

Administering IBM Data Virtualization
Manager
1.1



Note

Before using this information and the product it supports, read the information in [“Product legal notices” on page 455.](#)

This edition applies to Version 1 Release 1 of IBM Data Virtualization Manager for z/OS and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-07-07

© **Copyright International Business Machines Corporation 2017, 2022.**

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

© **Rocket Software, Inc. 2017, 2022.**

Contents

Figures.....	ix
Tables.....	xi
About this information.....	xv
How to send your comments to IBM.....	xvii
If you have a technical problem.....	xvii
Chapter 1. Overview.....	1
Chapter 2. Working with started task parameters.....	3
Invoking the task application.....	3
Viewing Details about a Parameter.....	3
Modifying a started task parameter.....	5
Adding a started task parameter to IN00 file.....	6
Chapter 3. General administrative functions.....	9
Managing users.....	10
Terminating a user connection.....	14
Handling user connections.....	14
Configuring virtual connections.....	15
Creating multiple administrative users.....	16
Displaying and modifying ISPF/AVZ session parameters.....	18
Displaying formatted control blocks.....	18
Displaying server statistics.....	21
Managing tokens.....	22
Displaying product tasks.....	23
Displaying product copies.....	25
Displaying 64-bit buffer information.....	27
Chapter 4. Rules and events.....	31
Rules and rule sets.....	31
Automatic limits.....	31
Variables for rules.....	32
Events.....	33
Authorization (ATH) events.....	33
Command (CMD) events.....	57
Exception (EXC) events.....	59
Global variable (GLV) events.....	73
Remote procedure call (RPC) events.....	74
SQL events.....	75
Time-of-day (TOD) events.....	76
Virtual table (VTB) events.....	77
API functions for rules.....	83
AVZVALUE API function.....	83
AVZINFO API function.....	89
AVZECURE API function.....	91
AVZSUBMIT API function.....	94

DISPLAY command.....	97
----------------------	----

Chapter 5. Configuring access to data sources 101

Configuring access to Adabas.....	101
Configuring the server started task JCL.....	101
Modifying the Data Virtualization Manager configuration member.....	102
Configuring security access to Adabas data.....	103
Configuring access to RDBMS.....	103
Configuring access to IBM Db2 for z/OS.....	104
Configuring access to distributed databases.....	119
Controlling display and access for native Db2 subsystems.....	132
Copying Alternate IDs for DRDA Data Sources Between DVM Servers.....	133
Configuring access to IMS databases.....	134
Configuring the server started task JCL.....	134
Installing virtual table and virtual view maps for IMS catalog access.....	134
Configuring ODBA Support.....	135
Modifying the DBCTL configuration member.....	136
Modifying the IMS Direct configuration member.....	137
Configuring access to IBM MQ.....	141
Configuring the server started task JCL.....	142
Modifying the Data Virtualization Manager configuration member for IBM MQ.....	142
Configuring virtual table rules for IBM MQ.....	142
Configuring access to native VSAM.....	144
Verifying access to native VSAM.....	144
Modifying the data and index buffers for VSAM files.....	144
VSAM record-level sharing.....	145
VSAM File Control and ERLY Servers.....	145
Configuring access to sequential files.....	152
Reading ahead tracks for sequential file access.....	152
Configuring access to IBM CICS.....	153
Modifying the Data Virtualization Manager configuration member.....	153
Configuring CICS.....	156
Configuring security.....	157
Configuring access to zFS files.....	157
Configuring access to data for IT Operational Analytics.....	157
Configuring access to System Management Facility (SMF) files.....	158
Configuring access to SYSLOG files.....	161
Configuring access to OPERLOG files.....	162
Configuring access to CA IDMS.....	162
Configuring the server started task JCL.....	163
Modifying the Data Virtualization Manager configuration member for CA IDMS.....	163
Verifying access to CA IDMS data.....	164
Configuring access to ADDI.....	165
Installing virtual tables and virtual target maps for ADDI access.....	166
Modifying the configuration member for ADDI access.....	166
Configuring virtual table rules for ADDI.....	170
Configuring authentication for ADDI.....	170
Configuring access to RAA.....	171
Installing virtual tables and virtual target maps for RAA access.....	172
Modifying the configuration member for RAA access.....	173
Configuring virtual table rules for RAA.....	176
Configuring authentication for RAA.....	176

Chapter 6. SQL solution..... 179

Chapter 7. NoSQL solution..... 181

Modifying the Data Virtualization Manager configuration member.....	181
---	-----

MongoDB interface examples.....	182
NoSQL for MongoDB interface example.....	182
JavaScript and Node.js.....	183
C#.....	184
Creating reports from a MongoDB data source.....	185
Supported MongoDB query language.....	186
Chapter 8. z/OS Connect solution.....	191
Configuring the z/OS Connect solution.....	191
Configuring the Data Virtualization Manager Service Provider on zcEE.....	191
Creating an IBM z/OS Connect Enterprise Edition server instance.....	193
Configuring the server started task JCL.....	194
Modifying the Data Virtualization Manager configuration member.....	194
Starting the Data Virtualization Manager server and z/OS Connect server.....	195
z/OS Connect.....	196
Data Virtualization Manager server configuration.....	196
Data Virtualization Manager configuration member parameters.....	200
Configuring Service Provider on zcEE.....	203
Chapter 9. Distributed transactions.....	205
Recoverable Resource Management Services (RRMS) and the two-phase commit.....	205
Enterprise transactions for Db2.....	205
Configuring support for distributed Db2 transactions.....	206
Configuring support for distributed Db2 transactions with the Microsoft Transaction Server	207
Enterprise transactions for CICS/TS and IMS.....	208
Configuring support for CICS and IMS distributed transactions.....	209
Chapter 10. Running multiple servers.....	211
Configuring additional Data Virtualization Manager servers.....	211
Using multiple Data Virtualization Manager servers as peers.....	212
Configuring multiple Data Virtualization Manager servers as peers.....	212
Creating virtual maps for tables on a peer server.....	216
z Systems Data Compression (zEDC).....	217
Enabling zEDC.....	217
Chapter 11. IDF for Db2 mainframe applications.....	219
Known Restrictions and Limitations.....	219
Configuring Data Virtualization Manager server for IDF.....	220
Populating DB2 communications database for IDF.....	221
Populating the SYSIBM.LOCATIONS table.....	221
Populating the SYSIBM.IPNAMES table.....	222
Populating the SYSIBM.USERNAMES table.....	223
Using IDF for Mainframe Applications to implement DB2 RESTful services.....	224
DB2 RESTful services provisioning.....	224
Creating a DB2 RESTful service using IDF.....	224
Accessing IDF from DB2I (SPUFI).....	226
Using IDF in DB2 application programs.....	227
Chapter 12. Security.....	229
Security Optimization Management (SOM).....	229
Enabling basic SOM support.....	229
Enabling advanced SOM support.....	231
Using PassTickets.....	232
Logon and logoff processing.....	232
ACEE retention and deletion.....	233
Configuring mixed case password for RACF password.....	233
Secure Sockets Layer (SSL).....	234

Enabling SSL support.....	234
Configuring AT-TLS manually.....	238
Network access control.....	238
Enabling network access control.....	239
Enterprise auditing.....	240
Using generic and extended IDs.....	241
Host side support.....	242
Creating a z/OS security environment.....	242
Enabling enterprise auditing.....	243
Protected resources.....	243
Defining resources to RACF.....	247
Defining resources to CA Top Secret.....	248
Defining resources to ACF2.....	248
Optional security jobs.....	248
ISPF load modules.....	249
RACF PassTickets.....	250
Defining security for RPCs.....	250
Information access with the TRACEDATA resource.....	251
Resource security for test versions of Data Virtualization Manager server.....	252
Configuring Adabas security.....	252
Configuring CICS security.....	253
MRO login security.....	253
MRO bind time security.....	254
Link security.....	254
User security.....	254
Virtual table SAF security.....	255
Chapter 13. Performance.....	257
System resources.....	257
CPU time limits.....	257
Enabling the program execution duration time limit mechanism.....	260
Modifying the client auxiliary storage cut-off parameter.....	261
Detecting sessions failures.....	262
Workload Manager (WLM).....	263
WLM enclaves.....	263
Configuring Workload Manager (WLM).....	264
Using the WLM Administration Tool.....	270
Workload Manager definitions.....	270
WLM classification rules.....	273
Using WLM classifications.....	273
Activating the WLM service policy.....	274
Verifying WLM classification.....	274
WLM Health Reporting.....	275
Server load balancing.....	276
Enabling load balancing for a group director.....	277
Enabling load balancing for CICS/TS.....	278
Enabling load balancing for Services.....	278
CICS failover.....	279
Enabling CICS failover.....	280
Block fetch.....	280
Enabling block fetch.....	281
Configuring DB2 for z/OS Continuous Block Fetch.....	281
MapReduce.....	282
Virtual Parallel Data.....	282
Innovation Access Method (IAM).....	284
Metadata repository.....	285
Work Area or Record Buffer.....	286

Work Area Expansion in a Post-Read Exit.....	286
64-bit buffer.....	286
Chapter 14. Logging and tracing server information.....	289
Server Trace.....	289
Displaying and navigating log entries	290
Locating entries in the server log.....	295
Filtering log entries	295
Labeling and locating specific log entries.....	298
Finding character strings in the server log.....	298
Capturing the entries from the server trace	299
Archiving the Server Trace.....	299
Setting a server trace profile value for a subsystem.....	302
System Management Facility logging.....	302
Enabling SMF logging	302
Record Subtype 01: Client System.....	306
Record Subtype 02: Internal Summary.....	309
Record Subtype 03: SEF Rule Disablement.....	310
Record Subtype 04: Global Variable.....	312
Record Subtype 05: Services (Non-SOAP requests).....	313
Record Subtype 06: Per Transaction SMF Records.....	316
Record Subtype 09: Storage Interval Summary.....	318
Record Subtype 10: APPC/MVS Interval Summary.....	319
Record Subtype 11: APPC/MVS Conversation Summary SMF.....	320
Record Subtype 13: DB2 SQL Errors.....	322
Record Subtype 14: Client Response Time.....	323
Record Subtype 16: Record Broker Statistics.....	325
Record Subtype 17: ADABAS Command by DBID Records.....	327
Record Subtype 18: Services Records.....	329
Record Subtype 19: Interval Usage Recording Options.....	331
Record Subtype 20: Transaction SMF Record.....	333
DB2 logging.....	335
Enabling DB2 logging.....	336
Record: Sessions.....	337
Record: Interval	340
Record: SQL Source.....	343
Record: Storage.....	345
Record: APPC/MVS.....	346
Records: Error Log.....	347
Record: Services.....	349
Record: Streams.....	355
Chapter 15. Sysplex Function.....	359
Defining and enabling Sysplex Function.....	359
Accessing Sysplex Function.....	360
AVZ Select.....	360
AVZ Execute.....	360
Chapter 16. Monitoring.....	363
Monitoring Data Virtualization Manager client response time.....	364
Monitoring streams with Server Trace.....	365
Instrumentation Server.....	366
Reducing the amount of tracing.....	366
Installing the Instrumentation Server.....	367
Using the Instrumentation Server in a sysplex.....	368
Monitoring and managing RRS transactions.....	369
RRS Manager display.....	369

Enabling two-phase commit transaction processing.....	369
Viewing active two-phase commit transactions.....	370
Viewing indoubt two-phase commit transactions.....	372
Displaying information about failed two-phase commit transactions.....	374
Invoking the RRS Units of recovery information	375
Chapter 17. Working with error messages.....	379
Viewing details about an error message.....	379
Sorting an error message.....	381
Changing the severity of an error message.....	381
Chapter 18. Migrating maps.....	383
Chapter 19. Virtualizing mainframe data.....	385
Multiple schema support.....	386
Prerequisites for adding virtual tables to a schema.....	387
Creating schema maps on the server.....	387
Moving the existing maps or virtual tables to a schema.....	388
Viewing schema maps from the ISPF panel.....	389
Using batch JCL jobs.....	389
Using the ISPF application.....	390
IBM Data Virtualization Manager for z/OS Interface for ACI.....	390
Adabas.....	412
IBM Data Virtualization Manager for z/OS Interface for CICS/TS.....	413
DB2.....	423
IBM Data Virtualization Manager for z/OS Interface for IMS DB: support for DBCTL.....	425
VSAM and sequential files.....	437
Using the Data Mapping Facility.....	443
Accessibility features.....	453
Product legal notices.....	455
Index.....	459

