » [agentTaskDefinition](#) » [filterDataItemName](#) property is not specified or references a data item from the agent's condition report that does not produce boolean values, all events are passed to the agent task.

This change affects:

- [bibus](#) » [agentTaskDefinition](#) » [filterDataItemName](#) property

Saving Report Output to File System

It is now possible to archive report output to a file system.

Use the [bibus](#) » [runOptionEnum](#) » [archive](#) value when running the report.

This change affects:

- [bibus](#) » [archiveDescriptor](#) class
- [bibus](#) » [archiveLocation](#) class
- [bibus](#) » [archiveOption](#) class
- [bibus](#) » [archiveOptionConflictResolution](#) class
- [bibus](#) » [archiveOptionDescriptor](#) class
- [bibus](#) » [archiveOptionSearchPathSingleObject](#) class
- [bibus](#) » [archiveOptionString](#) class

- [bibus](#) » [configuration](#) » [archiveLocations](#) property
- [bibus](#) » [archiveConflictResolutionEnum](#) enumeration set
- [bibus](#) » [archiveOptionEnum](#) enumeration set
- [bibus](#) » [deliveryChannelEnum](#) » [archive](#) value
- [bibus](#) » [runOptionEnum](#) » [archive](#) value

Improve Batch Processing

Runtime configuration parameters have been added to several service configuration classes to provide data related to the capacity of the associated service instance to perform batch activities. These parameters are used by the [dispatcher](#) and [monitorService](#) to distribute batch tasks to services with available capacity.

This results in the following general changes from previous versions:

- New runtime configuration parameters now exist for each service. These manage the maximum number of tasks allowed during peak or non-peak times, replacing settings previously used by the [monitorService](#).
- Runtime configuration parameters related to how peak and non-peak times are configured have moved from the [monitorService](#) to the [dispatcher](#). These properties no longer have any effect if configured for the [bibus](#) » [monitorService](#).
- These parameters have also been added to the [bibus](#) » [configuration](#) object and the [bibus](#) » [configurationFolder](#) object to facilitate management by property value acquisition.

This change affects:

- [bibus](#) » [agentService](#) » [asConnections](#) property
- [bibus](#) » [configuration](#) » [asConnections](#) property
- [bibus](#) » [configurationFolder](#) » [asConnections](#) property
- [bibus](#) » [dispatcher](#) » [asConnections](#) property
- [bibus](#) » [agentService](#) » [asPeakConnections](#) property
- [bibus](#) » [configuration](#) » [asPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [asPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [asPeakConnections](#) property
- [bibus](#) » [batchReportService](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsPeakAffineConnections](#) property
- [bibus](#) » [batchReportService](#) » [brsPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [brsPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [brsPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [brsPeakMaximumProcesses](#) property
- [bibus](#) » [batchReportService](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [brsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [cmsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [cmsConnections](#) property
- [bibus](#) » [contentManagerService](#) » [cmsConnections](#) property

- [bibus](#) » [dispatcher](#) » [cmsConnections](#) property
- [bibus](#) » [configuration](#) » [cmsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [cmsPeakConnections](#) property
- [bibus](#) » [contentManagerService](#) » [cmsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [cmsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [disConnections](#) property
- [bibus](#) » [configurationFolder](#) » [disConnections](#) property
- [bibus](#) » [dataIntegrationService](#) » [disConnections](#) property
- [bibus](#) » [dispatcher](#) » [disConnections](#) property
- [bibus](#) » [configuration](#) » [disPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [disPeakConnections](#) property
- [bibus](#) » [dataIntegrationService](#) » [disPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [disPeakConnections](#) property
- [bibus](#) » [configuration](#) » [dmsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dsConnections](#) property
- [bibus](#) » [deliveryService](#) » [dsConnections](#) property
- [bibus](#) » [dispatcher](#) » [dsConnections](#) property
- [bibus](#) » [configuration](#) » [dsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dsPeakConnections](#) property
- [bibus](#) » [deliveryService](#) » [dsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [dsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [idsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [idsConnections](#) property
- [bibus](#) » [dispatcher](#) » [idsConnections](#) property
- [bibus](#) » [indexDataService](#) » [idsConnections](#) property
- [bibus](#) » [configuration](#) » [idsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [idsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [idsPeakConnections](#) property
- [bibus](#) » [indexDataService](#) » [idsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [issConnections](#) property
- [bibus](#) » [configurationFolder](#) » [issConnections](#) property
- [bibus](#) » [dispatcher](#) » [issConnections](#) property
- [bibus](#) » [indexSearchService](#) » [issConnections](#) property
- [bibus](#) » [configuration](#) » [issPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [issPeakConnections](#) property

- [bibus](#) » [dispatcher](#) » [issPeakConnections](#) property
- [bibus](#) » [indexSearchService](#) » [issPeakConnections](#) property
- [bibus](#) » [configuration](#) » [iusConnections](#) property
- [bibus](#) » [configurationFolder](#) » [iusConnections](#) property
- [bibus](#) » [dispatcher](#) » [iusConnections](#) property
- [bibus](#) » [indexUpdateService](#) » [iusConnections](#) property
- [bibus](#) » [configuration](#) » [iusPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [iusPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [iusPeakConnections](#) property
- [bibus](#) » [indexUpdateService](#) » [iusPeakConnections](#) property
- [bibus](#) » [configuration](#) » [jsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [jsConnections](#) property
- [bibus](#) » [dispatcher](#) » [jsConnections](#) property
- [bibus](#) » [jobService](#) » [jsConnections](#) property
- [bibus](#) » [configuration](#) » [jsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [jsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [jsPeakConnections](#) property
- [bibus](#) » [jobService](#) » [jsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [mbsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mbsConnections](#) property
- [bibus](#) » [dispatcher](#) » [mbsConnections](#) property
- [bibus](#) » [mobileService](#) » [mbsConnections](#) property
- [bibus](#) » [configuration](#) » [mbsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mbsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [mbsPeakConnections](#) property
- [bibus](#) » [mobileService](#) » [mbsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [mmsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mmsConnections](#) property
- [bibus](#) » [dispatcher](#) » [mmsConnections](#) property
- [bibus](#) » [metricsManagerService](#) » [mmsConnections](#) property
- [bibus](#) » [configuration](#) » [mmsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [mmsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [mmsPeakConnections](#) property
- [bibus](#) » [metricsManagerService](#) » [mmsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [msNonPeakDemandBeginHour](#) property
- [bibus](#) » [configurationFolder](#) » [msNonPeakDemandBeginHour](#) property
- [bibus](#) » [dispatcher](#) » [msNonPeakDemandBeginHour](#) property
- [bibus](#) » [configuration](#) » [msPeakDemandBeginHour](#) property
- [bibus](#) » [configurationFolder](#) » [msPeakDemandBeginHour](#) property
- [bibus](#) » [dispatcher](#) » [msPeakDemandBeginHour](#) property
- [bibus](#) » [monitorService](#) » [msPeakDemandBeginHour](#) property
- [bibus](#) » [configuration](#) » [msPeakDemandMaximumTasks](#) property
- [bibus](#) » [configurationFolder](#) » [msPeakDemandMaximumTasks](#) property

- [bibus](#) » [dispatcher](#) » [msPeakDemandMaximumTasks](#) property
- [bibus](#) » [monitorService](#) » [msPeakDemandMaximumTasks](#) property
- [bibus](#) » [configuration](#) » [nonPeakDemandBeginHour](#) property
- [bibus](#) » [configurationFolder](#) » [nonPeakDemandBeginHour](#) property
- [bibus](#) » [dispatcher](#) » [nonPeakDemandBeginHour](#) property
- [bibus](#) » [configuration](#) » [pacsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [pacsConnections](#) property
- [bibus](#) » [dispatcher](#) » [pacsConnections](#) property
- [bibus](#) » [planningAdministrationConsoleService](#) » [pacsConnections](#) property
- [bibus](#) » [configuration](#) » [pacsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [pacsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [pacsPeakConnections](#) property
- [bibus](#) » [planningAdministrationConsoleService](#) » [pacsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [pdsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [pdsConnections](#) property
- [bibus](#) » [dispatcher](#) » [pdsConnections](#) property
- [bibus](#) » [planningDataService](#) » [pdsConnections](#) property
- [bibus](#) » [configuration](#) » [pdsMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [pdsMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [pdsMaximumProcesses](#) property
- [bibus](#) » [planningDataService](#) » [pdsMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [pdsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [pdsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [pdsPeakConnections](#) property
- [bibus](#) » [planningDataService](#) » [pdsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [pdsPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [pdsPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [pdsPeakMaximumProcesses](#) property
- [bibus](#) » [planningDataService](#) » [pdsPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [peakDemandBeginHour](#) property
- [bibus](#) » [configurationFolder](#) » [peakDemandBeginHour](#) property
- [bibus](#) » [dispatcher](#) » [peakDemandBeginHour](#) property
- [bibus](#) » [configuration](#) » [prsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [prsConnections](#) property
- [bibus](#) » [dispatcher](#) » [prsConnections](#) property
- [bibus](#) » [planningRuntimeService](#) » [prsConnections](#) property
- [bibus](#) » [configuration](#) » [prsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [prsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [prsPeakConnections](#) property
- [bibus](#) » [planningRuntimeService](#) » [prsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [ptsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [ptsConnections](#) property
- [bibus](#) » [dispatcher](#) » [ptsConnections](#) property

- [bibus](#) » [planningTaskService](#) » [ptsConnections](#) property
- [bibus](#) » [configuration](#) » [ptsPeakConnections](#) property
- [bibus](#) » [configurationFolder](#) » [ptsPeakConnections](#) property
- [bibus](#) » [dispatcher](#) » [ptsPeakConnections](#) property
- [bibus](#) » [planningTaskService](#) » [ptsPeakConnections](#) property
- [bibus](#) » [configuration](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [rsPeakMaximumProcesses](#) property
- [bibus](#) » [configurationFolder](#) » [rsPeakMaximumProcesses](#) property
- [bibus](#) » [dispatcher](#) » [rsPeakMaximumProcesses](#) property
- [bibus](#) » [reportService](#) » [rsPeakMaximumProcesses](#) property
- [bibus](#) » [configuration](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [rsPeakNonAffineConnections](#) property
- [bibus](#) » [reportService](#) » [rsPeakNonAffineConnections](#) property

Administration Console

Additional objects are created in the content store during initialization and upgrade. These objects are used by the Administration Console to group related tasks and provide finer-grained control over access to administration tasks.

This change affects:

- [“Capabilities” on page 1723](#)
- [“Content Administration” on page 1713](#)
- [“Current Activities” on page 1725](#)
- [“Data Source Connections” on page 1714](#)
- [“Dispatchers and Services” on page 1714](#)
- [“Distribution Lists and Contacts” on page 1715](#)
- [“Pagelets” on page 1712](#)
- [“Configuration” on page 1713](#)
- [“Console” on page 1712](#)
- [“Security” on page 1722](#)
- [“Status” on page 1725](#)
- [“Past Activities” on page 1726](#)
- [“Portlets” on page 1715](#)
- [“Printers” on page 1716](#)
- [“Schedules” on page 1727](#)
- [“Styles” on page 1717](#)
- [“System” on page 1727](#)
- [“Upcoming Activities” on page 1728](#)
- [“User Interface Profiles” on page 1723](#)
- [“Users, Groups, and Roles” on page 1724](#)

- [“Administration” on page 1711](#)
- [“Capabilities” on page 1730](#)
- [“Content Administration” on page 1730](#)
- [“Current Activities” on page 1731](#)
- [“Data Source Connections” on page 1732](#)
- [“Dispatchers and Services” on page 1733](#)
- [“Distribution Lists and Contacts” on page 1734](#)
- [“Portlets” on page 1728](#)
- [“Console” on page 1729](#)
- [“Past Activities” on page 1735](#)
- [“Portlets” on page 1735](#)
- [“Printers” on page 1736](#)
- [“cogadmin” on page 1729](#)
- [“Profiles Administration” on page 1737](#)
- [“Schedules” on page 1737](#)
- [“Styles” on page 1738](#)
- [“System” on page 1738](#)
- [“Upcoming Activities” on page 1739](#)
- [“Users Groups and Roles” on page 1739](#)

Schedule Management

This release improves the functionality related to the visibility and administration of scheduled and executing tasks.

The `bibus » history` class now contains additional information that facilitates improved reporting on scheduled tasks.

System performance has improved by maintaining information about a scheduled task that will run in the `eventManagementService` service store, information about running tasks in the `monitorService` service store, or information about completed tasks in the content store.

The `eventManagementService` service and `monitorService` service now support additional methods to retrieve information on tasks stored in these services.

This change affects:

- `asynch » runSpecification(specification, parameterValues, options)` method
- `asynch » runSpecification(specification, parameterValues, options)` method
- `paging » currentPage(conversation, parameterValues, options)` method
- `paging » currentPage(conversation, parameterValues, options)` method
- `paging » firstPage(conversation, parameterValues, options)` method
- `paging » firstPage(conversation, parameterValues, options)` method
- `paging » lastPage(conversation, parameterValues, options)` method
- `paging » lastPage(conversation, parameterValues, options)` method
- `paging » nextPage(conversation, parameterValues, options)` method
- `paging » nextPage(conversation, parameterValues, options)` method
- `paging » previousPage(conversation, parameterValues, options)` method
- `paging » previousPage(conversation, parameterValues, options)` method

- [bibus](#) » [asynchDetailEventRecord](#) class
- [bibus](#) » [eventManagementServiceSpecification](#) class
- [bibus](#) » [monitorServiceSpecification](#) class
- [bibus](#) » [pagingOption](#) class
- [bibus](#) » [pagingOptionInt](#) class
- [bibus](#) » [pagingOptionEnum](#) enumeration set
- [bibus](#) » [history](#) » [dispatcherID](#) property
- [bibus](#) » [history](#) » [maximumDetailSeverity](#) property
- [bibus](#) » [history](#) » [scheduleTriggerName](#) property
- [bibus](#) » [history](#) » [scheduleType](#) property
- [eventManagementService](#) service
- [monitorService](#) service

System Metrics

IBM Cognos 8 now defines a series of system metrics that can be used to track system performance. Threshold values can be defined for system metrics as part of the global configuration or for a dispatcher or service. These threshold values are used to enhance the reporting of the system metrics for the associated system resource.

This change affects:

- [bibus](#) » [systemMetricThresholds](#) class
- [bibus](#) » [uriValue](#) class
- [bibus](#) » [adaptiveAnalyticsService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [agentService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [annotationService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [batchReportService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [configuration](#) » [systemMetricThresholds](#) property
- [bibus](#) » [contentManagerCacheService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [contentManagerService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [dataAdvisorService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [dataIntegrationService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [dataMovementService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [deliveryService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [dimensionManagementService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [dispatcher](#) » [systemMetricThresholds](#) property
- [bibus](#) » [eventManagementService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [EVService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [graphicsService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [humanTaskService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [idVizService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [indexDataService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [indexSearchService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [indexUpdateService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [jobAndScheduleMonitoringService](#) » [systemMetricThresholds](#) property

- [bibus](#) » [jobService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [logService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [metadataService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [metricsManagerService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [migrationService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [mobileService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [monitorService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [planningAdministrationConsoleService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [planningDataService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [planningRuntimeService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [planningTaskService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [powerPlayService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [presentationService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [queryService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [relationalMetadataService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [reportDataService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [reportService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [repositoryService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [saCAMService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [systemService](#) » [systemMetricThresholds](#) property
- [bibus](#) » [systemMetricEnum](#) enumeration set
- [bibus](#) » [systemMetricThresholdsPropertyEnum](#) enumeration set

Task Retry

It is now possible to retry tasks that have failed. For example, a task may fail as the result of a database or network error. The options and parameter values associated with the original request are preserved, and are used during a retry.

Several new options were added to support retrying tasks, and to specify sub-tasks that should be skipped during the retry. In addition, the option defined by the `jobOptionEnum` enumeration set was redefined in [bibus](#) » [asynchOptionEnum](#) enumeration set to reflect the wider applicability of this option.

This change affects:

- [asynch](#) » [run\(objectPath, parameterValues, options\)](#) method
- [bibus](#) » [asynchOptionSearchPathSingleObjectArray](#) class
- [bibus](#) » [asynchOptionEnum](#) » [continueOnError](#) value
- [bibus](#) » [asynchOptionEnum](#) » [restartHistoryLocation](#) value
- [bibus](#) » [asynchOptionEnum](#) » [skipTaskHistoryLocations](#) value
- [bibus](#) » [history](#) » [restartEventID](#) property

Chart Hotspot Limit

It is now possible to limit the number of hotspots in a chart created by the [batchReportService](#) service or the [reportService](#) service.