Figures

- 1. Sysplex Function..... 359
- 2. IMS database representation..... 428
- 3. Using the Data Mapping Facility with the IBM Data Virtualization Manager for z/OS Interface for
IMB DB/SQL.....428
- 4. Data access path 1..... 429
- 5. Data access path 2..... 430
- 6. Data access path 3..... 431

Tables

1. Primary Option Menu - Interface Facilities.....	9
2. Primary Option Menu - Server Administration.....	10
3. Sort Names.....	20
4.	22
5. Sort Names For Tokens.....	22
6. Column Names.....	24
7. Line Commands.....	24
8. Column Names.....	25
9. Line Commands.....	27
10. 64-bit buffer information.....	28
11. Action Codes and return values.....	84
12. SQL parameters in group PRODSQL.....	118
13. SQL parameters in group PRODTRACE.....	118
14. CA IDMS EMPDEMO database maps.....	164
15. CA IDMS network schema IDMSNTWK maps.....	165
16. SQL access by data source type.....	179
17. Protected resources.....	244
18. Data Virtualization Manager access requirements.....	245
19. Optional security jobs.....	248
20. IBM Data Virtualization Manager for z/OS load modules.....	249
21. WLM Element Types.....	270
22. Server Trace panel columns.....	291
23. Profile filtering criteria.....	295

24. Profile filtering events.....	296
25. SMF Parameters	302
26. Subtype 01 Record Information.....	307
27. Subtype 02 Record Information.....	309
28. Subtype 03 Record Information.....	311
29. Subtype 04 Record Information.....	312
30. Subtype 05 Record Information.....	314
31. Subtype 06 Record Information.....	316
32. Subtype 09 Record Information.....	318
33. Subtype 10 Record Information.....	319
34. Subtype 11 Record Information.....	320
35. Subtype 13 Record Information.....	322
36. Subtype 14 Record Information.....	324
37. Subtype 16: Record Broker Statistics.....	325
38. Subtype 17 Record Information.....	327
39. Subtype 18 Record Information.....	329
40. Subtype 19 Record Information.....	331
41. Subtype 20: Transaction SMF Record.....	333
42. Sessions Record for DB2.....	338
43. Interval Record for DB2.....	341
44. SQL Source Record for DB2.....	343
45. Storage Record for DB2.....	345
46. APPC/MVS record for DB2.....	346
47. Error Log records for DB2.....	348
48. Services Record for DB2.....	350

49. Services Record for DB2: Interval recording using no specific criteria.....	351
50. Services Record for DB2: Interval Recording at the Web Services Level.....	351
51. Services for DB2: End of Session record	353
52. Streams Record for DB2.....	355
53. Error Message Description.....	380
54. Security permissions required to use the migration utility.....	383
55. Server ACI Facility.....	390
56. Conversions of COBOL data types to ODBC data types.....	405
57. ACI timeout values.....	408
58. FORMAT column types and the SQL equivalent.....	412
59. Server Adabas Data Mapping Facility.....	413
60. Server CICS Control Facility.....	413
61. Server IMS Data Mapping Facility.....	425
62. Access to file type by interface.....	437
63. Server VSAM/Sequential Data Mapping Facility.....	437
64. Server Data Mapping Facility.....	443

About this information

This information supports IBM Data Virtualization Manager for z/OS (5698-DVM) and contains information about the Data Virtualization Manager server, which is a component that is provided with IBM Data Virtualization Manager for z/OS.

Purpose of this information

This document presents the information you need to perform administrative tasks while using the Data Virtualization Manager server.

Who should read this information

This document is intended for system administrators, system programmers, and database administrators.

How to send your comments to IBM

We appreciate your input on this documentation. Please provide us with any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

Important: If your comment regards a technical problem, see instead [“If you have a technical problem”](#) on page xvii.

Send an email to comments@us.ibm.com.

Include the following information:

- Your name and address
- Your email address
- Your phone or fax number
- The publication title and order number:
 - IBM Data Virtualization Manager for z/OS Administration Guide
 - SC27-9303-00
- The topic and page number or URL of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM®, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are listed for sending comments. Instead, take one or more of the following actions:

- Visit the [IBM Support Portal \(support.ibm.com\)](https://support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Chapter 1. Overview

As an administrator, you need to perform several administrative tasks to enable you to use IBM Data Virtualization Manager for z/OS.

See the following sections for more information on administering the Data Virtualization Manager:

- [Chapter 2, “Working with started task parameters,” on page 3](#)
- [Chapter 3, “General administrative functions,” on page 9](#)
- [Chapter 4, “Rules and events,” on page 31](#)
- [Chapter 5, “Configuring access to data sources,” on page 101](#)
- [User Connections](#)
- [Chapter 9, “Distributed transactions,” on page 205](#)
- [Chapter 10, “Running multiple servers,” on page 211](#)
- [Chapter 11, “IDF for Db2 mainframe applications,” on page 219](#)
- [Chapter 12, “Security,” on page 229](#)
- [Chapter 13, “Performance,” on page 257](#)
- [Chapter 14, “Logging and tracing server information,” on page 289](#)
- [Chapter 16, “Monitoring,” on page 363](#)
- [Chapter 17, “Working with error messages,” on page 379](#)

Chapter 2. Working with started task parameters

This chapter describes how to manage the started task parameters of IBM Data Virtualization Manager for z/OS.

Invoking the task application

Complete the following steps to invoke the AVZ task application.

Procedure

1. Start the ISPF application and go to the IBM Data Virtualization Manager for z/OS's Primary Option menu.
2. Select the option **C** for **AVZ Admin**.
3. Press **ENTER**.

Results

The following panel is displayed:

```
Option ==>>          Server Management Menu          SSID: AVZ9
                                     More:          +
1 ISPF Session      - Display and modify ISPF/AVZ session parameters
2 AVZParms          - Display and modify AVZ main task parameters
3 AVZ Blocks        - Display formatted AVZ control blocks
4 AVZ Stats         - Display AVZ product statistics
5 AVZ Tokens        - Display and control product tokens
6 AVZ Modules       - Display product module information
7 AVZ Tasks         - Display product tasks
8 AVZ IP Tree       - Display the IP address tree
9 AVZ Prcs Blks    - Display the Cross Memory Process Blocks
10 AVZ RPC          - RPC Control Facility
11 AVZ Copies       - Display information about each copy of the product
12 AVZ Storage      - Display virtual storage information
13 Trace Archive    - Server Trace Archive Facility
14 AVZ Group        - Display all remote users in a group
15 NLS Tables       - Display National Language Support tables
16 Link             - Display Link Tables
17 RRS              - Display RRS Facilities
18 SOM              - Display and control Security Optimization and Management
```

Viewing Details about a Parameter

Complete the following steps to view parameter details.

Procedure

1. To the left of the parameter group you would like to view, type **D**, for display.
2. Press **ENTER**. The system will display the **Parameters** panel, showing a listing of all parameters in the selected parameter group as well as their default values.

In this example, the **PRODIMS** group is displayed:

```

----- Parameters ----- Scr 1 Row 1 of 87
LCs: D Display  E Edit  F Format  P Print CB  S Show CB

Parameter          Parameter
Description        Value
ACTIVATE IMS/ODBA SUPPORT      NO
APPLICATION GROUP NAME        'NONE'
CHECK FOR NOMFS IN TRANSACTION NO
CHECK IMS PSB USER ACCESS     YES
CONVERT NULLS TO BLANKS - IMS SERVER NO
CONVERT 3F TO THIS HEX VALUE  X'3F'
DDNAME USED TO ALLOCATE RESLIB 'CCTLDD'
DRA TERM TIMEOUT VALUE        10
DSNAME OF THE DRA RESLIB      'IMS.IFA4.SDFSRESL'
FAST PATH BUFFERS PER THREAD  1
FAST PATH OVERFLOW BUFFERS    1
FUNCTION LEVEL OF PRODUCT REGION X'03'
IDENTIFY RETRY WAIT TIME      60
IMS DLI PARAMETER LIST LOCATION ABOVE
IMS RCLASS VALUE             'IMS'
IMS RECONNECT INTERVAL        300 SECONDS
Command ===>                               Scroll ==> PAGE

```

- To the left of any particular parameter, type D to display more information. In the above example, more information about the parameter **ACTIVATE IMS/ODBA SUPPORT** is shown.
- Press **ENTER**. The system will display the **Parameter Information** panel, showing an explanation of the chosen parameter:

```

BROWSE   Parameter Information                               Line 0000000000 Col 001 064
***** Top of Data *****
PARM     IMSODBA

MESSAGE  ACTIVATE IMS/ODBA SUPPORT

EXPLAIN  The IMSODBA parameter controls whether the system will
         initialize the IMS/DB ODBA interface.
***** Bottom of Data *****

```

- Use the **END** command, or press the **F3** key, to return to the **Parameters** panel.
- To the left of any parameter, type F to view information about the parameter value. In this example, the parameter **ACTIVATE IMS/ODBA SUPPORT** is shown.
- Press **ENTER**. Another **Parameter Information** panel appears, showing parameter name, description text, whether it is updatable or read-only, maximum and minimum values, and the parameter value.

```

BROWSE   Parameter Information                               Line 0000000000 Col 001 061
***** Top of Data *****
Parameter Name          IMSODBA
Description Text        ACTIVATE IMS/ODBA SUPPORT
Group Name              PRODIMS
Updatable Parameter     N
Read-Only Parameter     N
Maximum Value           0
Minimum Value           0
Parameter Counter       69
Last Update Timestamp
Set During Initialization 0
Changed During Initialization 0
Set After Initialization 0
Changed After Initialization 0
Last Update Userid
Parameter Value         NO
***** Bottom of Data *****

```

- Use the **END** command, or press the **F3** key to return to the **Data Virtualization Server Parameters** panel.

Modifying a started task parameter

You can modify or update a started task parameter using the Data Virtualization ISPF application.

Procedure

1. Select AVZ Admin from the Data Virtualization Server Primary Option menu.
2. Press **ENTER**. The **Parameter Groups** panel will be displayed:

```
----- Parameter Groups ----- Row 1 of 39
  LCs: D Display  F Format  P Print CB  S Show CB

Parameter      Group
Group          Description
PRODACI        ACI CONFIGURATION PARAMETERS
PRODADABAS     ADABAS PARAMETERS
PRODAIB        ACTIONAL PARAMETERS
PRODAPPCMV     APPC/MVS PARAMETERS
PRODBROWSE     TRACE BROWSE PARAMETERS
PRODCICS       CICS PARAMETERS
PRODCOMM       COMMUNICATIONS PARAMETERS
PRODDMF        DMF (DATA MAPPING FACILITY) PARAMETERS
PRODEVENT      EXCEPTION EVENT PARAMETERS
PRODFILE       FILE PARAMETERS
PRODGLV        GLOBAL VARIABLE PARAMETERS
PRODIDF        INTEGRATED DRDA FACILITY (IDF) PARAMETERS
PRODIDMS       IDMS PARAMETERS
PRODIMS        IMS PARAMETERS
PRODJAVA       JAVA PARAMETERS
PRODLICENSE    LICENSING PARAMETERS

Command ==>>>                               Scroll ==>> PAGE
```

3. To the left of the selected parameter group, type D to display the parameters in the group. In this example, the group **PRODIMS** is displayed.
4. Press **ENTER**. The **Parameters** panel will be displayed, showing a list of all the parameters in the selected parameter group, as well as their default values:

```
----- Parameters ----- Scr 1 Row 1 of 87
  LCs: D Display  E Edit  F Format  P Print CB  S Show CB

Parameter      Description      Parameter      Value
Description
ACTIVATE IMS/ODBA SUPPORT      NO
APPLICATION GROUP NAME        'NONE'
CHECK FOR NOMFS IN TRANSACTION NO
CHECK IMS PSB USER ACCESS     YES
CONVERT NULLS TO BLANKS - IMS SERVER NO
CONVERT 3F TO THIS HEX VALUE  X'3F'
DDNAME USED TO ALLOCATE RESLIB 'CCTLDD'
DRA TERM TIMEOUT VALUE        10
DSNAME OF THE DRA RESLIB      'IMS.IFA4.SDFSRESL'
FAST PATH BUFFERS PER THREAD  1
FAST PATH OVERFLOW BUFFERS    1
FUNCTION LEVEL OF PRODUCT REGION X'03'
IDENTIFY RETRY WAIT TIME      60
IMS DLI PARAMETER LIST LOCATION ABOVE
IMS RCLASS VALUE              'IMS'
IMS RECONNECT INTERVAL        300 SECONDS

Command ==>>>                               Scroll ==>> PAGE
```

5. For the selected parameter, simply type over the existing value in the **PARAMETER VALUE** column. If the parameter value is longer than 21 characters (including the single quote characters) or contains mixed case characters, then type E to the left of the parameter to enter the **Edit** dialog to type over the existing parameter value.

Note: If the value is not updatable, you will not be able to type over it.

6. Press **ENTER** to save the change. If the value was successfully modified a VALUE/S MODIFIED message will be displayed in the upper-right corner.

7. Use the **END** command, or press the **F3** key, to return to the **Parameters** panel.

Adding a started task parameter to IN00 file

This section describes on how to add or modify a new started task parameter to the Server IN00 file.

Procedure

1. From the **ISPF primary option menu**, select the option **3 Utilities**.
2. Choose the option **4 Dslist**.
3. In the DSLIST option, enter *hlq.SAVZSAMP* (SAVZIN00) in **Dsname Level**. *hlq* is the high level qualifier.
4. Type **V** next to the data set *hlq.SAVZSAMP* (SAVZIN00) to view the files under the dataset.

```

Menu Options View Utilities Compilers Help
-----
DSLIST - Data Sets Matching hlq.SAVZSAMP                               Row 1 of 13
Command - Enter "/" to select action                               Message                               Volume
-----
V hlq.SAVZSAMP                                                    VOLSER
  hlq.SAVZSAMP.BK012420                                           VOLSER
  hlq.SAVZSAMP.BK121419                                           VOLSER
  hlq.SAVZSAMP.CB                                                 VOLSER
  hlq.SAVZSAMP.GW                                                 VOLSER
  hlq.SAVZSAMP.NEW                                                VOLSER
  hlq.SAVZSAMP.OLD                                                VOLSER
  hlq.SAVZSAMP.RPW                                                VOLSER
  hlq.SAVZSAMP.STRESS.TEST                                       VOLSER
  hlq.SAVZSAMP.ZOS12                                              VOLSER
  hlq.SAVZSAMP.ZOS2                                              VOLSER
  hlq.SAVZSAMP.ZOS3                                              VOLSER
  hlq.SAVZSAMP.ZOS3.D090904                                       VOLSER
***** End of Data Set list *****

```

5. Find your server's IN00 files using the find command. In the following example, *XYZY* is the server ID.

```

Menu Functions Confirm Utilities Help
-----
VIEW          hlq.SAVZSAMP                               Row 0000001 of 0000925
Name          Prompt          Size   Created          Changed          ID
-----
$A            1      2004/04/19      2004/04/19 17:20:05 ABCDEFI
#H            5      2000/06/27      2000/06/27 10:30:35 ABCDEFF
#SUB         18      2000/06/27      2000/06/27 10:26:01 ABCDEFF
#USAGE       48      2000/06/27      2000/06/27 10:46:23 ABCDEFF
@DB2IN0A    490     2003/04/08      2015/07/05 23:08:36 ABCDEFW
ADBAINEF    53      2013/03/07      2013/03/07 15:05:58 ABCDEFA
ADBAINNEY   205     2004/04/20      2015/07/05 23:08:36 ABCDEFW
ADBAINHL    786     2004/04/20      2015/07/15 18:24:05 ABCDEFA
ADBAIN00   1069    2004/04/20      2017/03/10 17:49:01 TSABH
ADBC        17      2010/03/04      2010/03/04 14:30:43 ABCDEF2
ADBCINEF    48      1999/09/23      1999/09/23 17:33:47 ABCDEFW
ADBCINGW   1212    2006/05/06      2015/07/05 23:08:36 ABCDEFW
ADBCIN0#   283     2000/11/15      2015/07/05 23:08:36 ABCDEFW
ADBCIN00   1464    2005/04/01      2020/04/17 09:56:02 TABDC7
ADBCIN01    288     2001/07/18      2015/07/05 23:08:36 ABCDEFW
ADBCIN02    906     2008/01/16      2015/07/05 23:08:37 ABCDEFW
Command ==> F XYZYIN00                               Scroll ==> PAGE

```

6. Type **V** to open the IN00 file in a read mode or **E** to open the IN00 file in a write mode.
7. Find the suitable section to add a parameter. The sections are grouped under **IMS**, **SMF**, **TRACEBROWSE**, etc. In the following example, the parameter **SMFFULLSQL** is added under **SMF** section.

```
000403 /*-----*/
000404 /* SET SOME SMF PARAMETERS */
000405 /*-----*/
000406 IF 1 = 1 THEN DO
000419 "MODIFY PARM NAME(SMFFULLSQL) VALUE(YES) "
```

If you have opened the file using **E** or edit mode, the changes will be saved immediately.

Chapter 3. General administrative functions

The Primary Option Menu for Data Virtualization Manager server provides access to interface facilities and administrative functions.

```

IBM Data Virtualization Manager for z/OS

Interface Facilities:
1 ACI          5 IDMS
2 Adabas      6 IMS
3 CICS        7 VSAM/Sequential
4 DB2         8 DSSPUFI

SSID      : xVZy
Version   : vv.yy.mm
Date      : yy/mm/dd
Time      : hh:mm

Data Virtualization Server Administration:
A Remote User      - Manage Remote Users
B Server Trace     - Server Trace Facility - SIS SSID: AVZ9
C AVZ Admin.       - Manage Data Virtualization Server
D Data Mapping     - Data Mapping Facility
E Rules Mgmt.     - Event Facility Management
F Monitor          - Monitor Server Activity
G Streams          - Streams Administration
H Services         - Services Administration
I Instrumentation  - Instrumentation Server Administration
S Sysplex Functions - Sysplex Enabled Commands

Option ===>

```

Type the number or letter that corresponds to the task that you want to perform.

The information in the upper right part of the panel includes the following information:

- SSID: The ID of the Data Virtualization Manager server subsystem to which you are connected.
- Version: The version, release, and maintenance number of Data Virtualization Manager server.
- Build: The build number or SVFX of the product.
- Date: The date this version and build originated.
- Time: The time this version and build originated.

Table 1 on page 9 describes the interfaces to various data sources.

Option	Description
ACI	The ACI API allows clients to connect to backend programs in remote TP environments. See “IBM Data Virtualization Manager for z/OS Interface for ACI” on page 390
Adabas	The Data Mapping facility (DMF) allows for the creation of Adabas data maps for Server. See “ Adabas” on page 412
CICS	The Server CICS Control Facility allows you to view and control the Server CICS environment. See “IBM Data Virtualization Manager for z/OS Interface for CICS/TS” on page 413.
DB2	The Server Database Control application allows you to view and modify the product Server Database table. See “DB2” on page 423
IDMS	The Server IDMS Control Facility allows you to monitor and control your access to IDMS/DC and IDMS/DB. See
IMS	The Server IMS Control Facility allows you to monitor and control your access to IMS/TM and IMS/DB. See “IBM Data Virtualization Manager for z/OS Interface for IMS DB: support for DBCTL” on page 425

Option	Description
VSAM/Sequential	The Data Mapping facility (DMF) allows for the creation of VSAM/ Sequential data maps for Server. See “VSAM and sequential files” on page 437.
DSSPUFI	Evaluate SQL statements and view the results.

Table 2 on page 10 describes the options in the primary ISPF menu.

Option	Description
Remote User	“Managing users” on page 10
Trace Browse	Trace Browse Facility
AVZ Admin	Chapter 3, “General administrative functions,” on page 9
Data Mapping	Manage the Data Mapping Facility (DMS). See “Using the Data Mapping Facility” on page 443
Rules Mgmt.	Event Facility Management. See “Limiting connections” on page 15
Monitor	Monitor Server Activity. See Chapter 16, “Monitoring,” on page 363
Streams	Streams Administration
Services	Services Administration
Instrumentation	Instrumentation Server Administration. See “Instrumentation Server” on page 366.
Sysplex Functions	Sysplex Enabled Commands. See Chapter 15, “Sysplex Function,” on page 359

Managing users

The **Manage Remote User** program allows you to view current and cumulative information about users connected to the IBM Data Virtualization Manager for z/OS.

Procedure

The type of information that you can view using this program includes:

- General information about the user, the link, and the security status of the host.
- TCP/IP-specific information. If the link is not a TCP/IP link, the fields contain asterisks.
- Information related to SQL processing. You can follow the progress of an application as it issues SQL statements.
- Information about the amount of time that the application is using in processing and in communication.
- Information about data compression and the amount of data that the application has transmitted and received.
- Information about the amount of time the application has spent processing and transmitting data.

To view information about a remote user:

1. From the **Primary Option Menu** panel, enter **A** to choose the option **Remote User**.
2. Press **ENTER**. The **Remote Users** panel is displayed.

The following table describes each column name on the Remote User panel and provides a sort name (if available) for each column.

Column name	Description	Sort name
HOST USERID	The user ID of the remote user.	USER
LAN USERID	The LAN user ID of the remote user.	LAN
HOST NAME	The link that is being used. For a local user, the name of the remote system that is being accessed. For a remote user, the name of the remote system that is accessing the local system.	HOST
LINK TYPE	The communication protocol.	TYPE
APPLICATION NAME	The application name that is specified in the APNA (Application Name) keyword of the ODBC data source.	APPLICATION
USER PARAMETER	The parameter that is specified in the USERPARM (Host User Parm) keyword of the ODBC data source.	PARAMETER
TCP/IP ADDRESS	The 4-byte IP address of a node.	IPADDR
REMOTE PORT	The port that is used by the remote Data Virtualization Manager server.	REMOTE
LOCAL PORT	The TCP/IP port that is used by the remote Data Virtualization Manager server.	LOCAL
IUCV PATH	The token that is used by Data Virtualization Manager server to communicate with TCP/IP.	PATH
SOCKET NUMBER	The number of the TCP/IP session.	SOCKET
DB2 SUBS	The DB2 subsystem to which the remote user is connected.	DB2
PLAN NAME	The name of the plan that was used to open a DB2 thread.	PLAN
SQL REASON	The most recent DB2 reason code.	REASON
SQL CODE	The most recent SQLCA SQLCODE value.	SQLCODE
SQL STMT-TYPE	The SQL verb.	SQLTYPE
STATEMENT NUMBER	The SQL statement number assigned by the preprocessor.	STMTNO
CURSOR NUMBER	The number of the cursor that is being used.	CURSOR
LOCKS HELD	The number of locks held.	
PROGRAM NAME	The Data Virtualization Manager server transaction program name.	PROGRAM