This improves performance when rendering charts that contain a large number of members.

This change affects:

- [bibus](#) » [batchReportService](#) » [brsChartHotspotLimit](#) property
- [bibus](#) » [configuration](#) » [brsChartHotspotLimit](#) property
- [bibus](#) » [configurationFolder](#) » [brsChartHotspotLimit](#) property
- [bibus](#) » [dispatcher](#) » [brsChartHotspotLimit](#) property
- [bibus](#) » [configuration](#) » [rsChartHotspotLimit](#) property
- [bibus](#) » [configurationFolder](#) » [rsChartHotspotLimit](#) property
- [bibus](#) » [dispatcher](#) » [rsChartHotspotLimit](#) property
- [bibus](#) » [reportService](#) » [rsChartHotspotLimit](#) property

PDF Options - Password Protection

New options have been introduced to control the ability to alter or manipulate PDF documents created by IBM Cognos 8. These options can be specified when running reports and allow for more granular control of accessibility options and print quality, and enforce control of these options using password protection on a PDF document.

You can use the [bibus](#) » [pdfOptionEnum](#) enumeration set to specify access controls anywhere other options are specified. For example, use [bibus](#) » [baseReport](#) » [options](#) property to specify PDF options for a report.

This change affects:

- [bibus](#) » [pdfOption](#) class
- [bibus](#) » [pdfOptionBoolean](#) class
- [bibus](#) » [pdfOptionPrintQuality](#) class
- [bibus](#) » [pdfOptionXMLEncodedXML](#) class
- [bibus](#) » [pdfOptionEnum](#) enumeration set
- [bibus](#) » [pdfPrintQualityEnum](#) enumeration set

spreadsheetML Output Format

It is now possible to generate report output in SpreadsheetML (Microsoft® Excel Office Open XML) format.

Use the [bibus](#) » [outputFormatEnum](#) » [spreadsheetML](#) value when specifying the [outputFormat](#) run option.

This change affects:

- [bibus](#) » [outputFormatEnum](#) » [spreadsheetML](#) value

Reporting Profiles

Users can now specify their preferred Reporting profile from a predefined set of customized UI profiles. The specified profile determines the Reporting user interface for each user. For example, an administrator can make available a profile for the specific needs of Financial report authors, so the user only sees the interface features required for that role.

Two user interface profile folders are constructed during upgrade/content store initialization. These folders organize the profiles. A user can select a profile if the user has `read` permission on the [bibus](#) » [uiProfile](#), and `traverse` permission on all of its ancestors.

This change affects:

- [bibus](#) » [reportStudioOption](#) class

- [bibus » reportStudioOptionSearchPathSingleObject](#) class
- [bibus » uiProfile](#) class
- [bibus » uiProfileFolder](#) class
- [bibus » reportStudioOptionEnum](#) enumeration set
- “Express - deprecated” on page 1679
- “Reporting Profiles - deprecated” on page 1678
- “User Interface Profiles - deprecated” on page 1678
- “Professional - deprecated” on page 1680

Disabling Selection-based Interactive Report Output Features

You can now disable selection-based features in interactive report output by specifying the run option `selectionBasedFeatures` with a value of `false`.

The default value for this option is `true`.

Disabling selection-based features turns off package drill-through and IBM Cognos Search features. This improves system throughput for large reports.

This change affects:

- [bibus » runOptionEnum » selectionBasedFeatures](#) value

Improved Context Metadata for Selection

Context information can now be obtained from the report server on demand rather than always being provided as part of the rendered output page, thereby improving overall system performance.

This change affects:

- [report » getContext\(conversation, parameterValues, options\)](#) method
- [report » getObjectContext\(objectPath, parameterValues, options\)](#) method
- [bibus » asynchDetailContext](#) class
- [bibus » contextOption](#) class
- [bibus » contextOptionFormat](#) class
- [bibus » contextOptionStringArray](#) class
- [bibus » contextOptionType](#) class
- [bibus » contextFormatEnum](#) enumeration set
- [bibus » contextOptionEnum](#) enumeration set
- [bibus » contextTypeEnum](#) enumeration set
- [bibus » documentContent » context](#) property
- [bibus » output » context](#) property
- [bibus » documentContent » contextBlockCount](#) property
- [bibus » output » contextBlockCount](#) property

E-Mail Activity and Administration

The run status `inactive` has been introduced to reflect the status of tasks that are inactive until a service can process them.

An e-mail task may be placed in this state by the [deliveryService](#) if the SMTP mail server is not available. When the [deliveryService](#) detects that the mail server is available, processing of the email task will continue.

This change affects:

- [bibus](#) » [runStatusEnum](#) » [inactive](#) value
- [monitor](#) method set

Schedule Priority

Users can now assign a priority to a schedule. Tasks with a higher priority run before tasks with a lower priority. This enables management of reports submitted to run, but are at a pending state in the job queue. Task priority values range from 1 to 5 (inclusive). 1 is the highest priority; 5 is the lowest. The default value is 3.

The capability [canUseSchedulingPriority](#) controls access to schedule priorities. Administrators can grant this capability by updating the [policies](#) property of the “Scheduling Priority” on page 1665.

The priority of a queued task can be updated using the [event](#) » [updateEvents\(events\)](#) method.

This change affects:

- [event](#) » [updateEvents\(events\)](#) method
- [bibus](#) » [userCapabilityEnum](#) » [canUseSchedulingPriority](#) value
- [event](#) method set
- “Scheduling Priority” on page 1665
- [bibus](#) » [asynchDetailEventRecord](#) » [priority](#) property
- [bibus](#) » [eventRecord](#) » [priority](#) property
- [bibus](#) » [schedule](#) » [priority](#) property

Run Agent Tasks on Failure

It is now possible to control the execution of agent tasks based on the [bibus](#) » [runStatusEnum](#) of other agent tasks.

In particular, tasks can be made to run only if other agent tasks have succeeded or failed. It is also possible to specify that tasks run regardless of the [bibus](#) » [runStatusEnum](#) of other tasks.

This change affects:

- [bibus](#) » [runConditionEnum](#) enumeration set
- [bibus](#) » [agentTaskDefinition](#) » [runCondition](#) property

Advanced Features for Report Output

New advanced features have been introduced to evaluate conditional subscriptions and to import saved output versions into Office Connection. The [advancedOutput](#) run option was added to control whether these and other advanced features for output versions are enabled.

This change affects:

- [bibus](#) » [runOptionEnum](#) » [advancedOutput](#) value
- [bibus](#) » [outputFormatEnum](#) » [layoutDataXML](#) value

Drill Through Assistant Capability

The capability `canUseDrillThroughAssistant` has been added to control access to the Drill Through Assistant. Administrators will be able to grant this capability by updating the `policies` property of the “Drill Through Assistant” on page 1638.

This change affects:

- `bibus » userCapabilityEnum » canUseDrillThroughAssistant` value
- “Drill Through Assistant” on page 1638

Package Hierarchies

You can now organize packages into hierarchies. Both `bibus » folder` and `bibus » package` objects can now contain `bibus » package` objects.

The `bibus » agentTaskDefinition » packageBase` property, `bibus » baseAgentDefinition » packageBase` property, `bibus » baseDataIntegrationTask » packageBase` property, `bibus » basePowerPlayClass » packageBase` property, `bibus » baseReport » packageBase` property, `bibus » folder » packageBase` property, `bibus » jobDefinition » packageBase` property, `bibus » jobStepDefinition » packageBase` property, `bibus » model » packageBase` property, `bibus » package » packageBase` property, `bibus » planningApplication » packageBase` property, `bibus » planningEList » packageBase` property, and `bibus » URL » packageBase` property are now deprecated. Use the `bibus » baseClass » ancestors` property instead.

Objects can now acquire the `bibus » account » routingHints` property, `bibus » group » routingHints` property, `bibus » package » routingHints` property, and `bibus » role » routingHints` property from a container object.

The deployment system now supports deployment of packages in hierarchies. The existing deployment option `package` has been deprecated. The deployment options `export` and `import` have been added to replace it. These options describe how Content Manager *exports* objects from the content store to the archive and *imports* objects from the archive to the content store.

In an SDK application that has not yet been upgraded to use the new options, the deprecated `package` option can be used only to import objects from an archive created with an earlier version of IBM Cognos 8 or by an SDK application that still uses the `package` option. An upgraded SDK application importing objects from such archives should use the `upgradeArchiveOptions` option to upgrade the options in the archive to their current equivalents.

The option information was also added to replace the `package` option. The `deployment » getDeploymentOptions(archive, options)` method uses it to return information about objects in the archive. Users should not construct this type of option.

The `bibus » pageletInstance » deploymentReferences` property, `bibus » portlet » deploymentReferences` property, and `bibus » portletInstance » deploymentReferences` property were added to support the deployment of these types of objects.

This change affects:

- `bibus » deploymentImportRule` class
- `bibus » deploymentObjectInformation` class
- `bibus » deploymentOptionImportRuleArray` class
- `bibus » deploymentOptionObjectInformationArray` class
- `bibus » deploymentOptionPackageInfo` class
- `bibus » deploymentOptionSearchPathSingleObjectArray` class
- `bibus » deploymentReference` class
- `bibus » pageletInstance » deploymentReferences` property

- [bibus](#) » [portlet](#) » [deploymentReferences](#) property
- [bibus](#) » [portletInstance](#) » [deploymentReferences](#) property
- [bibus](#) » [folder](#) » [items](#) property
- [bibus](#) » [package](#) » [items](#) property
- [bibus](#) » [account](#) » [routingHints](#) property
- [bibus](#) » [group](#) » [routingHints](#) property
- [bibus](#) » [package](#) » [routingHints](#) property
- [bibus](#) » [role](#) » [routingHints](#) property
- [bibus](#) » [deploymentOptionEnum](#) » [export](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [import](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [information](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [package](#) value
- [bibus](#) » [deploymentOptionEnum](#) » [upgradeArchiveOptions](#) value
- [bibus](#) » [agentTaskDefinition](#) » [packageBase](#) property
- [bibus](#) » [baseAgentDefinition](#) » [packageBase](#) property
- [bibus](#) » [baseDataIntegrationTask](#) » [packageBase](#) property
- [bibus](#) » [basePowerPlayClass](#) » [packageBase](#) property
- [bibus](#) » [baseReport](#) » [packageBase](#) property
- [bibus](#) » [folder](#) » [packageBase](#) property
- [bibus](#) » [jobDefinition](#) » [packageBase](#) property
- [bibus](#) » [jobStepDefinition](#) » [packageBase](#) property
- [bibus](#) » [model](#) » [packageBase](#) property
- [bibus](#) » [package](#) » [packageBase](#) property
- [bibus](#) » [planningApplication](#) » [packageBase](#) property
- [bibus](#) » [planningEList](#) » [packageBase](#) property
- [bibus](#) » [URL](#) » [packageBase](#) property

Object Icons

It is now possible to associate an alternate graphic resource with a [bibus](#) » [uiClass](#) object in the content store. The graphic is used when rendering the object in the portal, replacing the default Cognos8 icon.

This change affects:

- [bibus](#) » [uiClass](#) » [iconURI](#) property

Dashboard Fragments

The documentation was updated to reflect the current set of predefined objects for IBM Cognos 8 as follows:

- Some predefined portal objects have been renamed.
- New objects have been added to support new portal functionality.
- Previously undocumented portal objects have been added.

This change affects:

- “HTML Source” on [page 1754](#)

- [“IBM Cognos History Chart” on page 1752](#)
- [“IBM Cognos Metric List” on page 1753](#)
- [“Multi-page” on page 1748](#)
- [“Dashboard” on page 1747](#)
- [“IBM Cognos Metric Studio” on page 1751](#)

Deferring Creation of History Objects

To improve system performance, [bibus » history](#) objects are now created in the content store when the associated task has completed.

You can find information about queued tasks from the [eventManagementService](#) service using the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

You can find information about running tasks from the [monitorService](#) service using the [asynch » runSpecification\(specification, parameterValues, options\)](#) method.

This change affects:

- [monitor](#) method set

Deployment History

To improve system performance, users can determine the amount of history information created during a deployment by using the [recordingLevel](#) deployment option.

In addition, the history detail class [bibus » historyDetailDeploymentSummary](#) was added to record summary information related to deployments, such as the number of objects added, deleted, replaced and updated.

This change affects:

- [bibus » deploymentOptionAuditLevel](#) class
- [bibus » historyDetailDeploymentSummary](#) class
- [bibus » deploymentOptionEnum » recordingLevel](#) value

Parameter Method Set

The [parameter](#) method set was introduced to group methods related to obtaining parameter metadata.

This change affects:

- [parameter](#) method set
- [batchReportService](#) service
- [reportService](#) service

PDF Configuration Parameters

New configuration parameters have been introduced to control the production of PDF documents by the [batchReportService](#) and [reportService](#).

These parameters control PDF production variables such as the character encodings, embedded fonts, and compression used for the document.

This change affects:

- [bibus](#) » [pdfCharacterEncodingEnum](#) enumeration set
- [bibus](#) » [pdfCompressionTypeEnum](#) enumeration set
- [bibus](#) » [pdfFontEmbeddingEnum](#) enumeration set
- [bibus](#) » [batchReportService](#) » [brsPDFCharacterEncoding](#) property
- [bibus](#) » [configuration](#) » [brsPDFCharacterEncoding](#) property
- [bibus](#) » [configurationFolder](#) » [brsPDFCharacterEncoding](#) property
- [bibus](#) » [dispatcher](#) » [brsPDFCharacterEncoding](#) property
- [bibus](#) » [batchReportService](#) » [brsPDFCompressionLevel](#) property
- [bibus](#) » [configuration](#) » [brsPDFCompressionLevel](#) property
- [bibus](#) » [configurationFolder](#) » [brsPDFCompressionLevel](#) property
- [bibus](#) » [dispatcher](#) » [brsPDFCompressionLevel](#) property
- [bibus](#) » [batchReportService](#) » [brsPDFCompressionType](#) property
- [bibus](#) » [configuration](#) » [brsPDFCompressionType](#) property
- [bibus](#) » [configurationFolder](#) » [brsPDFCompressionType](#) property
- [bibus](#) » [dispatcher](#) » [brsPDFCompressionType](#) property
- [bibus](#) » [batchReportService](#) » [brsPDFEmbedFonts](#) property
- [bibus](#) » [configuration](#) » [brsPDFEmbedFonts](#) property
- [bibus](#) » [configurationFolder](#) » [brsPDFEmbedFonts](#) property
- [bibus](#) » [dispatcher](#) » [brsPDFEmbedFonts](#) property
- [bibus](#) » [configuration](#) » [rsPDFCharacterEncoding](#) property
- [bibus](#) » [configurationFolder](#) » [rsPDFCharacterEncoding](#) property
- [bibus](#) » [dispatcher](#) » [rsPDFCharacterEncoding](#) property
- [bibus](#) » [reportService](#) » [rsPDFCharacterEncoding](#) property
- [bibus](#) » [configuration](#) » [rsPDFCompressionLevel](#) property
- [bibus](#) » [configurationFolder](#) » [rsPDFCompressionLevel](#) property
- [bibus](#) » [dispatcher](#) » [rsPDFCompressionLevel](#) property
- [bibus](#) » [reportService](#) » [rsPDFCompressionLevel](#) property
- [bibus](#) » [configuration](#) » [rsPDFCompressionType](#) property
- [bibus](#) » [configurationFolder](#) » [rsPDFCompressionType](#) property
- [bibus](#) » [dispatcher](#) » [rsPDFCompressionType](#) property
- [bibus](#) » [reportService](#) » [rsPDFCompressionType](#) property
- [bibus](#) » [configuration](#) » [rsPDFEmbedFonts](#) property
- [bibus](#) » [configurationFolder](#) » [rsPDFEmbedFonts](#) property
- [bibus](#) » [dispatcher](#) » [rsPDFEmbedFonts](#) property
- [bibus](#) » [reportService](#) » [rsPDFEmbedFonts](#) property

Email Delivery Enhancements

It is now possible to compress large email attachments before sending them using the delivery service.

The properties [bibus](#) » [configuration](#) » [dsCompressAttachmentLimit](#), [bibus](#) » [configurationFolder](#) » [dsCompressAttachmentLimit](#), [bibus](#) » [deliveryService](#) » [dsCompressAttachmentLimit](#), and [bibus](#) » [dispatcher](#) » [dsCompressAttachmentLimit](#) define the maximum size, in MB, of an uncompressed email attachment sent by the delivery service. Attachments that exceed the size limit will be compressed before being sent.

Use a value of 0 (default) to disable email attachment compression.

Use a nil value to compress all email attachments.

Setting a non-nil, non-zero value improves performance when sending email messages with large attachments, such as report outputs.

This change affects:

- [bibus](#) » [configuration](#) » [dsCompressAttachmentLimit](#) property
- [bibus](#) » [configurationFolder](#) » [dsCompressAttachmentLimit](#) property
- [bibus](#) » [deliveryService](#) » [dsCompressAttachmentLimit](#) property
- [bibus](#) » [dispatcher](#) » [dsCompressAttachmentLimit](#) property

Data Movement Service Configuration Parameters

New configuration parameters have been introduced for the [dataMovementService](#).

This change affects:

- [bibus](#) » [configuration](#) » [dmsAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsConnections](#) property
- [bibus](#) » [configuration](#) » [dmsExecutionTimeLimit](#) property
- [bibus](#) » [configurationFolder](#) » [dmsExecutionTimeLimit](#) property
- [bibus](#) » [dataMovementService](#) » [dmsExecutionTimeLimit](#) property
- [bibus](#) » [dispatcher](#) » [dmsExecutionTimeLimit](#) property
- [bibus](#) » [configuration](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsPeakAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsPeakNonAffineConnections](#) property
- [bibus](#) » [configurationFolder](#) » [dmsPeakNonAffineConnections](#) property
- [bibus](#) » [dataMovementService](#) » [dmsPeakNonAffineConnections](#) property
- [bibus](#) » [dispatcher](#) » [dmsPeakNonAffineConnections](#) property
- [bibus](#) » [configuration](#) » [dmsQueueLimit](#) property
- [bibus](#) » [configurationFolder](#) » [dmsQueueLimit](#) property
- [bibus](#) » [dataMovementService](#) » [dmsQueueLimit](#) property
- [bibus](#) » [dispatcher](#) » [dmsQueueLimit](#) property

Capabilities Refinements

Additional administration capabilities have been defined to secure tasks in the Administration Console. The capability `canUseDirectoryTool` is obsolete and has been removed. It has been replaced by four new capabilities to provide more precise control over the tasks it secured. These new capabilities are `canUseDataSourcesTool`, `canUseDistributionListsAndContactsTool`, `canUsePrintersTool`, and `canUseUsersGroupsAndRolesTool`.

The “Directory - obsolete” on page 1621 is also obsolete and has been removed. It has been replaced by four new objects, one for each of the new capabilities. These new objects are the “Data Source Connections” on page 1620, the “Distribution Lists and Contacts” on page 1621, the “Printers” on page 1625, and the “Users, Groups, and Roles” on page 1629.

This change affects:

- `report` » `testDataSourceConnection(connectionString, credentials)` – obsolete method
- `bibus` » `userCapabilityEnum` » `canUseDataSourcesTool` value
- `bibus` » `userCapabilityEnum` » `canUseDirectoryTool` value
- `bibus` » `userCapabilityEnum` » `canUseDistributionListsAndContactsTool` value
- `bibus` » `userCapabilityEnum` » `canUseMonitorActivityTool` value
- `bibus` » `userCapabilityEnum` » `canUsePrintersTool` value
- `bibus` » `userCapabilityEnum` » `canUseUsersGroupsAndRolesTool` value
- “Distribution Lists and Contacts” on page 1621
- “Printers” on page 1625
- “Data Source Connections” on page 1620
- “Directory - obsolete” on page 1621
- “Run activities and schedules” on page 1626
- “Users, Groups, and Roles” on page 1629
- `bibus` » `pagelet` » `metadata` property

Client Token

The properties `authenticityToken` and `userSessionID` were added to the `bibus` » `CAM` class to provide protection against cross-site request forgery attacks.

The `bibus` » `CAM` » `userSessionID` property is calculated when the user logs on to IBM Cognos 8. The value of this property is maintained in a browser cookie.

The `bibus` » `CAM` » `authenticityToken` property is a pseudo-random value computed in the browser by applying the SHA-1 hash algorithm to the `userSessionID` value found in the cookie.

Server-side components recalculate the `authenticityToken` property and compare it to the value provided in the request. If the two values do not match the request is considered to be malicious and is not processed.

This change affects:

- `bibus` » `CAM` » `authenticityToken` property
- `bibus` » `CAM` » `userSessionID` property

New Security Roles

Two new default security roles have been added to the Cognos namespace to assist IBM Cognos Analytics administrators with configuring permissions and capabilities for users.

The “Readers” on page 1707 role identifies users who are allowed to read content stored in Public Folders but who are not allowed to update content.

The “Express Authors” on page 1698 role identifies users who are allowed to use the “Express - deprecated” on page 1679 IBM Cognos Analytics - Reporting profile.

The initial security policies for several objects were updated to incorporate policies for the new roles.

This change affects:

- “Public Folders” on page 1759
- “Express Authors” on page 1698
- “Readers” on page 1707
- “Bursting” on page 1658
- “HTML Items in Report” on page 1659
- “Run With Options” on page 1633
- “User Defined SQL” on page 1661
- “Cognos Viewer” on page 1631
- “Reporting” on page 1657
- “Scheduling” on page 1661
- “Watch Rules” on page 1669
- “Express - deprecated” on page 1679
- “Reporting Profiles - deprecated” on page 1678
- “Professional - deprecated” on page 1680

Default Value Change

To facilitate the use of alerts, the default value for the [bibus » baseAgentDefinition » allowNotification](#) property has been changed to `true`.

This change affects:

- [bibus » baseAgentDefinition » allowNotification](#) property

New Properties for Group and Role Classes

New properties were added to the [bibus » group](#) and [bibus » role](#) classes.

The [distributionMembers](#) property of the [bibus » group](#) and the [distributionMembers](#) property of the [bibus » role](#) return the subset of the [members](#) property of the [bibus » group](#) or the [members](#) property of the [bibus » role](#) that are used to distribute content such as report outputs.

The [securityMembers](#) property of the [bibus » group](#) and the [securityMembers](#) property of the [bibus » role](#) return the subset of the [members](#) property of the [bibus » group](#) or the [members](#) property of the [bibus » role](#) that are used to resolve security authorization queries. These properties are used when resolving security policies, for example.

New functions “[expandDistributionMembers\(object_set\)](#)” on page 1571 and “[expandSecurityMembers\(object_set\)](#)” on page 1572 were added. The function “[expandMembers\(object_set\)](#)” on page 1572 was deprecated.

This change affects:

- [bibus » group » distributionMembers](#) property
- [bibus » role » distributionMembers](#) property

- [bibus](#) » [group](#) » [members](#) property
- [bibus](#) » [role](#) » [members](#) property
- [bibus](#) » [group](#) » [securityMembers](#) property
- [bibus](#) » [role](#) » [securityMembers](#) property
- “[expandDistributionMembers\(object_set\)](#)” on page 1571
- “[expandMembers\(object_set\)](#)” on page 1572
- “[expandSecurityMembers\(object_set\)](#)” on page 1572

Notification List

Documentation has been updated to indicate that the [bibus](#) » [baseAgentDefinition](#) » [notificationList](#) property cannot be accessed using the BI Bus API.

SDK clients that require access to this property must use the [deliveryService](#) methods [delivery](#) » [addNotification\(objectPath\)](#), [delivery](#) » [clearNotifications\(objectPath\)](#), [delivery](#) » [deleteAllNotifications\(\)](#), [delivery](#) » [deleteNotification\(objectPath\)](#) and [delivery](#) » [queryNotification\(objectPath\)](#).

This change affects:

- [bibus](#) » [baseAgentDefinition](#) » [notificationList](#) property

mruFolder Retention Rules

The default retention rule for items in the user's [bibus](#) » [mruFolder](#) has been changed to use the [bibus](#) » [baseClass](#) » [modificationTime](#) property instead of the [bibus](#) » [baseClass](#) » [creationTime](#) property.

This change supports internal performance optimizations.

Existing retention rules are updated during the upgrade of the content store and during an import of a deployment.

This change affects:

- [bibus](#) » [mruFolder](#) » [items](#) property

Search Capability

The documentation incorrectly specified the class of the “[Execute Indexed Search](#)” on page 1640 as [bibus](#) » [securedFeature](#). It has been corrected to specify that this object is an instance of [bibus](#) » [securedFunction](#).

This change affects:

- “[Execute Indexed Search](#)” on page 1640

Search – For Internal Use Only

Documentation for services, classes, methods, and enumeration values related to search integration has been updated to indicate that they are included to support internal functionality only. Do not use these extensions.

This change affects:

- [indexTerm](#) » [addTermAssociation\(term, parameterValues, options\)](#) method
- [indexTerm](#) » [deleteTermAssociation\(term, parameterValues, options\)](#) method
- [indexUpdate](#) » [add\(objectPath, parameterValues, options\)](#) method

- [indexUpdate](#) » [delete\(objectPath, parameterValues, options\)](#) method
- [indexUpdate](#) » [get\(objectPath, parameterValues, options\)](#) method
- [bibus](#) » [asynchDetailCount](#) class
- [bibus](#) » [asynchDetailIndexData](#) class
- [bibus](#) » [asynchDetailIndexTerms](#) class
- [bibus](#) » [indexOption](#) class
- [bibus](#) » [indexOptionBoolean](#) class
- [bibus](#) » [indexOptionInt](#) class
- [bibus](#) » [indexSearchServiceSpecification](#) class
- [bibus](#) » [indexTerm](#) class
- [bibus](#) » [indexTermOption](#) class
- [bibus](#) » [indexTermOptionBoolean](#) class
- [bibus](#) » [indexTermOptionIndexTermArray](#) class
- [bibus](#) » [indexUpdateServiceSpecification](#) class
- [bibus](#) » [indexUpdateTask](#) class
- [bibus](#) » [indexOptionEnum](#) enumeration set
- [bibus](#) » [indexTermOptionEnum](#) enumeration set
- [indexTerm](#) method set
- [indexUpdate](#) method set
- [indexSearchService](#) service
- [indexUpdateService](#) service

Notices

This information was developed for products and services offered worldwide.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
Attention: Licensing

3755 Riverside Dr.
Ottawa, ON
K1V 1B7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's

- name
- user name
- password

for purposes of

- session management
- authentication
- enhanced user usability
- single sign-on configuration
- usage tracking or functional purposes other than session management, authentication, enhanced user usability and single sign-on configuration

These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <https://www.ibm.com/privacy/us/en/>.

Glossary

This glossary includes terms and definitions for IBM Cognos Software Development Kit.

The following cross-references are used in this glossary:

- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to [IBM Terminology](http://www.ibm.com/software/globalization/terminology) (<http://www.ibm.com/software/globalization/terminology>).

[A](#) [B](#) [C](#) [D](#) [F](#) [G](#) [H](#) [I](#) [J](#) [L](#) [M](#) [N](#) [P](#) [Q](#) [R](#) [S](#) [U](#)

A

access permission

A privilege that permits the access or use of an object.

affinity

In server processing, a property of a request that indicates how important it is that successive requests are allocated to the same server process and thread.

aggregate

A calculation that returns a single result value from several relational data rows or dimensional members. Typical examples are total and average.

alias

An alternative name used instead of a primary name.

anonymous access

A type of access that allows users and servers to access a server without first authenticating with it.

authentication

The process of validating the identity of a user or server.

authentication provider

The communication mechanism to an external authentication source. Functionalities, such as user authentication, group membership, and namespace searches, are made available through authentication providers.

axis

A part of a location step in a search path. An axis defines the direction of the search for objects in the object hierarchy of the content store. An axis indicates where to search for objects by defining a set of objects relative to the current context object. For example, the ancestor axis specifies the ancestor of the current context object.

B

BI Bus

See [Business Intelligence Bus](#).

burst

To create several report results by running a single report once. For example, the user can create a report that shows sales for each employee, and run it once, sending different results to regional managers by bursting on region.

Business Intelligence Bus (BI Bus)

An application programming interface (API) that uses SOAP and other standard protocols for message encoding, transport, and security. Using any compliant toolkit, the BI Bus can be integrated with other web portals and software applications.

C

calculated member

A member of a dimension whose measure values are not stored but are calculated at run time using an expression.

capability

A group of functions and features that can be hidden or revealed to simplify the user interface. Capabilities can be enabled or disabled by changing preference settings, or they can be controlled through an administration interface.

cascading prompt

A prompt that uses values from a previous prompt to filter the values in the current prompt or pick list.

CGI

See [Common Gateway Interface](#).

CM

See [Content Manager](#).

Common Gateway Interface (CGI)

An Internet standard for defining scripts that pass information from a web server to an application program, through an HTTP request, and vice versa.

condition

An expression that can be evaluated as true, false, or unknown. It can be expressed in natural language text, in mathematically formal notation, or in a machine-readable language.

content locale

A code that is used to set the language or dialect used for browsers and report text, and the regional preferences, such as formats for time, date, money, money expressions, and time of day.

Content Manager (CM)

The service that retrieves information from the content store, and saves information to the content store.

content store

The database that contains the data needed to operate, such as report specifications, published models, and security rights.

conversation

A series of related messages between client and server components of a system, such as the request/response exchange between a browser and a server.

credential

A set of information that grants a user or process certain access rights.

cube

A multidimensional representation of data needed for online analytical processing, multidimensional reporting, or multidimensional planning applications.

D

data source

The source of data itself, such as a database or XML file, and the connection information necessary for accessing the data.

deployment

The process of moving an application (such as a report or model) to a different instance. For example, reports are often created in a test environment and then deployed to production. When an application is deployed, it is exported, transferred, and imported.

deployment specification

A definition of what packages to move (deploy) between source and target environments, the deployment preferences, and the archive name. Deployment specifications are used for import and export.

dimension

A broad grouping of descriptive data about a major aspect of a business, such as products, dates, or locations. Each dimension includes different levels of members in one or more hierarchies and an optional set of calculated members or special categories.

dimensional data source

A data source containing data modeled using OLAP concepts, including dimensions, hierarchies, and measures.

F

fact

See [measure](#).

G

gateway

An extension of a web server program that transfers information from the web server to another server. Gateways are often CGI programs, but may follow other standards such as ISAPI and Apache modules.

governor

A set of rules to limit user activities, such as the execution of reports, that either take too long or consume too many resources.

group

A collection of users who can share access authorities for protected resources.

H

hierarchy

The organization of a set of entities into a tree structure, with each entity (except the root) having one or more parent entities and an arbitrary number of child entities.

I

item

See [member](#).

J

job step

The smallest part of a job that can be run separately. A job step can be a report or it can be another job.

L

layout

The arrangement of printed matter on a screen or page, including margins, line spacing, type specification, header and footer information, indents, and more.

level

A set of entities or members that form one section of a hierarchy in a dimension and represent the same type of object. For example, a geographical dimension might contain levels for region, state, and city.

locale

A setting that identifies language or geography and determines formatting conventions such as collation, case conversion, character classification, the language of messages, date and time representation, and numeric representation.

location step

A part of a search path that consists of an axis (either implied or explicit), an object test, and zero or more predicates. A search path can have multiple location steps. They are used to select a set of objects that function as the context in which to evaluate the next location step.

M

MDX

See [Multidimensional Expression Language](#).

measure

A performance indicator that is quantifiable and used to determine how well a business is operating. For example, measures can be Revenue, Revenue/Employee, and Profit Margin percent.

member

A unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products hierarchy. See also [member unique name](#).

member unique name (MUN)

A path of member names, one from each level in a hierarchy, defining the exact location of the member from either an OLAP data source or a dimensionally modeled relational source. For example, Geography.Europe.France.Paris uniquely identifies Paris, France, distinguishing it from other instances of Paris in the City level. See also [member](#).

model

A physical or business representation of the structure of the data from one or more data sources. A model describes data objects, structure, and grouping, as well as relationships and security. In Cognos Analytics, a model is created and maintained in Framework Manager. The model or a subset of the model must be published to the Cognos server as a package for users to create and run reports.

Multidimensional Expression Language (MDX)

The multidimensional equivalent of Structured Query Language (SQL).

MUN

See [member unique name](#).