Column name	Description	Sort name
CPU TIME	The total amount of CPU time.	
SQL COUNT	The total number of SQL operations, which includes SQL run, RPCs and stored procedures that are run, rollbacks and commits initiated from the client through an ODBC call, and operations to turn auto-commit on or off.	SQLCOUNT
CONNECT TIME	The total amount of elapsed time.	CONNECT
CURRENT STATE	One of the following states: PROCESS, SEND, RECEIVE.	STATE
STATE DURATION	The amount of time that is spent in the current state.	DURATION
FUNCTION CODE	The type of Data Virtualization Manager server processing.	FUNCTION
GENERIC USERID	The DB2 generic ID for the connection.	GENERIC
DB2 G-MBR	The group attachment member name.	G-MBR
DB2 TOKEN THREAD	The DB2 token thread value.	TOKEN
DB2 REQUESTING SITE NAME	The DB2 requesting site name.	
TOTAL SENT (KB)	Cumulative outbound data. One of the following: RAW (before compression) COMPRESSED (after compression) PERCENT $(1 - (\text{COMPRESSED} / \text{RAW})) * 100$	TOSENTR TOSENTC TOSENTP
CURRENT SENT	Last outbound transmission. One of the following: RAW (before compression) COMPRESSED (after compression) PERCENT $((1 - (\text{COMPRESSED} / \text{RAW})) * 100)$	CUSENTR CUSENTC CUSENTP
TOTAL RECEIVED (KB)	Cumulative inbound data. One of the following: RAW (before compression) COMPRESSED (after compression) PERCENT $((1 - (\text{COMPRESSED} / \text{RAW})) * 100)$	TORECVR TOREVCV TORECVP

Column name	Description	Sort name
CURRENT RECEIVED	Last inbound transmission. One of the following: RAW (before compression) COMPRESSED (after compression) PERCENT $((1-(\text{COMPRESSED}/\text{RAW}))*100)$	CURECVR CURECVC CURECVP
TELEPROCESSING PERCENT	The percentage of the total data transfer time.	TPPERCNT
HOST PROCESSING PERCENT	The percentage of the total data extraction time.	HOPERCNT
ENCLAVE COUNT	The number of WLM enclaves created for the user.	COUNT
ENCLAVE CPU	The enclave CPU time, in seconds.	ENCLAVE
TOTAL CPU	The total amount of CPU time, in seconds.	CPU
ACEE SOURCE	The source of the ACEE.	ACEE
EXTENDED CLIENT	The extended client user ID, workstation name, and application name.	EXTENDED

Note: Use the **PF10** key to scroll LEFT and **PF11** key to scroll RIGHT when viewing the columns in the panel.

3. On the Remote User panel, type one of the following line commands, and press Enter:

Command	Description
A	Lists the DB2 secondary authorization IDs.
C	Cancels the thread.
E	Starts the SQL EXPLAIN program, which requires MVS/Quick-Ref.
F	Formats the row.
I	Displays user information.
K	Terminates the user's connection with the Data Virtualization Manager server. To use this command, you must have UPDATE authority to the USERS resource or be using the same user ID as the connection that is being terminated. The task that supports the remote client fails with an X '222' abend. A reason code is not associated with this event.
P	Prints the control block.
S	Displays the control block.
T	Displays user trace. User trace provides summary information about user activity.

Command	Description
U	Displays user detail. User details provide resource utilization information.

Terminating a user connection

In the Remote User program, the **Kill** command is used to terminate a remote user's connection with the Data Virtualization Manager server. This command closes the TCP/IP session along with the driver. When a remote user's connection is terminated, the task supporting the remote client fails with an X '222' abend. A reason code is not associated with this event.

Before you begin

The **Kill** command can only be issued by a user with authorization to do so. Authorization is granted in either of the following cases:

- When the user is granted UPDATE authority to the USERS resource.
- When the user ID attempting to kill the connection is the same as the userID of the driver being killed. In this case, UPDATE authority is not checked.

About this task

The Server Trace program shows the following information:

- Authorization request for the kill operation
- Abend of the remote user's thread
- Close the server of the remote session

The **Kill** fails if the driver stops before the command runs. Typically, failure occurs when the **Kill** command is entered after the Remote User panel is requested.

Note: The Remote User panel does not automatically update after a user connection is stopped.

Handling user connections

Data Virtualization Manager server allows to configure the number of users who are logged on the server. At any one time, the number of users logged on the Data Virtualization Manager server cannot exceed the maximum number of users for which the server is licensed.

If a user tries to log on after the maximum number of users is reached, the Data Virtualization Manager server rejects the user or places the user in a queue until another user logs off, depending on the configuration of the Data Virtualization Manager server.

You can configure how you want the Data Virtualization Manager server to handle connections that exceed the licensed number of users.

Rejecting connections

To reject connections when the allowed number are exceeded, use the MODIFY PARM command in the IBM Data Virtualization Manager for z/OS Initialization EXEC, AVZSIN00, to set the CONCURRENTMX parameter:

```
"MODIFY PARM NAME(CONCURRENTMX) VALUE(xxxx) "
```

where *xxxx* is the maximum number of concurrent DB2 users. This value is a number 0 - 2000. When this value is reached, the Data Virtualization Manager server rejects further connections and returns an error message to the client.

Queuing connections

To queue connections, use the MODIFY PARM command in the IBM Data Virtualization Manager for z/OS configuration member, AVZSIN00, to set the REUSETHEADS and TARGETTHREADCOUNT parameters:

```
"MODIFY PARM NAME(REUSETHEADS) VALUE(YES) "  
"MODIFY PARM NAME(TARGETTHREADCOUNT) VALUE(XXXX) "
```

where:

PARM NAME (REUSETHEADS) controls whether threads are reused. If this flag is set, each thread is reused a number of times if possible. If this flag is not set, a new thread is always created for each new inbound session. Thread reuse may reduce CPU resource utilization considerably when DB2 threads are used frequently and/or client user IDs are cached and reused for persistent session support. Set to YES.

PARM NAME (TARGETTHREADCOUNT) limits the total number of ODBC and JDBC transaction processing threads allowed in the system. The system dynamically attaches up to this number of subtasks during product execution to handle requests that arrive on TCP/IP and MQ/Series sessions. The value of this parameter can be a number from 1 to 1000.

The TCPIP listener task issues an ACCEPT command and waits for new connection requests. When a new connection arrives, the listener checks the thread limit parameters. It then either attaches a new z/OS transaction subtask or pairs the newly accepted connection with a waiting or reusable transaction thread. If the number of connections exceeds the TARGETTHREADCOUNT value, the listener task waits until the count decreases. During this waiting period, the server does not accept any new connections.

When a connection is released, the new connection is allowed to connect. This support typically works best with applications that use coded logic to connect and reconnect frequently based on the work performed, rather than allowing idle connections to remain. This also works with IBM Data Virtualization Manager for z/OS's Virtual Connection Facility support, which controls connections that are based on the units of work.

Limiting connections

Using the Event Facility, you can set specific times for connection limits by modifying Data Virtualization Manager server parameters by using the SEF /*CMD rules. These rules can be run by z/OS automatically to run an SEF program in IBM Data Virtualization Manager for z/OS to set parameters to the specified values.

You can also write SEF /*ATH rules that run at each logon to limit a user to a specific number of connections. This prevents a single workstation from connecting to Data Virtualization Manager server multiple times without disconnecting and closing old connections.

Configuring virtual connections

The Virtual Connection Facility (VCF) component increases the number of client connections possible. When a client is active, it uses a standard "real" connection to z/OS. When the client is idle, the VCF option switches the client to a "virtual" connection. All shifts from "real" to "virtual" connections and back are transparent to the client and never interrupt a logical unit of work (LUW).

About this task

The VCF never switches from "virtual" to "real" until requests are ready for execution; that is, if LUW encapsulation is activated, the real connection is not supplied until the LUW is complete at the Data Driver.

On z/OS, the VCF maintains a thread pool, from which all active clients use a thread. When a client is idle, the z/OS thread is dropped. Use of a thread pool eliminates the overhead and time that is spent in z/OS thread creation, database thread creation, and security checking. The thread pool automatically shrinks and expands based on current activity levels for optimum performance that is balanced with low z/OS resource usage.

Set the following parameter in the AVZSIN00 configuration member:

```
"MODIFY PARM NAME(HOSTFUNCTIONALLEVEL) VALUE(x)"
```

Parameter	Description	Valid values
HOSTFUNCTIONALLEVEL	Shows what level of code the host is running. This value is passed back to the client so that the client knows what host capabilities are usable. This parameter cannot be set and is intended for technical support use only.	4 (default)

Creating multiple administrative users

The IBM Data Virtualization Manager for z/OS allows you to create and delete multiple administrative users to access the JDBC Gateway web console. The newly created administrative users can change the default password to access the JDBC Gateway console.

Before you begin

- The JDBC Gateway must be installed and running. See Installing JDBC Gateway section.
- The user must have access to the JDBC Gateway web console.

About this task

Follow these steps to create and delete multiple administrative users, and change the default password to unique credentials:

Procedure

1. You can run the AddUser, Changepassword, and DeleteUser commands in the command prompt. Alternatively, create a batch script file containing the curl commands with user inputs.

The following are the curl commands for AddUser, Changepassword, and DeleteUser operations:

Operation	Curl command
<p>AddUser</p>	<p>This command creates new administrative users with default credentials. Once the new administrative users are created, they can access the JDBC Gateway web console with the default user credentials.</p> <pre data-bbox="574 310 1463 426">curl -u <Installation_UserName>:<Installation_UserPassword> -d "username=<New_Admin_UserName>&password=<Default_Admin_UserPassword>" -H "Content-Type: application/x-www-form-urlencoded" -X POST <JGate_Web_Console_URL></pre> <p>Here,</p> <ul data-bbox="574 491 1463 762" style="list-style-type: none"> • The value <code>Installation_UserName</code> and <code>Installation_UserPassword</code> are the username and password of the user who installed the component. • The <code>New_Admin_UserName</code> is the username of the newly added administrative user, and <code>Default_Admin_UserPassword</code> is the administrative user's default password. • In the <code><JGate_Web_Console_URL></code>, replace <code>gateway</code> with <code>adduser</code> in the URL. <p>Note: The user who installs the application must run the script.</p>
<p>Changepassword</p>	<p>This command changes the default password provided by the installation user of the newly created administrative users.</p> <pre data-bbox="574 940 1463 1014">curl -u <Admin_UserName>:<Admin_UserName> -d "newpassword=<New_Password>" -H "Content-Type: application/x-www-form-urlencoded" -X POST <JGate_Web_Console_URL></pre> <p>Here,</p> <ul data-bbox="574 1094 1463 1344" style="list-style-type: none"> • <code><Admin_UserName></code> is the username of the newly added administrative user. • <code><Admin_UserPassword></code> is the default password provided by the installation user. • <code><New_Password></code> is the new password. • In the <code><JGate_Web_Console_URL></code>, replace <code>gateway</code> with <code>changepassword</code> in the URL. <p>Note: The newly created administrative users must run the script.</p>
<p>DeleteUser</p>	<p>This command deletes the JDBC Gateway administrative users.</p> <pre data-bbox="574 1486 1463 1560">curl -u <Installation_UserName>:<Installation_UserPassword> -d "deleteusername=<Delete_UserName>" -H "Content-Type: application/x-www-form-urlencoded" -X POST <JGate_Web_Console_URL></pre> <p>Here,</p> <ul data-bbox="574 1640 1463 1843" style="list-style-type: none"> • <code><Installation_UserName></code> and <code><Installation_UserPassword></code> are the username and password of the user who installed the component. • <code><Delete_UserName></code> is the username of the administrative user you want to delete. • In the <code><JGate_Web_Console_URL></code> replace <code>gateway</code> with <code>deleteuser</code> in the URL. <p>Note: The user who installs the application must run the script.</p>

- If you are using the batch script files, run it with the curl commands from the Windows command prompt using the user inputs provided in sequence.

Note: The batch files must be created manually.