N

namespace

1. A part of the model in which the names may be defined and used. Within a namespace, each name has a unique meaning.
2. For authentication and access control, a configured instance of an authentication provider that allows access to user and group information. In Framework Manager, namespaces uniquely identify query items and query subjects. Different databases are imported into separate namespaces to avoid duplicate names.

P

package

A subset of a model, which can be the whole model, to be made available to the Cognos server.

passport

Session-based information, stored and encrypted in Content Manager memory, regarding authenticated users. A passport is created the first time a user accesses Cognos 8, and it is retained until a session ends, either when the user logs off or after a specified period of inactivity.

policy specification

Information stored as an attribute of an object in the content store that defines what users and groups are authorized to use the object, and in what capacity.

predicate

The part of a node test that attempts to identify a node in a path.

product locale

The code or setting that specifies which language, regional settings, or both to use for parts of the product interface, such as menu commands.

project

In Framework Manager, a set of models, packages, and related information for administration, and for sharing model information.

prompt

A report element that asks for parameter values before the report is run.

Q

query

A specification for a set of data retrieved from a data source.

query item

A representation of a column of data in a data source. Query items may appear in a model or in a report and contain a reference to a database column, a reference to another query item, or a calculation.

R

repeater

In Reporting, a cell container that repeats values within itself with no predefined internal structure.

report specification

An executable definition of a report, including query and layout rules, which can be combined with data to produce a report output.

S

security provider

See [authentication provider](#).

session

The time during which an authenticated user is logged on.

style sheet

A specification of formatting instructions that, when applied to structured information, provides a particular rendering of that information (for example, online or printed). Different style sheets can be applied to the same piece of structured information to produce different presentations of the information.

summary

In reporting and analysis, an aggregate value that is calculated for all the values of a particular level or dimension. Examples of summaries include total, minimum, maximum, average, and count.

U

user

Any individual, organization, process, device, program, protocol, or system that uses the services of a computing system.

Index

Special Characters

.jar [20](#)

A

abstract classes

sources of inherited properties [56](#)

access manager will authenticate an external namespace [115](#)

access permissions [27](#)

access XML in Reporting [1460](#)

accessEnum enumeration set [1242](#)

account class [397](#)

accounts

locating in content store [1565](#)

action element [1541](#)

actions element [1557](#)

Actions element [1765](#)

activate(searchPath) method [171](#)

activateURI(uri) method [172](#)

activating

Content Manager service [87](#)

active server pages (ASP)

samples [1419](#)

adaptiveAnalyticsService class [406](#)

add(objectPath, parameterValues, options) method [176](#)

add(parentPath, object, options) method [177](#)

add(parentPath, objects, options) method [173](#)

addAnnotations(containerPath, objects, options) method [181](#)

addDrillPath(parentPath, object, options) method [183](#)

adding

folders [61](#)

adding a filter sample [1491](#)

adding a prompt sample [1491](#)

addNotification(objectPath) method [185](#)

addOptions class [410](#)

addressSMTP class [411](#)

addressSMTPArrayProp class [412](#)

addressSMTPProp class [412](#)

addTermAssociation(term, parameterValues, options) method [187](#)

adminFolder class [412](#)

administration

IBM® Cognos® content store [55](#)

IBM® Cognos® servers [85](#)

advanced configuration settings

RSVP.RENDER.VALIDATEURL [1434](#)

advanced routing

configuring [90](#)

using [92](#)

advanced settings

configuration [1577](#)

agent method set [159](#)

agentDefinition class [414](#)

agentDefinitionView class [415](#)

agentNotificationStatusEnum enumeration set [1242](#)

agentOption class [416](#)

agentOptionBoolean class [416](#)

agentOptionEnum enumeration set [1242](#)

agentOutputEnum enumeration set [1243](#)

agentOutputEnumProp class [416](#)

agentOutputHotList class [417](#)

agents

running [76](#)

agentService class [131](#), [418](#)

agentState class [420](#)

agentTaskDefinition class [421](#)

agentTaskState class [424](#)

aggregate attribute [1486](#)

aggregates

regular [1486](#)

reports [1486](#)

rollup [1486](#)

setting type [1486](#)

using model settings [1486](#)

aliasLocation class – obsolete [425](#)

aliasRoot class [425](#)

alternateItems element [1541](#)

analysis class [426](#)

ancestorInfo class [427](#)

ancestorInfoArrayProp class [428](#)

ancestorInfoProp class [428](#)

annotation class [429](#)

annotationFolder class [430](#)

annotationService class [431](#)

anonymous access [26](#)

anonymous logon [109](#)

anyTypeMIMEProp class [433](#)

anyTypeProp class [433](#)

anyURIArrayProp class [435](#)

anyURIProp class [435](#)

Apache Axis logging [115](#)

archiveConflictResolutionEnum enumeration set [1243](#)

archiveDescriptor class [436](#)

archiveLocation class [438](#)

archiveOption class [439](#)

archiveOptionConflictResolution class [440](#)

archiveOptionDescriptor class [440](#)

archiveOptionEnum enumeration set [1244](#)

archiveOptionSearchPathSingleObject class [441](#)

archiveOptionString class [441](#)

argument element [1557](#)

arguments

context [1499](#)

function [1498](#)

result [1498](#)

arguments element [1558](#)

ascending order [1473](#)

associate columns [1471](#)

asynch method set [159](#)

asynchDetail class [442](#)

asynchDetailAgentNotificationStatus class [442](#)

- asynchDetailAsynchSpecification class [443](#)
- asynchDetailContext class [443](#)
- asynchDetailCount class [444](#)
- asynchDetailDrillPathObject class [444](#)
- asynchDetailDrillThroughRequest class [445](#)
- asynchDetailDrillThroughTarget class [446](#)
- asynchDetailDrillThroughTargetURI class [448](#)
- asynchDetailEventID class [449](#)
- asynchDetailEventRecord class [449](#)
- asynchDetailIndexData class [453](#)
- asynchDetailIndexTerms class [455](#)
- asynchDetailMessages class [455](#)
- asynchDetailMIMEAttachment class [456](#)
- asynchDetailParameters class [457](#)
- asynchDetailParameterValues class [457](#)
- asynchDetailPromptPage class [458](#)
- asynchDetailReportMetadata class [458](#)
- asynchDetailReportObject class [458](#)
- asynchDetailReportOutput class [459](#)
- asynchDetailReportStatus class [460](#)
- asynchDetailReportStatusEnum enumeration set [1245](#)
- asynchDetailReportValidation class [461](#)
- asynchDetailROLAPDataSourceState class [461](#)
- asynchDetailROLAPMessages class [462](#)
- asynchDetailROLAPMetrics class [462](#)
- asynchDetailSelectionContext class [463](#)
- asynchDetailUnstructuredData class [463](#)
- asynchOption class [464](#)
- asynchOptionBoolean class [464](#)
- asynchOptionEncoding class [465](#)
- asynchOptionEnum enumeration set [1245](#)
- asynchOptionInt class [465](#)
- asynchOptionSearchPathSingleObject class [465](#)
- asynchOptionSearchPathSingleObjectArray class [466](#)
- asynchOptionStringArray class [466](#)
- asynchReply class [466](#)
- asynchReplyStatusEnum enumeration set [1247](#)
- asynchRequest class [468](#)
- asynchronous conversations
 - processing constraints [83](#)
 - typical [80](#)
- asynchronous methods
 - primary requests [78](#)
 - secondary requests [78](#)
- asynchSecondaryRequest class [470](#)
- asynchSpecification class [471](#)
- asynchStatusEnum enumeration set [1247](#)
- attribute
 - aggregate [1486](#)
 - autosummary [1488](#)
 - refVariable [1480](#)
 - refVariableValue [1480](#)
 - rollupaggregate [1486](#)
- auditLevelEnum enumeration set [1248](#)
- auditLevelEnumProp class [472](#)
- authentication [26](#)
- authentication method set [160](#)
- authoredAgentDefinition class [476](#)
- authoredPowerPlay8Report class [477](#)
- authoredReport class [479](#)
- authoredReport property [1457](#)
- authoring locale [1466](#)
- authoring reports [1457](#)
- automatic

- automatic (*continued*)
 - grouping and summarizing [1488](#)
- automaticAggregateOptimizationEnum enumeration set [1252](#)
- automating IBM Cognos actions
 - using BI Bus API [17](#), [18](#)
 - using Script Player [19](#)
 - using URL interface [18](#)
- autosummary attribute [1488](#)
- avoiding premature timeouts [119](#)
- axes
 - ancestor type [1568](#)
 - child type [1568](#)
 - descendant type [1568](#)
 - descendant-or-self type [1569](#)
 - parent type [1568](#)
 - specifying parent-child relationships during searches [1568](#)
 - using for content store searches [1565](#)
 - using in location steps [1565](#), [1568](#)
- axis element [1803](#)

B

- back(conversation, parameterValues, options) method [188](#)
- background(conversation) method [190](#)
- balancing
 - dispatcher loads [89](#)
- base64 decoding
 - troubleshooting XLS output formats [112](#)
- base64BinaryMIMEProp class [482](#)
- baseAgentDefinition class [483](#)
- baseAgentDefinitionActionEnum enumeration set [1252](#)
- baseAgentDefinitionActionEnumProp class [488](#)
- baseClass
 - source of derived properties [56](#)
- baseClass class [489](#)
- baseClassArrayProp class [499](#)
- baseClassProp class [502](#)
- baseDataIntegrationTask class [502](#)
- baseDataMovementTask class [507](#)
- baseHistoryDetail class [510](#)
- baseParameter class [512](#)
- baseParameterAssignment class [513](#)
- baseParameterAssignmentArrayProp class [514](#)
- baseParameterAssignmentProp class [514](#)
- basePowerPlay8Report class [514](#)
- basePowerPlay8ReportActionEnum enumeration set [1253](#)
- basePowerPlay8ReportActionEnumProp class [518](#)
- basePowerPlayClass class [518](#)
- baseProp class [520](#)
- baseReport class [523](#)
- baseReportActionEnum enumeration set [1253](#)
- baseReportActionEnumProp class [530](#)
- baseROLAPCubeConfiguration class [530](#)
- baseROLAPCubeConfigurationArrayProp class [532](#)
- baseROLAPCubeConfigurationProp class [533](#)
- baseROLAPDataSource class [533](#)
- baseRoutingRule class [535](#)
- baseRSSTask class [535](#)
- baseTextDirectionEnum enumeration set [1254](#)
- batchReportService class [132](#), [537](#)
- BI Bus [17](#)
- BI Bus API [17](#), [18](#)

- bibus schema [127](#)
- biBusHeader class [544](#)
- biBusHeaderExtension1 class [546](#)
- biDirectionalOptionEnum enumeration set [1254](#)
- bindingQualifierEnum enumeration set [1255](#)
- BlockConstraint element [1766](#)
- bookmark element [1542](#)
- booleanProp class [546](#)
- boundRangeParmValueItem class [548](#)
- browser cookie support [109](#)
- building
 - requests to run an object [66](#)

C

C#

- changing reports [42](#)
- connecting to IBM Cognos [26](#)
- deleting reports [34](#)
- executing reports [31](#)
- logging on to IBM Cognos [27](#)
- modifying reports [42](#)
- naming conventions [23](#)
- removing reports [34](#)
- running reports [31](#)
- scheduling reports [38](#)
- using the samples [23](#)

C# .NET

- samples [1424](#)
- services [129](#)

cacheOutput class [549](#)

CAF [1434](#)

CAF class [550](#)

calculated column [1489](#)

calculation element [1773](#)

CAM class [551](#)

CAMException class [554](#)

CAMID object-set function [1569](#)

CAMPassport class [555](#)

CAMProtect class [557](#)

CAMSettings class [558](#)

cancel(conversation) method [191](#)

cancelEvent(eventID) method [194](#)

cancelEvents(eventIDs) method [195](#)

capabilities [18](#)

capability class [558](#)

catalog class [559](#)

catalogFolder class [559](#)

cell element [1804](#)

cells element [1804](#)

changing

- object properties [57](#)
- relative processing capacity of dispatchers [89](#)
- severity of messages logged [88](#)

changing reports [40](#)

characterEncoding element [1542](#)

chart report

- filter measure values [1492](#)

childRef element [1529](#)

classEnum enumeration set [1255](#)

classEnumArrayProp class [560](#)

classEnumProp class [560](#)

classes

- Java [397](#)

classes (*continued*)

- types used by IBM® Cognos® [56](#)

clearCubeWorkloadLog(cubeNames, parameterValues, options) method [196](#)

clearNotifications(objectPath) method [198](#)

Client applications cannot connect to IBM® Cognos® BI [113](#)

code samples

- setup procedures [1419](#)

coding practices [97](#)

coding standards

- SDK-related recommended practices [97](#)

Cognos Connection, *See* IBM Cognos Connection

Cognos Extended Applications, *See* IBM Cognos Extended Applications

Cognos Portal Services (CPS), *See* IBM Cognos Portal Services (CPS)

Cognos toolkits [19](#)

Cognos.wsdl file

- location [19](#)
- using in IBM Cognos [17](#)

cognosdotnet_10_2.dll file [23](#)

cognosdotnetassembly_10_2.dll file [23](#)

collectParameterValues(objectPath, parameterValues, options) method [200](#)

collectParameterValues(objectPath, parameterValues, options) method – obsolete [202](#)

collectParameterValuesSpecification(specification, parameterValues, options) method [203](#)

collectParameterValuesSpecification(specification, parameterValues, options) method – obsolete [207](#)

color in a report [1478](#)

column

- calculated [1489](#)
- header [1472](#)
- order [1473](#)
- title in a list report [1467](#)

conditional block [1480](#)

conditional formatting [1479](#)

conditional processing [1480](#)

conditional report [1480](#)

conditionally

- hide [1480](#)
- show [1480](#)

conditionally display report objects [1480](#)

conditions

- testing [1480](#)

configuration

- locating objects in content store [1565](#)

configuration class [561](#)

configurationData class [625](#)

configurationDataEnum enumeration set [1266](#)

configurationFolder class [628](#)

configuring

- advanced routing [90](#)
- server groups [90](#)
- standby dispatchers [85](#)

conflictResolutionEnum enumeration set [1267](#)

connecting to IBM Cognos [25](#)

connection element [1529](#)

connections

- Client applications cannot connect [113](#)
- connecting to IBM® Cognos® BI [45](#)
- locating using dataSourceSignon searches [1570](#)

connections (*continued*)
 resolving ambiguities with multiple signons or connections [111](#)

Constraint element [1766](#), [1767](#)

Constraints element [1767](#)

contact class [685](#)

contains boolean function [1574](#)

content class [690](#)

content management [55](#)

Content Manager
 activating [87](#)
 running tasks [77](#)

content method set [160](#)

content store
 administration [55](#)
 data integrity and internal consistency [55](#)
 determining search paths of BI Bus API objects [1565](#)
 hierarchical relationships between objects [58](#)
 limiting number of stored objects [57](#)
 maintenance [64](#)
 optimistic concurrency control [55](#)
 property values and schema information [56](#)
 querying [60](#)
 relationships between referenced objects [59](#)
 searching [1565](#)

contentManagerCacheService class [692](#)

contentManagerQueryOption class [694](#)

contentManagerQueryOptionBoolean class [695](#)

contentManagerQueryOptionEnum enumeration set [1267](#)

contentManagerQueryOptionInt class [695](#)

contentManagerQueryOptionPropEnumArray class [695](#)

contentManagerQueryOptionRefPropArray class [696](#)

contentManagerQueryOptionString class [696](#)

contentManagerService class [134](#), [696](#)

contentStoreUtilizationConfiguration class [699](#)

contentTask class [700](#)

contentTaskOption class [703](#)

contentTaskOptionAuditLevelEnum class [703](#)

contentTaskOptionBoolean class [704](#)

contentTaskOptionClassEnumArray class [704](#)

contentTaskOptionCSUtilizationConfiguration class [705](#)

contentTaskOptionEnum enumeration set [1268](#)

contentTaskOptionRetentionRuleArray class [705](#)

contentTaskOptionSearchPathSingleObjectArray class [705](#)

context argument [1499](#)

contextFormatEnum enumeration set [1271](#)

contextOption class [706](#)

contextOptionEnum enumeration set [1271](#)

contextOptionFormat class [706](#)

contextOptionStringArray class [707](#)

contextOptionType class [707](#)

contextTypeEnum enumeration set [1272](#)

conversationContext class [707](#)

conversations
 major components [80](#)

convertDrillThroughContext(inputContext, parameterValues, options) method [208](#)

cookies
 browser support [109](#)

cookieVar class [709](#)

Coordinated Universal Time [109](#)

copy(objects, targetPath, options) method [209](#)

copyAccount(sourceAccountPath, targetAccountPath, options) method [212](#)

copyAccountOption class [709](#)

copyAccountOptionBoolean class [710](#)

copyAccountOptionEnum enumeration set [1272](#)

copying
 objects [61](#)
 user accounts [64](#)

copyOptions class [710](#)

copyRename(objects, targetPath, newNames, options) method [213](#)

create
 calculated column [1489](#)
 crosstab columns [1475](#)
 crosstab report [1474](#)
 crosstab rows [1475](#)
 filters in a report [1491](#)
 list report [1466](#)
 parameters in a report [1491](#)
 prompt in a report [1490](#), [1491](#)
 report specification [1461](#)

creating
 custom report functions [1497](#)
 deployment specifications [95](#)
 shortcuts [61](#)
 URLs [61](#)

credential class [711](#)

credentials
 parameter of logon method [47](#)

crosstab columns [1475](#)

crosstab report
 filter measure values [1492](#)

crosstab rows [1475](#)

crosstabColumns
 using in a report specification [1475](#)

crosstabCorner
 using in a report specification [1477](#)

crosstabFactCell
 using in a report specification [1477](#)

crosstabNodeMember
 using in a report specification [1475](#)

crosstabRows
 using in a report specification [1475](#)

crxSDK.h file [1497](#)

currency class [713](#)

currency format [1480](#)

currencyArrayProp class [714](#)

currencyProp class [714](#)

currentPage(conversation, parameterValues, options) method [215](#)

Custom Authentication Provider Solutions [45](#)

custom report functions
 creating [1497](#)

customizing
 applications based on SDK code samples [1419](#)

D

dashboard class [715](#)

dashboardOptionEnum enumeration set [1273](#)

data
 exporting [95](#)
 importing [95](#)

data limiting in a report [1490](#)

data sources
 testing connections [110](#)

- data types
 - C# .NET equivalents for XML [1427](#)
 - Java equivalents for XML [1427](#)
 - language-specific standards [1427](#)
 - XML-to-SQL mappings [1428](#)
- dataAdvisorService class [718](#)
- dataEnum enumeration set [1273](#)
- dataIntegrationService class [136](#), [719](#)
- dataIntegrationServiceSpecification class [721](#)
- dataIntegrationServiceSpecificationArrayProp class [722](#)
- dataIntegrationServiceSpecificationProp class [722](#)
- dataIntegrationTaskOption class [722](#)
- dataIntegrationTaskOptionBoolean class [723](#)
- dataIntegrationTaskOptionEnum enumeration set [1273](#)
- dataItem element [1558](#)
- dataMovementService class [137](#), [723](#)
- dataMovementServiceSpecification class [728](#)
- dataMovementServiceSpecificationProp class [728](#)
- dataMovementTask class [728](#)
- dataMovementTaskAlias class [730](#)
- dataSet class [731](#)
- dataSetFolder class [732](#)
- dataSource class [732](#)
- dataSource method set [161](#)
- dataSourceCapabilityEnum enumeration set [1275](#)
- dataSourceCommandBlock class [734](#)
- dataSourceCommandBlockProp class [734](#)
- dataSourceConnection class [735](#)
- dataSourceCredential class [737](#)
- dataSourceNameBinding class [738](#)
- dataSourceSignon class [739](#)
- dataSourceSignon object-set function [1570](#)
- date format [1480](#)
- date time format [1480](#)
- dateTimeProp class [740](#)
- daysEnum enumeration set [1276](#)
- decimalProp class [741](#)
- decimalSeparator element [1773](#)
- defaultLocale element [1773](#)
- defaultMeasure
 - using in a report specification [1477](#)
- defaultName property [114](#)
- defaultOutput object-set function [1571](#)
- defining
 - custom functions [1500](#)
 - slicer [1495](#)
- definingCells element [1805](#)
- delayEventsFor(eventIDs, for) method [217](#)
- delayEventsUntil(eventIDs, until) method [219](#)
- delete(objectPath, parameterValues, options) method [222](#)
- delete(objects, options) method [220](#)
- deleteAccount(objectPath, options) method [224](#)
- deleteAllNotifications() method [225](#)
- deleteHotList(objectPath) method [226](#)
- deleteNotification(objectPath) method [227](#)
- deleteOptions class [741](#)
- deleteTenants(tenantIDs) method [229](#)
- deleteTermAssociation(term, parameterValues, options) method [230](#)
- deleting
 - objects [63](#)
 - user accounts [63](#)
- deleting reports [33](#)
- deliver(conversation, parameterValues, options) method [231](#)
- delivering
 - scheduled reports [74](#)
- delivery method set [162](#)
- deliveryChannelEnum enumeration set [1276](#)
- deliveryOption class [742](#)
- deliveryOptionAddressSMTPArray class [743](#)
- deliveryOptionChannel class [743](#)
- deliveryOptionEnum enumeration set [1277](#)
- deliveryOptionMemoPart class [744](#)
- deliveryOptionSearchPathMultipleObjectArray class [744](#)
- deliveryOptionSearchPathSingleObject class [744](#)
- deliveryOptionString class [745](#)
- deliveryService class [138](#), [745](#)
- deliveryServiceSpecification class [747](#)
- dependentCells element [1805](#)
- deployment
 - viewing history [96](#)
- deployment applications
 - upgrading [125](#)
- deployment archives
 - listing [96](#)
 - moving [95](#)
 - viewing [96](#)
- deployment method set [162](#)
- deployment specifications
 - creating [95](#)
 - retrieving options [96](#)
- deploymentDetail class [748](#)
- deploymentImportRule class [750](#)
- deploymentObjectInformation class [751](#)
- deploymentOption class [752](#)
- deploymentOptionAnyType class [753](#)
- deploymentOptionArrayProp class [753](#)
- deploymentOptionAuditLevel class [753](#)
- deploymentOptionBoolean class [754](#)
- deploymentOptionClassEnumArray class [754](#)
- deploymentOptionEnum enumeration set [1278](#)
- deploymentOptionImportRuleArray class [754](#)
- deploymentOptionMultilingualString class [755](#)
- deploymentOptionObjectInformationArray class [755](#)
- deploymentOptionProp class [755](#)
- deploymentOptionResolution class [756](#)
- deploymentOptionSearchPathSingleObjectArray class [756](#)
- deploymentOptionString class [757](#)
- deploymentOptionStringArray class [757](#)
- deploymentReference class [757](#)
- deploymentReferenceArrayProp class [758](#)
- deploymentReferenceProp class [759](#)
- deploymentStatusEnum enumeration set [1284](#)
- deploymentStatusEnumProp class [759](#)
- descending order [1473](#)
- detail filters [1490](#)
- detail rows
 - showing [1488](#)
- detailSeverity element [1787](#)
- determineRouting(objectPaths) method [235](#)
- dimension element [1774](#)
- dimensionManagementService class [139](#), [759](#)
- dimensionManagementServiceSpecification class [764](#)
- directory class [764](#)
- disabling
 - objects [63](#)
- dispatcher class [140](#), [765](#)
- dispatcher method set [162](#)

- dispatcherID element [1788](#)
- dispatchers
 - balancing processing loads [89](#)
 - grouping in configuration folders [90](#)
 - monitoring status and request volumes [85](#), [86](#)
 - removing [87](#)
 - server groups [90](#)
 - testing [87](#)
- dispatcherTransportVar class [829](#)
- display element [1788](#)
- displayObject class [830](#)
- distributionList class [831](#)
- document class [833](#)
- documentContent class [835](#)
- documentVersion class [839](#)
- drill-through
 - dynamic [1535](#)
- drill-through specification
 - examples [1537](#)
- drill(conversation, parameterValues, options) method [236](#)
- drillOption class [840](#)
- drillOptionEnum enumeration set [1285](#)
- drillOptionParameterValues class [840](#)
- drillPath class [840](#)
- drillThrough method set [162](#)
- drillThroughAction class [844](#)
- drillThroughActionEnum enumeration set [1285](#)
- drillThroughOption class [844](#)
- drillThroughOptionAnyURI class [845](#)
- drillThroughOptionBoolean class [845](#)
- drillThroughOptionEnum enumeration set [1286](#)
- drillThroughOptionXMLEncodedXML class [845](#)
- drillThroughPath class [846](#)
- drillThroughRecipientEnum enumeration set [1287](#)
- drillThroughSpecification element [1543](#)
- durationProp class [846](#)
- dynamic drill-through [1535](#)

E

- edit specification
 - examples [1553](#)
 - URI [1556](#)
- editing report specifications [1457](#)
- editSpecification element [1558](#)
- effectiveLocale element [1774](#)
- element
 - action [1541](#)
 - actions [1557](#)
 - alternateItems [1541](#)
 - argument [1557](#)
 - arguments [1558](#)
 - axis [1803](#)
 - bookmark [1542](#)
 - cell [1804](#)
 - cells [1804](#)
 - characterEncoding [1542](#)
 - childRef [1529](#)
 - connection [1529](#)
 - dataItem [1558](#)
 - definingCells [1805](#)
 - dependentCells [1805](#)
 - detailSeverity [1787](#)
 - dispatcherID [1788](#)

- element (*continued*)
 - display [1788](#)
 - drillThroughSpecification [1543](#)
 - editSpecification [1558](#)
 - end [1788](#)
 - endTime [1788](#)
 - eventID [1789](#)
 - expression [1559](#)
 - extension [1543](#), [1559](#), [1805](#)
 - filter [1560](#)
 - filters [1789](#)
 - item [1544](#)
 - iwrTarget [1544](#)
 - lineageRequest [1529](#)
 - lineageResponse [1530](#)
 - metadataCell [1806](#)
 - metadataCells [1807](#)
 - modelSearchPaths [1545](#)
 - name [1530](#)
 - nullOrBlank [1560](#)
 - object [1530](#)
 - objectClass [1789](#)
 - objectQuery [1530](#)
 - objectQueryRef [1531](#)
 - objectRef [1531](#)
 - operator [1561](#)
 - order [1790](#)
 - owner [1790](#)
 - packageTarget [1545](#)
 - param [1531](#)
 - parameterAssignment [1545](#)
 - parameterAssignments [1546](#)
 - parameterName [1546](#)
 - parentEventID [1790](#)
 - parentRef [1532](#)
 - powerPlay7ReportTarget [1547](#)
 - powerPlay8ReportTarget [1547](#)
 - priority [1791](#)
 - promptPages [1491](#)
 - property [1532](#)
 - queryEventSpecification [1791](#)
 - queryResult [1532](#)
 - range [1561](#)
 - reportTarget [1547](#)
 - reportVariable [1480](#)
 - restartEventID [1791](#)
 - restartParentEventID [1792](#)
 - s [1807](#)
 - scheduleTrigger [1792](#)
 - scheduleType [1792](#)
 - scope [1548](#), [1793](#)
 - searchPath [1793](#)
 - searchPaths [1793](#)
 - selection [1808](#)
 - service [1533](#)
 - sf [1808](#)
 - simple [1562](#)
 - slicer [1490](#), [1495](#)
 - sort [1793](#)
 - sortItem [1794](#)
 - start [1794](#)
 - startTime [1795](#)
 - status [1795](#)
 - strings [1809](#)

- element (*continued*)
 - studios [1548](#)
 - subSort [1796](#)
 - summaryFilter [1492](#)
 - targetSearchPath [1549](#)
 - targetURI [1549](#)
 - text [1550](#)
 - transformation [1533](#)
 - type [1533](#)
 - uri [1550](#)
 - uriTarget [1551](#)
 - user [1796](#)
 - V5QuerySet [1534](#)
 - value [1563](#), [1809](#)
 - values [1810](#)
- element model group notation [1509](#), [1529](#), [1541](#), [1557](#), [1787](#), [1803](#)
- enabling Apache Axis logging [115](#)
- encodingEnum enumeration set [1288](#)
- end element [1559](#), [1788](#)
- ends-with boolean function [1574](#)
- endTime element [1788](#)
- enumeration sets
 - Java [1241](#)
- environment variables
 - passing to external applications [1453](#)
 - specifying in an XML file [1454](#)
 - types [1453](#)
- environmentVar class [847](#)
- errors
 - avoiding characters that are invalid [109](#)
 - avoiding invalid MIME header characters [110](#)
 - handling in requests [99](#)
 - interpreting long form SOAP faults [110](#)
 - maintaining data integrity and internal consistency of content store [55](#)
- event history
 - managing [70](#)
- event method set [163](#)
- eventID element [1789](#)
- eventManagementService class [141](#), [847](#)
- eventManagementServiceSpecification class [849](#)
- eventRecord class [849](#)
- eventTypeEnum enumeration set [1288](#)
- eventTypeEnumArrayProp class [851](#)
- eventTypeEnumProp class [851](#)
- EVService class [852](#)
- example
 - creating function set [1506](#)
- examples
 - changing reports [40](#)
 - connecting to IBM Cognos [25](#)
 - deleting reports [33](#)
 - executing reports [28](#)
 - logging on [26](#)
 - modifying reports [40](#)
 - passing environment variables [1454](#)
 - removing reports [33](#)
 - running reports [28](#)
 - running Script Player [44](#)
 - scheduling reports [36](#)
- executing reports [28](#)
- expandDistributionMembers object-set function [1571](#)
- expandMembers object-set function [1572](#)

- expandSecurityMembers object-set function [1572](#)
- exportDeployment class [853](#)
- exporting
 - data [95](#)
- expression
 - in a calculated column [1489](#)
 - specifying locales [1466](#)
- expression element [1559](#)
- extension element [1543](#), [1559](#), [1805](#)
- extra character in CSV report output [112](#)
- extract report specification [1457](#)

F

- faultDetail class [856](#)
- faultDetailArrayProp class [856](#)
- faultDetailMessage class [857](#)
- faultDetailProp class [857](#)
- favoritesFolder class [858](#)
- filelist
 - custom [1500](#)
- FileList.xml file [1497](#), [1500](#)
- files
 - crxSDK.h [1497](#)
 - custom filelist [1500](#)
 - FileList.xml [1497](#), [1500](#)
 - function definition [1500](#)
 - function description [1502](#)
- filter element [1560](#), [1775](#)
- filterExpression element
 - using in a report specification [1491](#)
- filtering
 - applying to a chart or crosstab report [1492](#)
 - data [1490](#)
 - measure values in a crosstab report [1492](#)
 - queries [1493](#)
 - using location step predicates [1569](#)
 - using node tests [1568](#)
 - with nil or not nil properties [1569](#)
- filters element [1789](#)
- findDrillThroughPaths(objectPath, parameterValues, options) method [238](#)
- fine-tune query performance [1482](#)
- firstPage(conversation, parameterValues, options) method [239](#)
- floatProp class [858](#)
- folder class [859](#)
- folder element [1775](#)
- folders
 - creating [61](#)
 - disabling [63](#)
 - grouping [60](#)
 - nesting [61](#)
- format data in a report [1480](#)
- formatEnum enumeration set [1289](#)
- formatting [1480](#)
- formatting reports [1478](#)
- formFieldVar class [863](#)
- forward(conversation, parameterValues, options) method [242](#)
- function argument [1498](#)
- function definition file [1500](#)
- function description file [1502](#)
- function element [1776](#)

- functions
 - boolean string searching [1573](#)
 - object-set searching [1569](#)
 - using for content store searches [1565](#)
 - using functions in search path [1569](#)
- Functions element [1768](#)
- functionsRoot element [1776](#)