Displaying and modifying ISPF/AVZ session parameters

The ISPF Session program identifies the subsystem name that is associated with your ISPF session. The settings are saved in the current user's profile variable pool under the ISPF application. This option is only available through the Data Virtualization Manager server ISPF panels

Procedure

To invoke the ISPF Session Parameters option:

- From the **ISPF Primary Menu**, select AVZ Admin and press ENTER.
- From the **Server Management Menu** menu, select option **1** for **ISPF Session** and press ENTER.

The **ISPF Session Parameters** panel is displayed.

```

----- ISPF Session Parameters -----
Subsystem Name (SSID) ==> AVZ9

Status Information:
Product version      ==> 01.01.00
Build number         ==> 0000
Build date           ==> 2023-09-29

Server Trace Facility ==> AVZ9 (SIS Subsystem ID)

```

The following table describes the ISPF panel and provides a sort name (if available).

Column Name	Description	Sort Name
SUBSYSTEM NAME (SSID)	The 4-character subsystem name for the product address space. This name is used to construct the name of the parameter EXEC for the debug facility.	NAME

- To change the subsystem name or the trace browse facility subsystem name for the Instrumentation Server (SIS), type the new name over the existing name and press **ENTER**. The status information updates to reflect the information for the subsystem associated with your ISPF session.

Displaying formatted control blocks

The Data Virtualization Manager server Internal Control Block program displays the contents of critical product controls. The information is formatted to show individual fields and is followed by a hexadecimal dump of the control block controls.

About this task

The following product controls are supported:

Control Block	Description
AECB	Security Optimization Block
BOST	Active Browse Status Block
CICO	CICS Control Area
CIEC	EXCI Control Area

CMAS	Product ASVT block
HSST	DFHSM Raw Recall Statistics
IDCO	IDMS Control Area
IMCO	IMS Control Area
IMDA	IMS/ODBA Control Block
OPCK	Execution Checklist
OPML	Message Lookup Table
OPMS	Product Master Block
OPPA	Product Parameter Table
OPPM	Permanent Data Area
OPPT	Protected Data Area
OPVN	Product Vendor Table Entry
OPVT	Module Address Vector Table
OPWK	System Work Area
OPZV	Services Interface Block
PHCT	Streams Main Control Block
RLCI	Compiler Interpreter Table
RMWB	DFHSM Pending Recall Table
RRCO	RRS Control Area
RXWS	GLOBAL. Work Space
RXWS	GLVEVENT. Work Space
SICA	SICA Sysplex and SIS Control
SMRW	SMRW Concurrent RPC Control
TIMR	Timer DIE Block
TITR	Timer Trace Table
VSSD	VSSD Storage Detail
WMCB	WLM Control Block
WMCB	WLM Control Block

Procedure

To invoke the Data Virtualization Manager serverControl Block Option:

1. From the **ISPF Primary Menu**, select **AVZ Admin** and press **ENTER**.

The **Server Management Menu** is displayed.

2. From the **Server Management Menu**, select option **3** for **AVZ Blocks** and press **ENTER**.

The **Server Internal Control Blocks** panel is displayed.

```

----- Internal Control Blocks ----- Row 1 of 30
LCs: D Display F Format P Print CB S Show CB

Block Control Block          Virtual Storage Pr
Name Description            ASID Address Length Ky Note
AECB Security Optimization Block 0191 7E5AE000 00005030 80
BOST Active Browse Status Block 0191 00001000 00000800 88
CICO CICS Control Area        0191 00008000 00002000 80
CIEC EXCI Control Area        0191 3D8B8000 00001000 80
CMAS Product ASVT block       0191 7F2E1000 000086C8 80
HSST DFHSM Raw Recall Statistics 0191 7F709000 00000100 80
IDCO IDMS Control Area        0191 3D8BA000 00001000 80
IMCO IMS Control Area         0191 0000A000 00001000 80
IMDA IMS/ODBA Control Block    0191 3D8BB000 00000ABC 88
OPCK Execution Checklist       0191 3CA40000 00000360 00
OPML Message Lookup Table     0191 7F70B000 00018000 40
OPMS Product Master Block     0191 32B05000 0000A000 40
OPPA Product Parameter Table  0191 3CE79000 00045CA0 00
OPPM Permanent Data Area      0191 2D0F1000 00003000 40
OPPT Protected Data Area      0191 7F427000 00001000 40
OPVN Product Vendor Table Entry 0191 37556D58 00000200 40
Command ==>                               Scroll ==> PAGE

```

The program supports the following line commands:

- **D**: Displays the product control block specified by the selected row.
- **F**: Formats the block selection entry for the selected row.
- **P**: Prints the block selection entry for the selected row.
- **S**: Displays the block selection entry for the selected row.

In the following example, formatted information of CICS Control Area (CICO) is displayed.

```

BROWSE Internal Control Block Information Line 0000000000 Col 001 053
***** Top of Data *****
Block Name (Acronym) CICO
Internal Entity Name CICO
Virtual Storage Address X'00008000'
Storage Size X'00002000'
Block's ASID X'0191'
Virtual Storage Protect Key X'80'
Short Description CICS Control Area
Description CICS Control Area
***** Bottom of Data *****

```

You can sort the columns according to their sort names. The describes the sort names of the columns available.

Column Name	Description	Sort Name
BLOCK NAME	The name of the control block.	NAME
ASID	The ASID where the control block resides.	ASID
VIRTUAL ADDRESS	The virtual address of block storage.	ADDRESS
STORAGE LENGTH	The virtual size of block storage.	SIZE
PR KEY	The storage protection key.	KEY
DESCRIPTION	The description of the control block.	

Table 3. Sort Names (continued)		
Column Name	Description	Sort Name
NOTE	The last line command action performed.	

3. Type the command to the left of the line and press **ENTER**.

Displaying server statistics

The Data Virtualization Manager server allows to display the server statistic details to diagnose problems.

About this task

Data Virtualization Manager server provides the following statistics that help to diagnose problems.

- **Task Process Blocks.** These blocks contain information needed to represent a dispatchable unit of work in the Data Virtualization Manager server. The blocks also contain space for dynamic save and work areas that are referred to as the stack. The size of the process block depends on the Data Virtualization Manager server features that are licensed.
- **ESTAE Routine Process Blocks.** End-of-task statistics are counts of the number of times the subsystem interface (SSI) was entered for the termination of all tasks and address spaces.
- **ABEND and LOGREC Statistics.** These statistics provide a summary of how often and how many abends are occurring.

Procedure

Perform the following steps to display server statistics:

1. From the **ISPF Primary Menu**, select AVZ Admin and press ENTER.

The **Server Management Menu** is displayed.

2. From the **Server Management Menu**, select option **4** for AVZ **Stats** and press **ENTER**. The **Product Statistics** panel is displayed.

```

----- Product Statistics -----
Command ==>>
Task process blocks
  Number of process blocks in use ..... 1
  Number of attached subtasks ..... 5
PC routines process blocks
  Total PC process block allocations ..... 983632
  Blocks in use ..... 0
  Blocks recaptured ..... 0
  High water clock value ..... 16NOV 18:14:07.185926
  Last allocation failure clock value .....
  Allocation failure count ..... 0
  High water count ..... 6
ESTAE routine process blocks
  Total ES process blocks allocations ..... 0
  Blocks in use ..... 0
  Blocks recaptured ..... 0
  High water clock value .....
  Last allocation failure clock value .....

```

The following topics describe the fields of the server statistics display:

Table 4.	
Server Statistics	Description
Task Process Blocks	These blocks contain information about a piece of work in the DVM Server that needs to be dispatched. They also contain space for dynamic saves and work areas referred to as stack. The size of the process block depends on the licensed DVM server features.
PC Routine Process Blocks	The PC statistics counts the number of times the Sub-System Interface (SSI) was entered to terminate all the tasks and address spaces.
ESTAE Routine Process Blocks	The End-of-task statistics counts the number of times the SSI was entered to terminate all tasks and address spaces.
ABEND and LOGREC Statistics	These statistics summarizes how often and the number of times the abends have occurred.

Managing tokens

The Token Control program allows you to view and manage Data Virtualization Manager server execution tokens. This program is used to determine the status of a token, view token data, and kill tokens.

Procedure

To invoke the Token Control option:

1. From the **ISPF Primary Menu**, select AVZ Admin and press ENTER.

The **Server Management Menu** is displayed.

2. From the **Server Management Menu**, select option **5** for AVZ Tokens.

The **Token control** panel is displayed.

```

----- Token Control ----- Scr 1 Row 1 of 0
LCs: D Display Data F Format K Kill Token P Print CB S Show CB

Token      User      Host      Data
ID         ID        Name     Size  Note
**End**

```

The program also supports the following line commands:

- **D**: Displays the data associated with the token.
- **F**: Formats the data for the selected row.
- **K** Kills (deletes) the token and frees the associated data.
- **P**: Prints the token control block for the selected row.
- **S**: Displays the token control block for the selected row.

The following table describes each column name on the ISPF panel and provides the sort name (if available).

Table 5. Sort Names For Tokens		
Column Name	Description	Sort Name
TOKEN ID	The hexadecimal string that uniquely identifies the token.	ID

<i>Table 5. Sort Names For Tokens (continued)</i>		
Column Name	Description	Sort Name
USER ID	The client user ID that created the token.	USER
HOST NAME	The host name of the client that created the token. This field typically contains the IP address of the client in dot notation.	HOST
DATA SIZE	The size of the data area associated with the token.	SIZE
CREATION TIME	The GMT time when the token and the data area associated with the token were created.	CREATION
LAST USED	The GMT time when the token was last used. Any use of token (access, update) resets the date and time values.	LAST
TIMEOUT	The number of seconds when token can go unused before it is timed out. Note that this is always from the last use of any kind.	TIMEOUT
USER DATA	The user data that was specified when the token was created. This is not the same as the data associated with the token.	DATA

3. Type the desired command to the left of the line and press **ENTER**.

Displaying product tasks

The tasks program provides current and cumulative information about Data Virtualization Manager server. This section describes on how to invoke the product tasks program and the available commands on the program.

Procedure

Perform the following steps to invoke the product tasks option:

1. From the **ISPF Primary Menu**, select AVZ Admin and press ENTER.

The **Server Management Menu** is displayed.

2. Select option **7** for AVZ Tasks.

The **Tasks** panel is displayed.

```

----- Tasks ----- Scr 1 Row 1 of 100
LCs: C Cancel Thread F Format K Kill Task P Print CB S Show CB
     T SQL Trace      U User Detail

TCB      Connection  Execution  Program   CPU
Address  ID              State      Name      Time      Note
008BDBA0 001332FD      Running    TRACE     121.554S
008BD6F0 001332FF      Running    OPCIEC    004.861S
008A4218 00133313      Running    ZCPMGR    004.107S
00899AE8 00133318      Running    OPCKLM    002.705S
008B6850 00133303      Running    OPIMSR    .735354S
008B2220 0013330A      Running    GLVA      .607014S
008B6220 00133305      Running    OPOACN    .549330S
008BD3D8 00133302      Running    OPRRRM    .538958S
008A5A60 00133310      Running    SEFFULL   .300114S
008A4080 00133315      Running    OPMAXM    .179258S
008995C8 00133317      Running    OPSGIN    .172620S
008BD888 001332FE      Running    OPMAEF    .167719S
0088FAF0 0013333C      Running    OPMAEC    .072843S
0086D128 00133342      Running    OPMAEC    .072394S
008F71A0 001332FC      Running    OPINAS    .056643S
Command ===>

```

Scroll ===> PAGE

Table 6 on page 24 describes each column name on the ISPF panel and provides a sort name (if available).

Column Name	Description	Sort Name
TCB ADDRESS	The name of the product module.	TCB
CONNECTION ID	The address of the product module.	CONNECTION
EXECUTION STATE	The status of subtasks.	
PROGRAM NAME	The product module size.	PROGRAM
CPU TIME	The amount of CPU time used by the TCB.	CPU
SMAF ADDRESS	The address of the SMAF control block.	SMAF
TASK TYPE	A description of the type of task.	TASK

3. Type the desired command to the left of the line and press ENTER.

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents. It also supports the primary SORT and LOCATE commands.