G

- gatewayMapping class [864](#)
- gatewayMappingArrayProp class [864](#)
- gatewayMappingProp class [865](#)
- gateways [19](#)
- gcPolicyEnum enumeration set [1289](#)
- genericOption class [865](#)
- genericOptionAnyURI class [866](#)
- genericOptionBoolean class [866](#)
- genericOptionStringArray class [867](#)
- genericOptionXMLEncodedXML class [867](#)
- get report from content store [1457](#)
- get(objectPath, parameterValues, options) method [244](#)
- getActiveContentManager() method [245](#)
- getConfiguration(properties) method [246](#)
- getContext(conversation, parameterValues, options) method [247](#)
- getCubeMessages(cubeNames, parameterValues, options) method [249](#)
- getCubeMetrics(cubeNames, parameterValues, options) method [251](#)
- getCubeState(cubeNames, parameterValues, options) method [252](#)
- getDeploymentOptions(archive, options) method [254](#)
- getFormatSamples(name) method [256](#)
- getIdentity() method [257](#)
- getObjectContext(objectPath, parameterValues, options) method [258](#)
- getOutput(conversation, parameterValues, options) [79](#)
- getOutput(conversation, parameterValues, options) method [259](#)
- getParameters(objectPath, parameterValues, options) method [262](#)
- getParametersSpecification(specification, parameterValues, options) method [265](#)
- getPromptValues(conversation, parameterValues, options) method [268](#)
- getPromptValues(conversation, parameterValues, options) method – obsolete [271](#)
- Globalreportstyles.css [1478](#)
- glossary [1925](#)
- governor class [867](#)
- governorArrayProp class [868](#)
- governorBoolean class [868](#)
- governorEnum enumeration set [1290](#)
- governorInt class [869](#)
- governorProp class [869](#)
- graphic class [869](#)
- graphicsService class [871](#)
- group class [874](#)
- group footer [1472](#)
- group report [1469](#)
- group span [1471](#)
- grouped
 - column [1469](#)

- grouped (*continued*)
 - repeater [1469](#)
- grouped column [1471](#)
- grouping
 - automatic [1488](#)
 - dispatchers in configuration folders [90](#)
- groups
 - creating [48](#)
 - identifying using membership searches [1573](#)
 - removing [49](#)
 - removing members [49](#)
- guid class [877](#)
- guidArrayProp class [877](#)
- guidelines for URL commands [108](#)
- guidProp class [878](#)

H

- hasProperty boolean function [1574](#)
- hdrSession class [878](#)
- hierarchicalParmValueItem class [879](#)
- hierarchies [35](#)
- hierarchy element [1777](#)
- highlight rows in a report [1479](#)
- history class [880](#)
- historyDetail class [883](#)
- historyDetailAgentService class [884](#)
- historyDetailDataMovementService class [884](#)
- historyDetailDeploymentSummary class [885](#)
- historyDetailMigrationService class [886](#)
- historyDetailRelatedHistory class [887](#)
- historyDetailRelatedReports class [888](#)
- historyDetailReportService class [888](#)
- historyDetailRequestArguments class [889](#)
- holdEvent(eventID) method [272](#)
- holdEvents(eventIDs) method [273](#)
- hot swapping PowerCubes [93](#)
- http
 - //developer.cognos.com/schemas/report [1459](#)
- humanTask class [889](#)
- humanTaskService class [892](#)

I

- IBM Cognos
 - starting components using URLs [1435](#)
 - See also* IBM Cognos
- IBM Cognos Application Firewall [1434](#)
- IBM Cognos Software Development Kit
 - advanced settings [1577](#)
- IBM Cognos toolkits [19](#)
- IBM® Cognos® Portal Services (CPS)
 - extended applications not working [114](#)
- ID-based object access
 - using storeID function for object-set searching [1573](#)
- Identity class [894](#)
- idVizService class [142](#), [895](#)
- importDeployment class [896](#)
- importing
 - data [95](#)
- incorrect metadata [113](#)

- incrementallyLoadCubes(cubeNames, parameterValues, options) method [274](#)
- indexDataService class [898](#)
- indexOption class [900](#)
- indexOptionBoolean class [900](#)
- indexOptionEnum enumeration set [1291](#)
- indexOptionInt class [901](#)
- indexOptionSearchPathMultipleObjectArray class [901](#)
- indexSearchService class [142](#), [902](#)
- indexSearchServiceSpecification class [904](#)
- indexTerm class [904](#)
- indexTerm method set [164](#)
- indexTermOption class [905](#)
- indexTermOptionBoolean class [905](#)
- indexTermOptionEnum enumeration set [1293](#)
- indexTermOptionIndexTermArray class [906](#)
- indexUpdate method set [164](#)
- indexUpdateService class [143](#), [906](#)
- indexUpdateServiceSpecification class [908](#)
- indexUpdateServiceSpecificationProp class [909](#)
- indexUpdateTask class [909](#)
- installedComponent class [912](#)
- installedComponentEnum enumeration set [1293](#)
- installedComponentEnumProp class [912](#)
- installing
 - custom functions [1502](#)
- interactive report [1465](#)
- interactiveReport class [913](#)
- interfaces [159](#)
- interpreting long form SOAP faults [110](#)
- interval format [1480](#)
- intProp class [914](#)
- invalid characters
 - MIME headers [110](#)
 - programming scenarios [109](#)
- invalid MIME header characters [110](#)
- isolationLevelEnum enumeration set [1298](#)
- item element [1544](#)
- iwrTarget element [1544](#)

J

Java

- classes [397](#)
- connecting to IBM Cognos [25](#)
- deleting reports [34](#)
- enumeration sets [1241](#)
- file locations [20](#)
- IBM Cognos toolkit [20](#)
- logging on to IBM Cognos [27](#)
- methods [171](#)
- modifying reports [41](#)
- naming conventions [20](#)
- running reports [28](#)
- samples [1419](#)
- scheduling reports [36](#)
- services [129](#)
- setup procedures [21](#), [22](#)
- setup procedures for samples [1423](#)
- using the samples [21](#)
- jobDefinition class [925](#)
- jobOption class – deprecated [929](#)
- jobOptionBoolean class – deprecated [929](#)
- jobOptionEnum enumeration set – obsolete [1298](#)

- jobs
 - running [76](#)
- jobService class [145](#), [930](#)
- jobStepDefinition class [931](#)

L

- language-specific naming conventions [108](#)
- languageProp class [933](#)
- languages
 - multiple [1481](#)
- last boolean function [1574](#)
- lastPage(conversation, parameterValues, options) method [276](#)
- launch IBM Cognos PowerPlay using a URL [1442](#)
- launchable class [934](#)
- layout element
 - using in a report specification [1461](#)
- layouts element
 - using in a report specification [1461](#)
- level element [1777](#)
- library
 - custom report function [1497](#)
- limiting data in a report [1490](#)
- lineage(conversation, parameterValues, options) method [279](#)
- lineageRequest element [1529](#)
- lineageResponse element [1530](#)
- linearUnitEnum enumeration set [1299](#)
- list column [1467](#)
- list element
 - using in a report specification [1466](#)
- list footer [1472](#)
- list header [1472](#)
- list page footer [1472](#)
- list page header [1472](#)
- list report [1465](#)
- list report sample [1466](#)
- list reports
 - creating [1466](#)
- listArchives() method [280](#)
- listColumnRowSpan element
 - using in a report specification [1469](#)
- listGroups element
 - using in a report specification [1469](#)
- listing
 - deployment archives [96](#)
- listSeparator element [1778](#)
- listTenants(options) method [282](#)
- listTenantsOptions class [936](#)
- load balancing
 - IBM® Cognos® dispatchers [89](#)
- loadBalancingModeEnum enumeration set [1299](#)
- loadBalancingModeEnumProp class [936](#)
- locale class [937](#)
- locale element [1778](#)
- localeArrayProp class [938](#)
- localeMapEntry class [938](#)
- localeMapEntryArrayProp class [934](#), [938](#)
- localeMapEntryProp class [939](#)
- localeProp class [939](#)
- locales
 - supporting multiple locales in a report specification [1481](#)

- locales element [1779](#)
- localizing
 - English API text exposed to users [110](#)
- location steps
 - abbreviated syntax [1565](#), [1567](#), [1568](#)
 - composition [1565](#), [1567](#)
 - containing predicate filters [1569](#)
 - search path notations [1565](#)
 - search paths [1567](#)
 - unabbreviated syntax [1565](#), [1567](#), [1568](#)
- logging
 - default location of error messages [88](#)
 - errors categorized by severity [88](#)
 - properties [88](#)
- logging on
 - and credential parameter [47](#)
 - anonymous logon support [109](#)
- logoff() method [282](#)
- logon(credentials, roles) method [283](#)
- logService class [939](#)

M

- maintaining
 - content store [64](#)
- managing
 - event history [70](#)
 - user accounts [63](#)
- managing service headers [97](#)
- map report [1465](#)
- mapContentLocale(locale, normalize) method [286](#)
- mapLayerTypeEnum enumeration set [1300](#)
- mapOption class [941](#)
- mapOptionEnum enumeration set [1300](#)
- mapping
 - types [1501](#)
- mapProductLocale(locale, normalize) method [287](#)
- MAQuery elements [1765](#)
- MAQuery.xsd file [1765](#)
- MAResponse elements [1772](#)
- MAResponse.xsd file [1765](#), [1772](#)
- measure element [1779](#)
- member element [1780](#)
- membership object-set function [1573](#)
- memo class [941](#)
- memoPart class [943](#)
- memoPartAgentObject class [944](#)
- memoPartAgentObjectLinks class [945](#)
- memoPartComposite class [946](#)
- memoPartMIMEAttachment class [946](#)
- memoPartObject class [947](#)
- memoPartObjectLinks class [948](#)
- memoPartString class [948](#)
- message class [948](#)
- metadata
 - unable to view after adding query [113](#)
- metadata API
 - request element [1765](#)
 - ResponseRoot element [1772](#)
- Metadata element [1768](#)
- metadata method set [165](#)
- metadataCell element [1806](#)
- metadataCells element [1807](#)
- metadataModelExpression class [949](#)
- metadataModelItemName class [949](#)
- metadataModelItemNameArrayProp class [949](#)
- metadataModelItemNameProp class [950](#)
- metadataService class [146](#), [950](#)
- metadataServiceLineageSpecification class [954](#)
- metadataServiceModelInformationSpecification class [955](#)
- metadataServiceSpecification class [955](#)
- methods
 - Java [171](#)
 - secondary requests [1417](#)
- Methods [171](#)
- metrics
 - running tasks [77](#)
- metricsDataSourceETLTask class [956](#)
- metricsExportTask class [956](#)
- metricsFileImportTask class [956](#)
- metricsMaintenanceTask class [957](#)
- metricsManagerService class [957](#)
- MHT
 - end of line missing in output [110](#)
- migrationCubeMapping class [959](#)
- migrationMapping class [959](#)
- migrationService class [147](#), [960](#)
- migrationServiceSpecification class [962](#)
- migrationServiceSpecificationProp class [963](#)
- migrationTask class [963](#)
- migrationTaskOption class [966](#)
- migrationTaskOptionEnum enumeration set [1301](#)
- migrationTaskOptionMappingArray class [967](#)
- migrationTaskOptionResolution class [967](#)
- migrationTaskOptionSearchPathSingleObject class [967](#)
- migrationTaskOptionSearchPathSingleObjectArray class [968](#)
- mobileOption class [968](#)
- mobileOptionEnum enumeration set [1302](#)
- mobileOptionSearchPathMultipleObjectArray class [969](#)
- mobileService class [148](#), [969](#)
- model class [971](#)
- models
 - creating or modifying with Script Player [19](#)
 - optimizing for reporting purposes [89](#)
- modelSearchPath element [1781](#)
- modelSearchPaths element [1545](#)
- modelView class [974](#)
- modify report specification [1457](#)
- modifying
 - object properties [57](#)
- modifying reports [40](#)
- modifying summaries [1486](#)
- monitor method set [165](#)
- monitoring
 - dispatcher status [85](#), [86](#)
 - error log messages [88](#)
 - request volumes [85](#), [86](#)
- monitorOption class [974](#)
- monitorOptionBoolean class [974](#)
- monitorOptionEnum enumeration set [1302](#)
- monitorService class [149](#), [975](#)
- monitorServiceSpecification class [976](#)
- monthsEnum enumeration set [1303](#)
- move(objects, targetPath, options) method [288](#)
- moveOptions class [976](#)
- moveRename(objects, targetPath, newNames, options) method [290](#)
- moving

- moving (*continued*)
 - deployment archives [95](#)
 - reports [61](#)
- mruFolder class [977](#)
- multilingual names
 - values of locale-related properties [56](#)
- multilingualString class [979](#)
- multilingualStringProp class [979](#)
- multilingualToken class [980](#)
- multilingualTokenProp class [980](#)
- multiple queries
 - filtering [1493](#)
- multiple servers and updateMetadata method [111](#)
- multithreaded applications
 - troubleshooting [115](#), [120](#)
- multithreaded applications
 - managing biBusHeader objects [120](#)

N

- name element [1530](#)
- namespace
 - report [1459](#)
- namespace class [114](#), [981](#)
- namespaceCapabilityEnum enumeration set [1304](#)
- namespaceFolder class [983](#)
- nameValue class [984](#)
- naming conventions
 - .NET standards [108](#)
 - C# standards [108](#)
 - Java standards [108](#)
- navigationPath class [985](#)
- navigationPathArrayProp class [986](#)
- navigationPathProp class [986](#)
- nnameProp class [986](#)
- nextPage(conversation, parameterValues, options) method [292](#)
- nil class [987](#)
- nmtokenArrayProp class [987](#)
- nmtokenProp class [987](#)
- node tests
 - search paths [1568](#)
 - using for content store searches [1565](#)
 - using in location steps [1565](#), [1568](#)
- nonNegativeIntegerProp class [988](#)
- normalizeLocale(locale) method [295](#)
- notifications
 - running tasks [77](#)
- null element [1560](#)
- nullOrBlank element [1560](#)
- number format [1480](#)

O

- object element [1530](#)
- objectClass element [1789](#)
- objectLink class [989](#)
- objectQuery element [1530](#)
- objectQueryRef element [1531](#)
- objectRef element [1531](#)
- objects
 - building requests to run [66](#)
 - content store [1607](#)

- objects (*continued*)
 - copying [61](#)
 - deleting [63](#)
 - executing with run (objectPath, parameterValues, options) [65](#)
 - setting security [50](#)
 - specifying run options [66](#)
 - specifying run parameters [66](#)
- Objects
 - controlling visibility [61](#)
- open report from clipboard [1460](#)
- operator element [1561](#)
- optimize query performance [1482](#)
- optimizing
 - dispatcher loads [89](#)
 - model designs to support reporting [89](#)
 - server performance [89](#)
 - upgrading report specifications [113](#)
- optimizing reports [1482](#)
- option class [990](#)
- optionArrayProp class [993](#)
- optionProp class [994](#)
- options
 - specifying on a run request [66](#)
- order element [1790](#)
- orderEnum enumeration set [1305](#)
- organizing
 - dispatchers in configuration folder groups [90](#)
 - objects [63](#)
 - objects in the Web portal [60](#)
- output class [995](#)
- outputEncapsulationEnum enumeration set [1305](#)
- outputFormatEnum enumeration set [1306](#)
- overall footer [1472](#)
- overall header [1472](#)
- overriding
 - model summaries [1486](#)
- ownedBy object-set function [1573](#)
- owner element [1790](#)

P

- package class [999](#)
- package deployment option
 - upgrading [125](#)
- package element [1782](#)
- packageActionEnum enumeration set [1308](#)
- packageActionEnumProp class [1005](#)
- packageConfiguration class [1005](#)
- packages
 - disabling [63](#)
- packageTarget element [1545](#)
- page class [1007](#)
- pageDefinition class [1008](#)
- pagelet class [1009](#)
- pageletFolder class [1010](#)
- pageletInstance class [1011](#)
- pageOrientationEnum enumeration set [1308](#)
- paging method set [165](#)
- pagingOption class [1012](#)
- pagingOptionEnum enumeration set [1308](#)
- pagingOptionInt class [1012](#)
- param element [1531](#)
- parameter class [1013](#)

- parameter method set [166](#)
- parameterAssignment element [1545](#)
- parameterAssignmentDataItem class [1014](#)
- parameterAssignments element [1546](#)
- parameterCapabilityEnum enumeration set [1309](#)
- parameterDataSource class [1014](#)
- parameterDataTypeEnum enumeration set [1310](#)
- parameterName element [1546](#)
- parameters
 - format for values of type xsdDate [109](#)
 - format for values of type xsdDateTime [109](#)
 - retrieving from content store [57](#)
 - specifying on a run request [66](#)
- parameterValue class [1014](#)
- parameterValueArrayProp class [1020](#)
- parameterValueProp class [1021](#)
- parentEventID element [1790](#)
- parentRef element [1532](#)
- parmValueItem class [1021](#)
- pass parameters to a report [1491](#)
- passing
 - environment variables [1454](#)
- passports [26](#)
- Path element [1769](#)
- pauseCubes(cubeNames, parameterValues, options) method [296](#)
- pdfCharacterEncodingEnum enumeration set [1311](#)
- pdfCharacterEncodingEnumProp class [1022](#)
- pdfCompressionTypeEnum enumeration set [1312](#)
- pdfCompressionTypeEnumProp class [1023](#)
- pdfFontEmbeddingEnum enumeration set [1313](#)
- pdfFontEmbeddingEnumProp class [1023](#)
- pdfOption class [1024](#)
- pdfOptionBoolean class [1025](#)
- pdfOptionEnum enumeration set [1314](#)
- pdfOptionPrintQuality class [1025](#)
- pdfOptionXMLEncodedXML class [1025](#)
- pdfPrintQualityEnum enumeration set [1315](#)
- percentage format [1480](#)
- performance
 - tuning IBM Cognos servers [89](#)
- performance of report [1482](#)
- periodical class [1026](#)
- permission boolean function [1574](#)
- permission class [1027](#)
- permissions
 - setting security [50](#)
- personalDataSourceSignon object-set function [1570](#)
- personalization class [1027](#)
- personalizationFolder class [1028](#)
- ping(dispatcherPath) method [298](#)
- pingReply class [1029](#)
- planningAdministrationConsoleService class [150](#), [1030](#)
- planningAdministrationConsoleServiceSpecification class [1032](#)
- planningAdministrationConsoleServiceSpecificationProp class [1032](#)
- planningApplication class [1033](#)
- planningDataService class [1034](#)
- planningMacroTask class [1037](#)
- planningRuntimeService class [150](#), [1040](#)
- planningRuntimeServiceSpecification class [1041](#)
- planningStateEnum enumeration set [1316](#)
- planningStateEnumProp class [1042](#)
- planningTask class [1042](#)
- planningTaskService class [151](#), [1045](#)
- planningTaskServiceSpecification class [1047](#)
- planningTaskServiceSpecificationProp class [1047](#)
- policies
 - security [50](#)
- policy class [1048](#)
- policyArrayProp class [1048](#)
- policyProp class [1049](#)
- portal class [1049](#)
- portalDisplayModeEnum enumeration set [1316](#)
- portalListSeparatorEnum enumeration set [1316](#)
- portalOption class [1050](#)
- portalOptionBoolean class [1050](#)
- portalOptionDisplayMode class [1050](#)
- portalOptionEnum enumeration set [1317](#)
- portalOptionInt class [1051](#)
- portalOptionListSeparator class [1051](#)
- portalOptionSearchPathSingleObject class [1051](#)
- portalOptionSearchPathSingleObjectArray class [1052](#)
- portalOptionString class [1052](#)
- portalOptionXMLEncodedXML class [1052](#)
- portalPackage class [1053](#)
- portalSkin class [1054](#)
- portalSkinFolder class [1056](#)
- portlet class [1056](#)
- portletFolder class [1058](#)
- portletInstance class [1059](#)
- portletProducer class [1059](#)
- positiveIntegerProp class [1062](#)
- PowerCubes
 - hot swapping [93](#)
- powerPlay7ReportTarget element [1547](#)
- powerPlay8DataEnum enumeration set [1318](#)
- powerPlay8Option class [1062](#)
- powerPlay8OptionAnyURI class [1063](#)
- powerPlay8OptionBoolean class [1064](#)
- powerPlay8OptionData class [1064](#)
- powerPlay8OptionEnum enumeration set [1318](#)
- powerPlay8OptionLanguageArray class [1064](#)
- powerPlay8OptionOutputFormat class [1065](#)
- powerPlay8OptionSaveAs class [1065](#)
- powerPlay8OptionSearchPathSingleObject class [1066](#)
- powerPlay8OptionString class [1066](#)
- powerPlay8OutputFormatEnum enumeration set [1320](#)
- powerPlay8OutputFormatEnumArrayProp class [1067](#)
- powerPlay8OutputFormatEnumProp class [1067](#)
- powerPlay8Report class [1068](#)
- powerPlay8ReportTarget element [1547](#)
- powerPlay8ReportView class [1068](#)
- powerPlay8SaveAsEnum enumeration set [1321](#)
- powerPlayCube class [1069](#)
- powerPlayDataBlock class [1069](#)
- powerPlayDataBlockArrayProp class [1070](#)
- powerPlayDataBlockEnum enumeration set [1321](#)
- powerPlayDataBlockProp class [1070](#)
- powerPlayDataEnum enumeration set [1321](#)
- powerPlayOption class [1070](#)
- powerPlayOptionArrayProp class [1071](#)
- powerPlayOptionBoolean class [1071](#)
- powerPlayOptionData class [1072](#)
- powerPlayOptionEnum enumeration set [1322](#)
- powerPlayOptionOutputFormat class [1072](#)
- powerPlayOptionProp class [1072](#)

- powerPlayOutputFormatEnum enumeration set [1322](#)
- powerPlayReport class [1073](#)
- powerPlayReportActionEnum enumeration set [1323](#)
- powerPlayReportActionEnumProp class [1073](#)
- powerPlayService class [151](#), [1074](#)
- powerPlayServiceReportSpecification class [1078](#)
- powerPlayServiceSpecification class [1078](#)
- powerPlayServiceSpecificationProp class [1079](#)
- predicates
 - content store searches [1565](#)
 - location steps [1565](#), [1567](#), [1569](#)
 - searching [1569](#)
- presentationService class [1079](#)
- previousPage(conversation, parameterValues, options) method [299](#)
- primary requests [78](#)
- printer class [1080](#)
- priority element [1791](#)
- processing
 - conditional [1480](#)
- processing constraints
 - asynchronous conversations [83](#)
- productCapability class [1083](#)
- programs
 - upgrading [121](#)
- prompt [1490](#)
- promptButton element
 - using in a report specification [1491](#)
- promptCacheModeEnum enumeration set [1323](#)
- promptInfo class [1081](#)
- promptOption class [1082](#)
- promptPages
 - using in a report specification [1491](#)
- promptPaging method set [166](#)
- promptTypeEnum enumeration set [1324](#)
- propEnum enumeration set [1324](#)
- properties
 - changing default timeout settings [111](#), [119](#)
 - logging [88](#)
 - modifying [57](#)
 - multilingual values [56](#)
 - serverGroup [90](#)
 - values stored with schema information [56](#)
 - viewing general property values [56](#)
- Properties element [1769](#), [1770](#)
- property element [1532](#)
- Property element [1770](#), [1771](#)
- provider class [1083](#)

Q

- queries
 - filtering [1493](#)
- queries element
 - using in a report specification [1461](#)
- query class [1084](#)
- query element
 - using in a report specification [1461](#)
- query selects incorrect metadata [113](#)
- query(objectPath, parameterValues, options) method [305](#)
- query(searchPath, properties, sortBy, options) method [302](#)
- queryCountOptions class [1085](#)
- queryCountResult class [1085](#)

- queryDrillPath(objectPath, parameterValues, options) method [308](#)
- queryEventSpecification element [1791](#)
- querying
 - content store [60](#)
- queryItem element [1782](#)
- queryItemFolder element [1784](#)
- queryMetadata(request) method [310](#)
- queryModeEnum enumeration set [1356](#)
- queryMultiple(requests) method [311](#)
- queryMultipleCache(requests, options) method [312](#)
- queryMultipleOptions class [1085](#)
- queryMultipleResult class [1086](#)
- queryNotification(objectPath) method [314](#)
- queryOptionEnum enumeration set [1356](#)
- queryOptions class [1086](#)
- queryProcessingEnum enumeration set [1357](#)
- queryReply class [1088](#)
- queryRequest class [1088](#)
- queryResult element [1532](#)
- queryService class [153](#), [1090](#)
- queryServiceTask class [1097](#)
- querySubject element [1784](#)
- queryTaskOptionEnum enumeration set [1357](#)
- queryTenantMembership(tenantIDs) method [315](#)

R

- range element [1561](#)
- recommended practices [118](#)
- refProp class [1099](#)
- refreshCubeDataCache(cubeNames, parameterValues, options) method [316](#)
- refreshCubeMemberCache(cubeNames, parameterValues, options) method [318](#)
- refreshCubeSecurity(cubeNames, parameterValues, options) method [320](#)
- refVariable
 - attribute [1480](#)
- refVariableValue
 - attribute [1480](#)
- registering
 - custom report functions [1500](#)
- relationalMetadataService class [153](#), [1100](#)
- relationalMetadataServiceSpecification class [1104](#)
- relationships
 - hierarchy of objects in content store [58](#)
 - IBM® Cognos® classes [57](#)
 - parent and child objects [57](#)
 - reference type [57](#), [59](#)
 - specifying using a location step axis [1568](#)
- release(conversation) method [322](#)
- releaseEvent(eventID) method [325](#)
- releaseEvents(eventIDs) method [326](#)
- removing
 - dispatchers [87](#)
- removing reports [33](#)
- render(conversation, parameterValues, options) method [328](#)
- repeater report [1465](#)
- repeater table [1465](#)
- report
 - aggregates [1486](#)
 - chart [1465](#)
 - color [1478](#)

- report (*continued*)
 - conditional formatting [1479](#), [1480](#)
 - crosstab [1474](#)
 - currency [1480](#)
 - data formats [1480](#)
 - detail filters [1490](#)
 - formatting [1478](#), [1480](#)
 - header [1472](#)
 - interactive [1465](#)
 - list [1465](#)
 - map [1465](#)
 - repeater [1465](#)
 - repeater table [1465](#)
 - styles [1478](#)
 - summaries [1486](#)
 - summary filters [1490](#)
 - totals [1486](#)
 - types [1465](#)
 - variables [1480](#)
- report class [1104](#)
- report element
 - using in a report specification [1466](#)
- report functions
 - prototype [1497](#)
 - return value [1499](#)
- report method set [167](#)
- report specification
 - aggregates [1486](#)
 - structure [1461](#)
 - validation [1459](#)
- report specification schema changes [125](#)
- report specifications
 - upgrading [121](#)
 - upgrading to optimize performance [113](#)
 - using [1457](#)
- reportCache class [1105](#)
- reportDataService class [1107](#)
- reportDataServiceAgentDefinition class [1108](#)
- reportDataServiceSpecification class [1109](#)
- reportDataServiceSpecificationProp class [1110](#)
- reporting examples [24](#)
- reportNet.wsdl file, [19](#)
- reports
 - changing [40](#)
 - connecting [25](#)
 - deleting [33](#)
 - executing [28](#)
 - extra character in CSV output [112](#)
 - grouping [60](#)
 - modifying [40](#)
 - moving [61](#)
 - removing [33](#)
 - running [28](#)
 - scheduled distributions [74](#)
 - scheduling [36](#)
 - sorting [60](#)
- reportSaveAsEnum enumeration set [1362](#)
- reportService class [154](#), [1110](#)
- reportServiceAnalysisSpecification class [1117](#)
- reportServiceDrillThroughSpecification class [1117](#)
- reportServiceDrillThroughSpecificationProp class [1118](#)
- reportServiceInteractiveReportSpecification class [1118](#)
- reportServiceMetadataSpecification class [1119](#)
- reportServiceQueryDrillPathOption class [1119](#)
- reportServiceQueryDrillPathOptionBoolean class [1120](#)
- reportServiceQueryDrillPathOptionEnum enumeration set [1362](#)
- reportServiceQueryOption class [1120](#)
- reportServiceQueryOptionBoolean class [1121](#)
- reportServiceQueryOptionEnum enumeration set [1362](#)
- reportServiceQueryOptionSpecificationFormat class [1121](#)
- reportServiceQuerySpecification class [1121](#)
- reportServiceReportSpecification class [1122](#)
- reportServiceSpecification class [1122](#)
- reportStudioOption class [1122](#)
- reportStudioOptionEnum enumeration set [1363](#)
- reportStudioOptionSearchPathSingleObject class [1123](#)
- reportTarget element [1547](#)
- reportTemplate class [1123](#)
- reportVariable
 - element [1480](#)
- reportVersion class [1124](#)
- reportView class [1127](#)
- repositoryRule class [1128](#)
- repositoryRuleArrayProp class [1129](#)
- repositoryRuleProp class [1130](#)
- repositoryService class [156](#), [1130](#)
- request element [1772](#)
- requests
 - building requests to run an object [66](#)
 - executing with wait (conversation, parameterValues, options) [78](#)
 - primary and secondary requests using asynchronous methods [78](#)
- requirements
 - administrator privileges [85](#)
- resolving ambiguous data source connections [111](#)
- resource class [1132](#)
- ResponseRoot element [1785](#)
- Repository Service
 - API reference [1811](#)
- restartCubes(cubeNames, parameterValues, options)
 - method [330](#)
- restartEventID element [1791](#)
- restartParentEventID element [1792](#)
- result argument [1498](#)
- retention problems [111](#)
- retention rules
 - avoiding database overload [111](#)
 - avoiding unwanted overwrites [111](#)
 - limiting number of stored objects [57](#)
- retentionRule class [1133](#)
- retentionRuleArrayProp class [1135](#)
- retentionRuleProp class [1136](#)
- retrieve report specification [1457](#)
- retrieveCredential(namespace) method [331](#)
- retrying
 - tasks [72](#)
- return values
 - report function [1499](#)
- rolapCubeAdministration method set [168](#)
- rolapCubeConfiguration class [1136](#)
- rolapCubeMessages class [1139](#)
- rolapCubeMetrics class [1140](#)
- rolapDataSource class [1140](#)
- rolapDataSourceStateEnum enumeration set [1363](#)
- rolapMessage class [1141](#)
- rolapMetric class [1141](#)

- rolapMetricDateTime class [1142](#)
- rolapMetricNumber class [1142](#)
- rolapMetricString class [1144](#)
- rolapOptionEnum enumeration set [1364](#)
- rolapVirtualCubeConfiguration class [1144](#)
- rolapVirtualDataSource class [1144](#)
- role class [1145](#)
- roles
 - identifying using membership searches [1573](#)
 - locating using dataSourceSignon searches [1570](#)
 - removing [49](#)
 - removing members [49](#)
- rollupaggregate attribute [1486](#)
- root class
 - locating in content store [1565](#)
- routing
 - adding hints [90](#)
 - advanced [92](#)
- routingHintObject class [1149](#)
- routingInfo class [1150](#)
- routingRuleObject class [1150](#)
- routingTableEntry class [1151](#)
- routingTableEntryArrayProp class [1151](#)
- routingTableEntryProp class [1152](#)
- rssOption class [1152](#)
- rssOptionAnyURI class [1152](#)
- rssOptionEnum enumeration set [1364](#)
- rssOptionSearchPathSingleObject class [1153](#)
- rssOptionString class [1153](#)
- RSVP.RENDER.VALIDATEURL [1434](#)
- run (objectPath, parameterValues, options)
 - using to execute objects [65](#)
- run report specification [1457](#)
- run(objectPath, parameterValues, options) method [333](#)
- runAt(startTime, objectPath, parameterValues, options) method [340](#)
- runConditionEnum enumeration set [1365](#)
- runConditionEnumProp class [1153](#)
- running
 - agents [76](#)
 - Content Manager tasks [77](#)
 - jobs [76](#)
 - metrics-related tasks [77](#)
 - notification tasks [77](#)
 - service-based tasks [72](#)
 - tasks [65](#)
- running reports [28](#)
- runningStateEnum enumeration set [1366](#)
- runningStateEnumProp class [1154](#)
- runOption class [1155](#)
- runOptionAnyURI class [1156](#)
- runOptionArrayProp class [1156](#)
- runOptionBoolean class [1157](#)
- runOptionData class [1157](#)
- runOptionDateTime class [1157](#)
- runOptionEnum enumeration set [1367](#)
- runOptionInt class [1158](#)
- runOptionLanguageArray class [1158](#)
- runOptionMultilingualString class [1158](#)
- runOptionNameValueArray class [1159](#)
- runOptionOutputEncapsulation class [1159](#)
- runOptionPromptCacheMode class [1159](#)
- runOptionProp class [1160](#)
- runOptionSaveAs class [1160](#)

- runOptionString class [1161](#)
- runOptionStringArray class [1161](#)
- runSpecification(specification, parameterValues, options) method [342](#)
- runStatusEnum enumeration set [1373](#)
- runTimeState class [1161](#)