The program also supports the following line commands:

Line Commands	Description
C	Cancels the thread.
F	Formats the information for the selected row.
K	Kills the selected task.
P	Prints the associated control block for the selected row.
S	Displays the control block for the selected row.

Table 7. Line Commands (continued)	
Line Commands	Description
T	Displays an SQL trace for the selected task.
U	Displays user detail for the selected row.

Displaying product copies

The product control program provides information about different copies of Data Virtualization Manager server in use. This is a view-only program.

Procedure

Perform the following steps to invoke the product copies program:

1. From the **ISPF Primary Menu**, select AVZ Admin and press ENTER.

The **Server Management Menu** is displayed.

2. Select option **11** for AVZ Copies.

The **Product Control** panel is displayed.

```

----- Product Control ----- Scr 1 Row 1 of 15
  LCs: F Format  P Print CB  S Show CB

Product  Group   Product  Network  Transfer  Current  Note
Name     Name     Status   Type     Status    Users
AVZG     None     Up       OTC/IP   Enabled    0
AVZM     None     Up       OTC/IP   Enabled    0
AVZS     None     Up       OTC/IP   Enabled    0
AVZU     None     Down     None     None       0
AVZW     None     Up       OTC/IP   Enabled    0
AVZY     None     Up       OTC/IP   Enabled    0
AVZ1     None     Up       OTC/IP   Enabled    0
AVZ2     None     Up       OTC/IP   Enabled    0
AVZ3     None     Up       OTC/IP   Enabled    3
AVZ4     None     Up       OTC/IP   Enabled    0
AVZ5     None     Up       OTC/IP   Enabled    0
AVZ6     None     Up       OTC/IP   Enabled    0
AVZ8     None     Down     OTC/IP   Disabled   0
AVZ9     None     Up       OTC/IP   Enabled    1
AVZX     None     Up       OTC/IP   Enabled    0
**End**
Command ==>>>                               Scroll ==>> PAGE

```

Table 8 on page 25 describes each column name on the ISPF panel and provides a sort name (if available).

Table 8. Column Names		
Column Name	Description	Sort Name
PRODUCT NAME	The 4-character name of the server subsystem (started task) with which this ISPF session is to communicate.	PRODUCT
GROUP NAME	The name of the load balancing group.	GROUP
PRODUCT STATUS	The status of the server (for example, up or down).	STATUS

<i>Table 8. Column Names (continued)</i>		
Column Name	Description	Sort Name
NETWORK TYPE	The type of network on which the Server is running (for example, TCP/IP or SNA using LU6.2 protocol).	NETWORK
TRANSFER STATUS	The status of transfer.	
CURRENT USERS	The number of current users of the server.	
PRODUCT VERSION	The product version and release number.	
HI-WATER USERS	The highest number of users connected concurrently.	
LICENSED MAXIMUM	The maximum number of users allowed by license.	
INSTALLATION MAXIMUM	The maximum number of users allowed.	
UNALLOC<16 MEG	The size of the free storage in the blocks of virtual storage allocated to this server that reside below the 16 MB line.	
UNALLOC>16 MEG	The size of the free storage in the blocks of virtual storage allocated to this server that reside above the 16 MB line.	
MINIMUM <16 MEG	The lowest size of the owned blocks of virtual storage allocated to this server that reside above the 16 MB line minus the amount of free storage in those blocks.	
MINIMUM>16 MEG	The lowest size of the owned blocks of virtual storage allocated to this server that reside above the 16 MB line minus the amount of free storage in those blocks.	
ALLOCATED<16 MEG	The size of the owned blocks of virtual storage allocated to this server that reside below the 16 MB line.	
ALLOCATED>16 MEG	The size of the owned blocks of virtual storage allocated to this server that reside above the 16 MB line.	
SSCT ADDRESS	The address for the subsystem control block.	

<i>Table 8. Column Names (continued)</i>		
Column Name	Description	Sort Name
SSVT ADDRESS	The address for the subsystem vector table.	
OPMS ADDRESS	The address for the main product control block.	
OPPM ADDRESS	The address for the product permanent data area.	
ASID VALUE	The MVS address space index.	

3. Type the desired command to the left of the line and press ENTER.

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents. It also supports the primary SORT and LOCATE commands.

The program also supports the following line commands:

<i>Table 9. Line Commands</i>	
Line Commands	Description
F	Formats the information for the selected item.
P	Prints the control block information related to the selected item.
S	Displays the control block information related to the selected item.

Displaying 64-bit buffer information

The ISPF Session program displays the 64-bit buffer information. This information will help customers better manage buffer allocation and see how allocations are happening. Using this ISPF panel, user can set the storage buffer values and improve the performance.

Procedure

To display the 64-bit buffer information:

1. From the **ISPF Primary Menu**, select **AVZ Admin** and press **ENTER**. The server management menu panel is displayed.
2. From the **Server Management Menu**, select option **12** for **AVZ Admin** and press **ENTER**. The virtual storage information panel is displayed.
3. From the **Virtual Storage Information** menu, select option **10** for **AVZ Buffer Pools** and press **ENTER**. The **64-Bit Storage Pool** panel is displayed.

```

----- 64 Bit Storage Pool ----- Row 1 of 11
Command
===>
    SCROLL ===> CSR
LCs: F Format P Print CB S Show CB

    Size(mb)  Blocks    Current  Max    Full  Requests  Total Pool
              Size(mb)
Note
 24          128       0        4      0     9         3072      BROWSED
 1           16       0        0      0     0          16        BROWSED
 2           256      0        4      0    22         512        FORMATTED
 3            8       0        0      0     0           24
 4          1024      0        2      0    28         4096
 6            4       0        2      0     4           24
16           16       0        0      0     0           256
64           16       0        0      0     0          1024
128          8       0        0      0     0          1024
256          8       0        0      0     0          2048
512          4       0        0      0     0          2048
**End**

```

The program supports the following line commands:

- F: Formats the buffer info in detail for the selected row.
- P: Prints the buffer info entry for the selected row.
- S: Displays the control block information for the selected row.

Type the command to the left of the line and press **ENTER**. For example, the formatted buffer information for the selected line is displayed when “F” line command is entered.

```

BROWSE 64 Bit Storage Pool      Line 0000000000 Col 001 049
Command ===>
===> CSR
                                                                    SCROLL
*****
Pool Size(MB)                24
Number of Blocks              128
Current Buffer Count           0
Max Number of Blocks          4
No Buffer available Count      0
Total Number of Requests      9
Total Pool Size(MB)           3072
**End**

```

The following table describes each column name on the ISPF panel and 64-bit buffer information.

<i>Table 10. 64-bit buffer information</i>	
Column	Description
Size	This column displays the buffer size in MB.
Blocks	This column displays the total number of buffer pools allocated.
Current	This column displays the number of pool buffers currently in use.
Max	This column displays the high-water mark for the buffer pool.
Full	This column displays the number of times the buffer pool was full and the number of times the storage request had to be issued to get a buffer.
Requests	This column displays the number of requests serviced by this buffer pool.

<i>Table 10. 64-bit buffer information (continued)</i>	
Column	Description
Total Pool Size	This column displays the product of Size and Blocks column.
Note	This column displays the action performed on each line.

Chapter 4. Rules and events

Using a rule, you can configure an automatic response to an event. For example, you can allow a critical application to download data any time, and allow a non-critical application to download data only during specific hours.

To enable a rule, one needs UPDATE access to the SAVZXVTB data set since Data Virtualization Manager applies updates to the VTOC on the volume where SAVZXVTB resides.

For example, to restrict the number of times that a user ID can log on to the server, create a LOGON rule to limit the user ID to three logons a day and to take a specific action if the user ID tries to log on more than three times.

Rules and rule sets

The rules and rule set can consist of different components.

A rule can have the following parts:

- Criterion
- Header statement
- One or more process sections
- Return values
- Variables

Automatic limits

A rule can include customizable limits that control many aspects of your configuration including queries, connections, and sessions.

Rules are configured in the Data Virtualization Manager configuration member that is shipped in data set member `h1q.SAVZEXEC (AVZSIN00)`.

You can view rules by selecting `E (AVZ Admin.) > 2 (AVZ Params)` from the Primary Option Menu. To modify a rule, locate the parameter, change its value, and press **Enter**. This modifies the parameter for the existing Data Virtualization Manager session. To make the change permanent, modify the parameter in the AVZSIN00 configuration member.

During installation, a default value is specified for each of the following limits.

Overall per session CPU limit

When this limit is reached, the session is automatically terminated. The security product or a product parameter can provide the limit.

Per Db2® connection CPU limit

When this limit is reached, the current Db2 connection is automatically terminated, and all associated Db2 resources are released.

Per SQL query CPU limit

When this limit is reached, the current SQL query is automatically terminated, and all associated Db2 resources are released.

Inactivity time-out

This limit automatically terminates the session of any user that is inactive for the specified period. Use this limit to minimize security exposures and release resources that are held by inactive users.

Maximum timer-on limit

This limit prevents the execution of any SQL statement that exceeds a specified value. The limit prevents excessive resource utilization.

Maximum rows limit

This limit restricts the number of rows that a query returns.

RPC maximum rows limit

This limit restricts the number of rows that an RPC can generate. This limit is set as a host parameter and is enforced on the host (Data Virtualization Manager server).

Dropped connection detection

This mechanism detects clients that failed or are no longer connected to the network. When a dropped connection is detected, the host session is terminated, and all resources are released.

Lock control facility

This mechanism detects clients that are holding a Db2 lock (share, update, or exclusive) for an excessive period. When the limit is reached, the session is terminated, and the lock is released.

Dynamic SQL control facility

This mechanism allows dynamic SQL to be rejected on the host. Use this mechanism to enforce the use of static SQL.

Maximum concurrent users

This limit controls the maximum number of concurrent users and is enforced on the host.

Variables for rules

When you create a rule, you can use dynamic variables, global variables, temporary variables, and event-specific variables. These variables are used in REXX programming.

Dynamic variables

Dynamic variables are created when the process section of a rule references or sets the value of a simple or compound variable. Dynamic variables exist only while a rule runs and are freed when the REXX environment is deleted. Dynamic variables cannot be accessed by non-REXX procedures and functions. The following code fragment shows two simple variables, I and COUNT, and one compound variable, stemvar.I:

```
do I = 1 to COUNT
  stemvar.I = "InitValue"
end
```

Global variables

Global variables have one of the following stem values:

- GLOBAL
- GLOBAL n , where n is an integer 1 - 9

Global variables can be created, modified, or managed by selecting option **E** (Rules Mgmt.) from the Data Virtualization Manager server - Primary Option Menu and then selecting **1** (Global Variables). To create a new global variable, enter *S variable_name* and press Enter.

Global variables are stored in the global variable checkpoint data set. When a global variable is referenced, the value of the variable is retrieved from the checkpoint data set. The value of a global variable persists across restarts of the product and is shared by all rules. If the **SEFGLVEVENTS** parameter is set to YES in the Data Virtualization Manager configuration member AVZSIN00, you can create a rule to intercept the change and perform additional processing.

Temporary variables

Temporary variables, which begin with the stem value GLVEVENT, exist only during an event and are deleted when the event is over. Temporary variables are used by high-level language (HLL) routines

that create and interrogate these types of variables. To create or access a temporary variable, use the SDBVALUE API function. A rule can reference a temporary variable by name.

Event variables

When an event occurs, event variables are created. These variables pass information about the event to the rules for the event. For example, ATH.AUPWDBSS is an event variable for the LOGON event. The value of the ATH.AUPWDBSS variable is the Db2 subsystem name that the connection string provides. You can use this variable in a rule that restricts logons to a specific Db2 subsystem.