S

- s element [1807](#)
- saCAMService class [156](#), [1163](#)
- sample
 - crosstab report [1474](#)
- samples
 - active server pages (ASP) [1419](#)
 - adding a filter [1491](#)
 - adding a prompt [1491](#)
 - C# [23](#)
 - C# .NET [1424](#)
 - C# .NET sample setup procedures [1426](#)
 - creating a list report [1466](#)
 - filtering measure values in a crosstab report [1492](#)
 - Java [1419](#)
 - Java sample setup procedures [1422](#)
 - Java setup [1423](#)
 - Java setup steps [21](#), [22](#)
 - location of ASP files [1419](#)
 - location of C# .NET files [1426](#)
 - location of Java files [1420](#)
 - overriding default Content Manager end point [1423](#)
 - searching content store [1565](#)
 - setting access permissions [1423](#)
 - setup procedures [1419](#)
 - using samples with network printers [112](#)
- SAP BW
 - interpreting errors [112](#)
- save report specification [1457](#)
- schedule class [1164](#)
- scheduleDailyPeriodEnum enumeration set [1375](#)
- scheduledEvent class [1171](#)
- scheduleEndTypeEnum enumeration set [1375](#)
- scheduleEvent(eventID) method [353](#)
- scheduleEvents(eventIDs) method [354](#)
- scheduleTrigger element [1792](#)
- scheduleType element [1792](#)
- scheduleTypeEnum enumeration set [1376](#)
- scheduling
 - report distributions [74](#)
 - tasks [70](#)
- scheduling reports [36](#)
- schema changes [125](#)
- schemaInfo class [1172](#)
- schemas
 - information in content store [56](#)
- scope element [1548](#), [1793](#)
- Script Player
 - running [44](#)
- sdk application
 - report specification [1457](#)
- search paths
 - location steps [1567](#)
 - using functions [1569](#)
- searching
 - content store [1565](#)

- searching (*continued*)
 - determining paths of objects in content store [1565](#)
- searchPath element [1793](#)
- searchPathMultipleObject class [1173](#)
- searchPaths element [1793](#)
- searchPathSingleObject class [1174](#)
- searchPathSingleObjectArrayProp class [1176](#)
- searchPathSingleObjectProp class [1176](#)
- secondary requests
 - asynchronous methods [78](#)
- securedFeature class [1176](#)
- securedFunction class [1177](#)
- security
 - applying in IBM® Cognos® [45](#)
 - credential parameter of logon method [47](#)
 - managing for use with URL commands [109](#)
 - removing groups and roles [49](#)
 - removing members from groups and roles [49](#)
 - setting policies and permissions [50](#)
 - setting up for use with URL commands [109](#)
 - using CAMID function for object-set searching [1569](#)
 - using permission function for boolean searching [1574](#)
- selection context
 - examples [1799](#)
- selection element [1808](#)
- selectionContextFormatEnum enumeration set [1379](#)
- selectRoles(roles) method [355](#)
- selectValue element
 - using in a report specification [1491](#)
- sequencingEnum enumeration set [1379](#)
- server groups
 - configuring [90](#)
- servers
 - tuning performance [89](#)
- service element [1533](#)
- service headers
 - managing [97](#)
- service-based tasks
 - running [72](#)
- services
 - activating Content Manager [87](#)
 - C# .NET [129](#)
 - Java [129](#)
 - list [130](#)
 - starting [86](#)
 - stopping [86](#)
 - testing [87](#)
- session class [1177](#)
- sessions
 - locating in content store [1565](#)
- set permission
 - samples [1423](#)
- setCookieVar class [1179](#)
- setting
 - log messages [88](#)
 - security policies and permissions [50](#)
- setup procedures
 - Java [21](#), [22](#)
 - Java samples [1423](#)
- severityEnum enumeration set [1380](#)
- severityEnumProp class [1180](#)
- sf element [1808](#)
- shortcut class [1181](#)
- shortcutAgentRSTask class [1182](#)
- shortcutRSTask class [1182](#)
- shortcuts
 - creating [61](#)
 - troubleshooting broken links [61](#)
- signons
 - locating using dataSourceSignon searches [1570](#)
 - resolving ambiguities with multiple connections or signons [111](#)
- simple element [1562](#)
- simple object access protocol
 - used in IBM Cognos [17](#)
- simpleParmValueItem class [1183](#)
- slicer
 - defining [1495](#)
 - element [1490](#), [1495](#)
- slow report [1482](#)
- smtpContentDispositionEnum enumeration set [1381](#)
- smtpContentTypeEnum enumeration set [1381](#)
- SOAP, [17](#)
- softwareEdition class [1184](#)
- softwareEditionArrayProp class [1185](#)
- softwareEditionProp class [1185](#)
- sort class [1186](#)
- sort data [1473](#)
- sort element [1793](#)
- sorting
 - objects [63](#)
- sortItem [1473](#)
- sortItem element [1794](#)
- sortList [1473](#)
- special characters
 - encoding for use in URL commands [109](#)
- specification class [1186](#)
- specificationFormatEnum enumeration set [1381](#)
- specificationOption class [1186](#)
- specificationOptionEnum enumeration set [1382](#)
- specificationOptionString class [1187](#)
- specificationOptionXMLEncodedXML class [1187](#)
- SQL
 - XML data type mappings [1428](#)
- SQL class [1188](#)
- standaloneCAM method set [168](#)
- standards
 - SDK-related coding practices [97](#)
- standby dispatchers
 - configuring [85](#)
- start element [1562](#), [1794](#)
- Start_at element [1772](#)
- startCubes(cubeNames, parameterValues, options) method [356](#)
- starting
 - IBM Cognos components using URLs [1435](#)
 - IBM® Cognos® services [86](#)
- starts-with boolean function [1575](#)
- startService(servicePath) method [358](#)
- startTime element [1795](#)
- status element [1795](#)
- stopCubes(cubeNames, parameterValues, options) method [359](#)
- stopping
 - IBM® Cognos® services [86](#)
- stopService(servicePath, immediately) method [361](#)
- storedProcedureTask class [1188](#)
- storeID object-set function [1573](#)

- string format [1480](#)
- stringArrayProp class [1191](#)
- stringMapEntry class [1192](#)
- stringMapEntryArrayProp class [1192](#)
- stringMapEntryProp class [1193](#)
- stringProp class [1193](#)
- strings element [1809](#)
- studios element [1548](#)
- styles [1478](#)
- stylesheet [1478](#)
- styling [1478](#)
- subquery in a report [1493](#)
- subscriptionFolder class [1198](#)
- subscriptionOption class [1199](#)
- subscriptionOptionEnum enumeration set [1383](#)
- subscriptionOptionSearchPathSingleObject class [1199](#)
- subSort element [1796](#)
- summaries
 - automatic [1488](#)
 - modifying in a report specification [1486](#)
- summaries in reports [1486](#)
- summary filters [1490](#)
- summary footer [1472](#)
- summary header [1472](#)
- summaryFilter element [1492](#)
- system method set [169](#)
- systemMetricEnum enumeration set [1383](#)
- SystemMetricThresholds
 - troubleshooting [118](#)
- systemMetricThresholds class [1200](#)
- systemMetricThresholdsPropertyEnum enumeration set [1386](#)
- systemOptionEnum enumeration set [1387](#)
- SystemService class [157](#), [1202](#)

T

- targetSearchPath element [1549](#)
- targetURI element [1549](#)
- tasks
 - performing using URLs [1431](#)
 - retrying [72](#)
 - running [65](#)
 - running Content Manager [77](#)
 - running metrics-related tasks [77](#)
 - running notifications [77](#)
 - running service-based tasks [72](#)
 - scheduling [70](#)
- temporaryObjectLocationEnum enumeration set [1388](#)
- tenancy class [1203](#)
- tenant class [1204](#)
- tenantInfo class [1205](#)
- tenants class [1205](#)
- terminateSessions(search) method [363](#)
- testDataSourceConnection(connectionString, credentials) method [364](#)
- testDataSourceConnection(connectionString, credentials) method – obsolete [366](#)
- testDataSourceConnectionWithInfo(connectionString, credentials) method [367](#)
- testing
 - dispatchers and services [87](#)
- testing conditions [1480](#)
- testing connections to data sources [110](#)

- text element [1550](#)
- threading
 - deserializer not found error in multithreaded Java client [115](#)
 - multithreaded applications [120](#)
- time format [1480](#)
- timeouts
 - changing default settings [111](#), [119](#)
- timeProp class [1206](#)
- timeZone class [1206](#)
- tokenArrayProp class [1207](#)
- tokenProp class [1207](#)
- toolkits
 - C# and .NET naming conventions [108](#)
 - IBM Cognos [19](#)
 - Java naming conventions [108](#)
 - Java toolkit setup procedures [20](#)
 - language-specific data types [1427](#)
 - Visual Studio .NET Framework setup procedures [23](#)
 - XML data types for C# .NET [1427](#)
 - XML data types for Java [1427](#)
- tracking class [1208](#)
- transformation element [1533](#)
- transientStateFolder class [1209](#)
- translating
 - English API text exposed to users [110](#)
- translation of exposed API text [110](#)
- trigger(triggerName) method [368](#)
- troubleshooting
 - access manager will authenticate an external namespace [115](#)
 - avoiding premature timeouts [111](#), [119](#)
 - axis attachment support [116](#)
 - base64 decoding XLS formats [112](#)
 - biBusHeader objects in multithreaded applications [120](#)
 - broken shortcut links [61](#)
 - caption of an unrecoverable error is not displayed in the browser [117](#)
 - Client applications cannot connect to IBM® Cognos® [113](#)
 - deprecated SDK methods may timeout [117](#)
 - deserializer not found error in multithreaded Java client [115](#)
 - enabling Apache Axis logging [115](#)
 - extra character in CSV report output [112](#)
 - IBM® Cognos® Portal Services not working [114](#)
 - improving performance by upgrading the report specification [113](#)
 - improving server performance [89](#)
 - incorrect metadata [113](#)
 - interpreting long form SOAP faults [110](#)
 - invalid characters [109](#)
 - invalid MIME header characters [110](#)
 - keyTransformation only partially works [116](#)
 - language-specific naming conventions [108](#)
 - logged information for multiple report runs are collapsed [117](#)
 - MHT output format [110](#)
 - multiple servers and updateMetadata method [111](#)
 - multithreaded applications [115](#), [120](#)
 - performance degradation due to excessive error logging [88](#)
 - performance problems in Java applications due to serialization of null objects [117](#)
 - query selects incorrect metadata [113](#)

- troubleshooting (*continued*)
 - resolving ambiguous data source connections [111](#)
 - resources [97](#)
 - retention problems [111](#)
 - SystemMetricThresholds [118](#)
 - testing connections to data sources [110](#)
 - translation of exposed API text [110](#)
 - triggerName containing the question-mark symbol (?) may be misinterpreted [116](#)
 - triggername issues [116](#), [117](#)
 - ui.drillThroughTargetParameterValues [118](#)
 - unable to update defaultName property of a namespace [114](#)
 - unable to view metadata after adding query [113](#)
 - unable to view running reports [116](#)
 - updateMetadata method requires SDK on server [111](#)
 - updateMetadata method with multiple servers [111](#)
 - using SDK samples with network printers [112](#)
 - using the selectRoles method has no effect [117](#)
 - viewing error log messages [88](#)
 - XLWA output format [110](#)
- tuning
 - server performance [89](#)
- type element [1533](#)
- types of reports [1465](#)

U

- ui.drillThroughTargetParameterValues
 - troubleshooting [118](#)
- uiClass
 - source of derived interface properties [56](#)
- uiClass class [1210](#)
- uiComponentEnum enumeration set [1388](#)
- uiComponentEnumArrayProp class [1216](#)
- uiComponentEnumProp class [1216](#)
- uiProfile class [1217](#)
- uiProfileFolder class [1218](#)
- unable to update [114](#)
- unable to update defaultName property [114](#)
- unable to update defaultName property of a namespace [114](#)
- unboundedEndRangeParmValueItem class [1219](#)
- unboundedStartRangeParmValueItem class [1219](#)
- update(object, options) method [373](#)
- update(objects, options) method [370](#)
- updateActionEnum enumeration set [1389](#)
- updateDrillPath(object, options) method [376](#)
- updateEvents(events) method [377](#)
- updateMetadata method requires SDK on server [111](#)
- updateMetadata method with multiple servers [111](#)
- updateMetadata(request) method [379](#)
- updateOptions class [1220](#)
- updating
 - object properties [57](#)
- upgrading
 - axis 1.4 [122](#)
 - cam_passport cookie [123](#)
 - changing the .NET project references [121](#)
 - deployment applications [125](#)
 - Multi-Version Coexistence [124](#)
 - package deployment option [125](#)
 - report specification schema [125](#)
 - report specifications [121](#)
 - report specifications after a software upgrade [113](#)

- upgrading (*continued*)
 - SDK applications [121](#)
 - to IBM Cognos Analytics Version 10.1.0 [122](#)
 - to IBM Cognos Analytics Version 10.2.0 [121](#)
 - to IBM Cognos Analytics Version 8.3 [125](#)
 - to IBM Cognos Analytics Version 8.4 [124](#)
 - XLS [124](#)
- URI [1556](#)
- uri element [1550](#)
- uriTarget element [1551](#)
- uriValue class [1221](#)
- uriValueArrayProp class [1222](#)
- uriValueProp class [1222](#)
- URL class [1222](#)
- URL commands
 - guidelines [108](#)
- URL validation [1434](#)
- urlRSSTask class [1224](#)
- URLs
 - creating [61](#)
 - recommended practices [1431](#)
 - starting IBM Cognos components [1435](#)
 - starting IBM Cognos PowerPlay [1442](#)
- URLS
 - accessing services using [1431](#)
 - url_xml [1431](#)
- usageEnum enumeration set [1390](#)
- user accounts
 - copying [64](#)
 - deleting [63](#)
 - managing [63](#)
- user element [1796](#)
- userCapabilityCache class [1225](#)
- userCapabilityEnum enumeration set [1391](#)
- userCapabilityEnumArrayProp class [1225](#)
- userCapabilityEnumProp class [1226](#)
- userCapabilityPermission class [1226](#)
- userCapabilityPolicy class [1227](#)
- userCapabilityPolicyArrayProp class [1228](#)
- userCapabilityPolicyProp class [1228](#)
- userInterfaceProfile class [1228](#)
- userPreferenceVar class [1229](#)
- using SDK samples with network printers [112](#)

V

- V5QuerySet element [1534](#)
- validate method set [169](#)
- validate report specification [1459](#)
- validate(objectPath, parameterValues, options) method [381](#)
- validate(objectPath, parameterValues, options) method – obsolete [380](#)
- validateContentLocale(locale) method [384](#)
- validateHintEnum enumeration set [1414](#)
- validateOption class [1230](#)
- validateOptionBoolean class [1231](#)
- validateOptionEnum enumeration set [1415](#)
- validateOptionHint class [1231](#)
- validateOptionValidateSeverity class [1231](#)
- validateProductLocale(locale) method [385](#)
- validateSeverityEnum enumeration set [1415](#)
- validateSpecification(specification, parameterValues, options) method [387](#)

- validateSpecification(specification, parameterValues, options) method – obsolete [386](#)
- validation
 - URL [1434](#)
- value element [1563](#), [1809](#)
- values element [1810](#)
- variables
 - using in a report specification [1480](#)
- versions
 - optimistic concurrency control [55](#)
- view report specification XML in Reporting [1460](#)
- viewing
 - deployment history [96](#)
 - dispatcher status [85](#), [86](#)
 - error log messages [88](#)
 - request volumes [85](#), [86](#)
- Visibility
 - of objects [61](#)
- visualization class [1232](#)

W

- wait (conversation, parameterValues, options)
 - using with requests [78](#)
- wait(conversation, parameterValues, options) [79](#)
- wait(conversation, parameterValues, options) method [391](#)
- Web services [19](#)
- Web services description language
 - used in IBM Cognos [17](#)
- webServiceTask class [1233](#)
- weeksEnum enumeration set [1416](#)
- WSDL file, [19](#)
- WSDL, [17](#)

X

- XLWA
 - end of line missing in output [110](#)
- XML in Reporting [1460](#)
- XMLBase [1556](#)
- xmlEncodedXML class [1236](#)
- xmlEncodedXMLArrayProp class [1238](#)
- xmlEncodedXMLMIMEProp class [1238](#)
- xmlEncodedXMLProp class [1239](#)
- xsdDateTime and xsdDate parameter values [109](#)