Most event variables are read-only; however, some can be modified. Changes to modifiable event variables are cumulative. The first rule that runs uses the original value of the variable. Each rule that later runs uses the value that the previous rule modified. Even if a rule modifies the value of a variable, all rules that are eligible to run still run.

Events

Data Virtualization Manager server allows you to create rules for event types.

You can create rules for the following types of events:

- Authorization (ATH) events that occur when the Data Virtualization Manager configuration performs authorization processing for a controlled resource.
- Command (CMD) events that occur when the Data Virtualization Manager configuration receives a command from a z/OS® console.
- Exception (EXC) events that occur when tasks exceed limits or fail. These events are generated only when the SEFGLVENTS parameter is set to allow them.
- Global variable events (GLV) that occur when the value of a global variable is changed.
- Publish (PUB) events occur in response to INSERT, UPDATE, and DELETE.
- RPC events occur in response to remote procedure calls.
- SQL events occur before a SQL statement is run.
- Time-of-day (TOD) events occur at specific times.
- Virtual tables (VTB) rules allow you to have a single virtual table that can use to represent many data sets of the same structure.

For each event, you can create one or more rules. Within each rule, you specify an action to take in response to the event. For example, you might create two rules for the LOGON event. In one rule, you specify that if an ID attempts to log on more than three times within a 24-hour period, subsequent logon requests are rejected. In another rule, you might specify that all logs on attempts from a specific ID are rejected.

Authorization (ATH) events

This section describes the types of authorization (ATH) events.

All authorization events

This event occurs when an authorization request is made. A rule for this event can reject, accept, or modify the request.

Return values

When an ATH event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

Values for these variables are set only when an ATH rule processes an ATH event.

Criterion	Variable name	Contents	Data type
ALL (all variables)	ATH.OPAU13WA	The WAITS flag is on if the wait state is allowed and is off if wait state is not allowed. If the wait state is not allowed, actions that cause the task to enter a wait state are not allowed.	Character, read only
ALL	ATH.OPAUACSR	The type of access that is being requested. The following are valid values for the access type, except for LOGON requests: <ul style="list-style-type: none"> • ADD • CONTROL • DISPLAY • DEFINE • EXECUTE • INFO • LIST • KILL • MODIFY • READ • SHOW • SET • WRITE 	Character, read only

Criterion	Variable name	Contents	Data type
ALL	ATH.OPAUERMG	A REXX program can specify the error message to send to the client.	Character, read-write
ALL	ATH.OPAURQRC	The request return code. The following are valid values: <ul style="list-style-type: none"> • 00: Request allowed • 04: Request must be modified • 08: Request failed • 12: Request abended • 16: Product address space is unavailable 	Character, read-only
ALL	ATH.OPAURQSR	The type of request that is being processed. The following are valid values: <ul style="list-style-type: none"> • CICSCONNECTIONS: CICS® connections • CONTROLBLOCKS: Product control blocks • DATABASES: Product databases • DATAMAP: Data map definitions • FILE: Shared server QSAM/BPAM data sets • GLOBALS: Global variables • LINKS: Communication links • LOGON: Password and user validation • PARMS: Product parameters • RPC: Remote procedure call • AVZ: AVZ command • SEF: Event Facility commands • TRACEDATA: Detailed Trace Browse data • TRACEBROWSE: Trace browse • TSO: Time Share Option • USERS: Remote users 	

Criterion	Variable name	Contents	Data type
ALL	ATH.OPAUSRID	The search ID, which is created by combining the request type with the access type, for example: <ul style="list-style-type: none"> • PARMS.SHOW displays a product parameter • SEF.INFO obtains SEF information. 	
ALL	ATH.OPAUUSID	The user ID that is being validated (LOGON), the user ID being logged off (LOGOFF), or the user ID for the task that is requesting access to the controlled resource. Note: A rule for the LOGON event can change the value of the user ID so that the rule-generated user ID can be used for subsequent validation by the security product. Rules for other authorization events should not attempt to alter the ATH.OPAUUSID variable.	Character, read-only, except as noted
ALL	ATH.USER	The user area is passed to all rules that run in response to the same event.	Read-only

Control block events

This event occurs when a control block is accessed or updated. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an ATH event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.

Return value	Description
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

CONTROLBLOCK variables are used for events that pertain to accessing or updating a product control block.

Variable name	Contents	Data type
ATH.AUBKCBAD	The address of the control block.	Character, read-only
ATH.AUBKCBAS	The address space (ASID) of the control block.	Numeric, read-only
ATH.AUBKCBLN	The length of the control block.	Numeric, read-only
ATH.AUBKCBNA	The name of the control block.	Character, read-only

Database events

This event occurs when a database is defined, accessed, or updated. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When a database event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

DATABASE variables are used for events that pertain to defining, accessing, or updating a product database.

Variable name	Contents	Data type
ATH.AUDBHOST	The host name of the database.	Numeric, read-only
ATH.AUDBNAME	The name of the database.	Character, read-only
ATH.AUDBTYPE	The type of the database.	Character, read-only

Global variable events

This event occurs when a global variable is defined, accessed, or updated. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an ATH event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determination to allow or deny access to the requested resource.

Variables

The following variables are available.

Variable name	Contents	Data type
ATH.AUGLDELN	The length of the name of the global variable.	Numeric, read-only
ATH.AUGLDENA	The name of the global variable.	Character, read-only

Variable name	Contents	Data type
ATH.AUGLOPCH	<p>The operation. The following are valid values:</p> <ul style="list-style-type: none"> • A: Add a global variable. • D: Drop a global variable. • E: Check for the existence of a global variable. • F: Check for the existence of a global variable and obtain (return) the value. • I: Obtain information about a global variable. • L: List information about a global variable. • O: Obtain a global variable. • R: Remove a global variable. • S: Subtree processing. • T: Subtree information processing • U: Update a global variable. • V: Value processing. 	Character, read-only
ATH.AUGLRQTY	<p>The type of the access request. The following are valid values:</p> <ul style="list-style-type: none"> • A: READ access • U: UPDATE access 	Character, read-only

IMSLTERM events

This event occurs when the IMSLTERM (IMS logical terminal) authorization event occurs. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an IMSLTERM event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.

Return value	Description
Other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

The following variable is available. The IMSLTERM variable is used for events that pertain to IMSLTERM.

Descriptive name	Variable name	Contents	Data type
Virtual table name	ATH.AULTNAME	The name of the virtual table.	Character, read-only

Communication link events

This event occurs when a communication link is defined, accessed, or updated. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an communication link event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUSERMG variable, which for most authorization requests, returns an error message to the requestor.
Other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

LINKS variables are used for events that pertain to defining, accessing, or updating a communication link.

Variable name	Contents	Data type
ATH.AULIHOST	The host name for the link. This name might be truncated. To avoid the additional processing that is required to resolve the host name, the server does not usually obtain or provide the client host name.	Character, read-only

Variable name	Contents	Data type
ATH.AULIIPAD	The TCP/IP address in 4-byte binary form.	Binary, read-only
ATH.AULILU	The LU 6.2 name that is set only for LU 6.2 links.	Character, read-only
ATH.AULIMODE	The LU 6.2 mode name that is set only for LU 6.2 links.	Character, read-only
ATH.AULITYPE	The link type. The following are valid values: <ul style="list-style-type: none"> • 6: LU 6.2 link • T: IBM TCP/IP link • I Interlink TCP/IP 	Character, read-only

Log off events

This event occurs after the client session to the host is terminated. Therefore, no response data can be sent to the client.

A rule for this event can provide the following responses:

- Write messages to a console or to the Trace Browse. The error message variable (ATH.OPAUERMG) can also be set. This value of this variable displays in the Trace Browse if ATH messages are being traced.
- Write SMF records. The SDBINFO function can be used in addition to the ATH event variables passed to this routine.
- Access and update other resources. For example, a global variable can be modified to show that the current user is no longer connected.

Return values

When an log-off event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

Variables

LOGOFF variables are used for events that pertain to writing messages to a console or Trace Browse, writing SMF records, or accessing and updating other resources.

Descriptive name	Variable name	Contents	Data type
Termination code	ATH.AULGABCD	The termination code, which is a 4-byte hexadecimal string. The value is 0000 if the current thread terminated normally.	Character, read-only
Authorization scheme	ATH.AULGAUSC	The authorization scheme. The following are valid values: <ul style="list-style-type: none"> • SDBECURE: The user ID was created by using the SDBECURE API. • RA-PROXY: A RUNAUTH (proxy) user ID log off. • BASIC: An HTTP authorization, request header scheme. 	Character
Cache	ATH.AULGCAUS	The user ID cache flag. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): The user ID is logged off. • 1: If the user ID was previously cached and is retained in the cache. 	Character, read-write
Connection token	ATH.AULGCNTK	The connection token is an 8-byte hexadecimal string. To identify the terminating task, this value can be passed to the SDBINFO function. This value is only required for test (TSO) versions of the main product address space.	Character, read-only
CPU time	ATH.AULGCPTM	The CPU time that is used by the current task, which is specified in seconds and fractions of a second.	Character, read-only
Elapsed time	ATH.AULGELTM	The elapsed time of the current task, which is specified in seconds and fractions of a second.	Character, read-only
GMT logon time	ATH.AULGLGGM	The GMT logon time, which is provided as a timestamp. The format is YYYY/MM/DD-HH:MM:SS.NNNNNN..	Character, read-only
Local logon time	ATH.AULGLGTM	The local logon time, which is provided as a timestamp. The format is YYYY/MM/DD-HH:MM:SS.NNNNNN..	Character, read-only

Descriptive name	Variable name	Contents	Data type
Uncompressed bytes	ATH.AULGWRT0	The total number of uncompressed bytes. It is provided by using the next field.	Character, read-only
Wait	ATH.APAU13WA	<p>The WAITS flag. The following are valid values:</p> <ul style="list-style-type: none"> • 0 (zero): WAITS are not allowed. • 1: WAITS are allowed. <p>If WAITS are not allowed, I/O and other services that might cause the task to enter a wait state are not allowed. Some logoff operations occur during end-of-task processing when it is important to monitor the wait-allotted flag to prevent unwanted subtask terminations.</p>	

Log on events

This event occurs when a logon occurs.

A rule for this event can provide the following responses:

- Set or reset all of the execution limits for the current client user ID. The default values are passed to the rule. If the default values are not changed, they are used.
- Set the return value to REJECT, and use the ATH.OPAUERMG variable to send an error message.
- Set the return value to ACCEPT. Using this return value bypasses the password validation that the security product does. Use ACCEPT only if you do not have a security product that is installed and rely on
- Modify the user ID before the security product processes it.

Return values

When an ATH event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.

Return value	Description
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

LOGON variables are used for events that pertain to setting or resetting execution limits for the current client user ID, rejecting the current logon attempt, bypassing password validation, or modifying a user ID before it is processed by RACF/ACF2.

Descriptive name	Variable name	Contents	Data type
Security optimization	ATH.AUPWAEAC	The Security Optimization flag. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): Security optimization is not active. • 1: Security optimization is active. 	Character, read-only
Security optimization cache	ATH.AUPWAERT	The amount of time, in seconds, that the security optimization cache entry is retained for the user.	Character, read-only
Application name	ATH.AUPWAPNA	The name of the application. This value is optionally set by the ODBC application.	Character, read-write
Authentication scheme	ATH.AUPWAUSC	The authentication scheme for the logon. The following are valid values: <ul style="list-style-type: none"> • SDBECURE: A logon by using the SDBECURE API • RA-PROXY: A RUNAUTH (proxy) user ID logon • BASIC: An HTTP authorization, header user ID logon 	Character, read-write
User ID cache	ATH.AUPWCAUS	A user ID cache flag. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): Suppresses caching for the user ID • 1: If the client user ID/acee (access control element entry) is or could be cached for reuse. 	Character, read-only
ODBC connection string	ATH.AUPWCNSR	The ODBC connection string from the client.	Character, read-write
Base CPU time interval	ATH.AUPWCPBA	The base CPU time interval for time slicing.	Character, read-write

Descriptive name	Variable name	Contents	Data type
Error CPU time limit	ATH.AUPWCPPER	The error CPU time limit that is checked by the check limits task.	Character, read-write
Failure CPU time limit	ATH.AUPWCPPFA	The failure CPU time limit that is checked by the check limits task.	Character, read-write
Execution time interval	ATH.AUPWCPIPIN	The execution time interval for time slicing.	Character, read-write
CPU time limit	ATH.AUPWCPTM	The CPU time limit that is checked by the ODBC task.	Character, read-write
Plan name	ATH.AUPWDBPN	The plan name. This value is provided in the connection string.	Character, read-write
Db2 subsystem name	ATH.AUPWDBSS	The Db2 subsystem name. This value is provided in the connection string.	Character, read-write
Database user ID	ATH.AUPWDBUS	The database user ID that is used to connect to Db2. When you use CAF, you can switch the user ID, but you cannot switch the user ID with RRSF unless you are using Enterprise Auditing.	Character, read-write
Task priority	ATH.AUPWDPPR	The z/OS task dispatch priority of the current task, which is a value 0 - 225.	Character, read-write
Enterprise auditing	ATH.AUPWENTL	The enterprise auditing flag. If this flag is set to 1, enterprise auditing requests from the client are accepted. If the flag is set to any other value, requests are ignored.	Character, read-write
Exclusive lock	ATH.AUPWEXFA	The exclusive lock time limit, which is checked by the check limits task.	Character, read-write
Application internal name	ATH.AUPWINNA	The application internal name, if available. This value, which is available only for non-console-mode Windows 32-bit applications, is obtained from the Windows version resources.	Character, read-only

Descriptive name	Variable name	Contents	Data type
New plain-text password	ATH.AUPWLGNW	A new plain-text password, which the application provides. The PROVIDEPASSWORDS parameter controls this variable. If the PROVIDEPASSWORDS is set to YES, the variable is set to a non-blank string. Otherwise, the variable is set to blank characters. The password can only be changed if the PROVIDEPASSWORDS parameter is set to CHANGE.	Character, read-write
Plain-text password	ATH.AUPWLGPW	The plain-text password, which the application provides. The PROVIDEPASSWORDS parameter controls this variable. If the PROVIDEPASSWORDS is set to YES, the variable is set to a non-blank string. Otherwise, the variable is set to blank characters. The password can only be changed if the PROVIDEPASSWORDS parameter is set to CHANGE.	Character, read-write
Network user ID	ATH.AUPWLNID	The network user ID from the client.	Character, read-write
Application module name	ATH.AUPWMDNA	The application module name, if available. This is the name of the application that is using the .NET client.	Character, read-only
Maximum rows generated	ATH.AUPWMXCA	The maximum number of rows that a call RPC can generate before an error is reported to the RPC.	Character, read-write
Maximum rows fetched	ATH.AUPWMXRW	The maximum number of rows that can be fetched before SQL code +100 is simulated.	Character, read-write

Descriptive name	Variable name	Contents	Data type
Maximum timerons	ATH.AUPWMXTM	The maximum timerons limit, which is checked by the client task. A timeron is a unit of measurement used to give a rough relative estimate of the resources, or cost, required by the database server to execute two plans for the same query. The resources calculated in the estimate include weighted CPU and I/O costs.	Character, read-write
Single logon	ATH.AUPWNTLG	The single logon flag from the client. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): The client did not use a single logon. • 1: The client used a single logon. 	Character, read-only
RPC enqueue limit	ATH.AUPWRPEH	The RPC enqueue time limit that the check limits task checks.	Character, read-write
RPC execution limit	ATH.AUPWRPEL	The RPC execution time limit.	Character, read-write
Share lock limit	ATH.AUPWSHFA	The share lock time limit that the check limits task checks.	Character, read-write
Per SQL CPU limit	ATH.AUPWSQFA	The per SQL CPU time limit that the check limits task checks.	Character, read-write
Update lock limit	ATH.AUPWUPFA	The update lock time limit that the check limit task checks.	Character, read-write
User parameter	ATH.AUPWUSPA	The User parameter from the client.	Character, read-write
PassTicket authentication	ATH.AUPWSPT	The PassTicket flag. The following are valid: <ul style="list-style-type: none"> • 0 (zero): The user is not using a PassTicket for authentication. • 1: The user is using a PassTicket for authentication. 	Character, read-write
Error wait time	ATH.AUPWWAER	The error wait time limit that the check limits task checks.	Character, read-write

Descriptive name	Variable name	Contents	Data type
Failure wait time	ATH.AUPWWAFA	The failure wait time limit that is checked by the check limits task.	Character, read-write
Warning wait time	ATH.AUPWWAWN	The warning wait time limit that is checked by the check limits task.	Character, read-write
WAITS flag	ATH.OPAU13WA	The WAITS flag. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): WAITS are not allowed • 1: WAITS are allowed If WAITS are not allowed, I/O and other services that might cause the task to enter a wait state are not allowed.	Character, read-write
Accept type string	ATH.OPAUACSR	The accept type string.	Character, read-only
Error message	ATH.OPAUERMG	The error message.	Character, read-only
Request type string	ATH.OPAURQSR	The request type string.	Character, read-only
Rule-invocation match string	ATH.OPAUSRID	The rule-invocation match string.	Character, read-only
Client user ID	ATH.OPAUUSID	The client user ID being logged on to the system.	Character, read-only

MQ events

This event occurs when an IBM MQ resource is defined. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an MQ event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

The MQSERIES variable is used for authorization of events that pertain to defining an MQ resource.

Descriptive name	Variable name	Contents	Data type
Queue manager	ATH.AUMQQMGR	The name of the queue manager. This name is set only for actions that are specific to one queue manager. This field is not set when the list of queue managers is being requested by a caller.	Character, read-only

Parameter events

This event occurs when a parameter is updated or accessed. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When a parameter event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

The PARMS variable is used for authorization of events that pertain to accessing or updating a product parameter.

Descriptive name	Variable name	Contents	Data type
Product parameter name	ATH.AUPAPANA	The product parameter name.	Character, read-only

RPC events

This event occurs when an attempt is made to run an RPC. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an RPC event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

RPC variables are used for authorization of events that pertain to execution of an RPC.

Descriptive name	Variable name	Contents	Data type
RPC module name	ATH.AURPNASR	The RPC module name that was extracted from the SQL call statement.	Character, read-only
SQL code REXX variable	ATH.AURPSQCD	The SQL code REXX variable. If RPC execution is rejected, the variable is set to a negative value.	Character, read-only

AVZ events

This event occurs when an attempt is made to run the AVZ command. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an AVZ event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

AVZ variables are used for authorization of events that pertain to execution of an AVZ command.

Descriptive name	Variable name	Contents	Data type
Options string	ATH.AUSDOTSR	The AVZ command Options string, such as 5.2.	Character, read-only
Subsystem name	ATH.AUSDSSNA	The subsystem name.	Character, read-only

SEF events

This event occurs when an attempt is made to run the SEF (event facility) command runs. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When an SEF event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

SEF variables are used for authorization of events that pertain to the running of an SEF command.

Descriptive name	Variable name	Contents	Data type
Subcommand for the SEF ARCHIVE verb	ATH.AUSEARSB	The subcommand for the SEF ARCHIVE verb.	Character, read-only
Current operation	ATH.AUSEAUOP	A flag that shows if the current operation affects the event procedure rule set. The following are valid values: <ul style="list-style-type: none"> • 0 (zero): • 1: 	Character, read-only
Rule set name	ATH.AUSEAURS	The ATH rule set name.	Character, read-only
Command buffer length	ATH. AUSEBULN	The SEF command buffer length.	Character, read-only
Command buffer	ATH.AUSECMBU	The SEF command buffer.	Character, read-only
z/OS dsname	ATH.AUSEDNA	The SEF rule set z/OS data set name (dsname for file management commands).	Character, read-only

Descriptive name	Variable name	Contents	Data type
Event procedure name	ATH.AUSERLNA	The SEF command event procedure name (member name for file management commands).	Character, read-only
Command request	ATH.AUSERQTY	<p>The SEF command request type. The following values are valid for rule set commands:</p> <ul style="list-style-type: none"> • A: Set auto-enable flags • B: Set auto-enable flags and enable them • C: Reset auto-enable flags and disable them • D: Disable rules • E: Enable rules • F: Refresh rules • I: Set dsname index (dsname with STAR) • L: List rule set or rule • R: Archive comand • S: Set or resent subsystem string • T: Test timer rules or another test • U: Show rule • X: Transfer data • Y: Set or reset SYSID string • Z: Reset auto-enable flag 	Character, read-only
		<p>The following values are valid for file-management commands:</p> <ul style="list-style-type: none"> • 3: Open a data set • 4: Close a data set • 5: Refresh a data set • 6: File list • 7: Quiesce a data set • 8: Allocate a data set • 9: Deallocate a ddname 	

Descriptive name	Variable name	Contents	Data type
		<p>The following values are valid for TSO server management commands:</p> <ul style="list-style-type: none"> • F: TSOSRV_LIST • K: TSOSRV_QUEUES • M: TSOSRV_STOP • O: TSOSRV_RESETO • P: TSOSRV_FREE • Q: TSOSRV_EXECSTATUS 	
SEF rule set name	ATH.AUSERSNA	The SEF command rule set name (ddname for file-management commands).	Character, read-only
SEF command verb string	ATH.AUSEVBSR	The SEF command verb string.	Character, read-only

Token events

This event occurs when a token is accessed. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When a token event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

TOKENS variables are used for authorization of events that pertain to the access of an execution token.

Descriptive name	Variable name	Contents	Data type
Host name	ATH.AUTKHONA	The host name field, which contains the host name of the client that created the current token. This field is not set for multiple token fetch requests.	Character, read-only
ID string	ATH.AUTKIDSR	The token ID string, which contains the token ID that is being accessed or deleted. This field is not set for multiple token fetch requests.	Character, read-only
User data	ATH.AUTKUSDA	The user data field, which contains the user data of the token that is being accessed or deleted. This field is not set for multiple token fetch requests.	Character, read-only
User ID	ATH.AUTKUSID	The user ID field, which contains the user ID of the client that created the current token. This field is not set for multiple token fetch requests.	Character, read-only

TSO events

This event occurs when a TSO command runs. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When a TSO event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.

Return value	Description
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

TSO variables are used for authorization of events that pertain to execution of a TSO command.

Descriptive name	Variable name	Contents	Data type
Buffer length	ATH.AUOSBULN	The TSO command buffer length.	Character, read-only
Buffer	ATH.AUOSCMBU	The TSO command buffer.	Character, read-only
Command verb string	ATH.AUOSVBSR	The TSO command verb string.	Character, read-only

User events

This event occurs when information about a remote user is accessed, when a remote user connection is terminated, and when a cancel Db2 thread operation occurs. A rule for this event can accept or reject the request or allow the security product to determine if the request is allowed.

Return values

When a user event ends, the rule sets a return value. The server evaluates the return value and invokes z/OS security routines.

Return value	Description
ACCEPT	Access to the requested resource is allowed, and additional processing by the z/OS security subsystem is not performed.
REJECT	Access to the requested resource is denied, and additional processing by the z/OS security subsystem is not performed. The rule can include the ATH.OPAUERMG variable, which for most authorization requests, returns an error message to the requestor.
Any other value	If another value or no value is returned, the z/OS security subsystem performs validation checking. The security product makes the final determine to allow or deny access to the requested resource.

USERS variables are used for authorization of events that pertain to accessing or killing connections of a remote user.

Descriptive name	Variable name	Contents	Data type
Connection ID	ATH.AUUSCNID	The connection ID, which is set only for stop or cancel operations.	Character, read-only
User name	ATH.AUUSKILL	The name of the user to stop or cancel.	Character, read-only

Descriptive name	Variable name	Contents	Data type
Connection type	ATH.AUUSTYPE	<p>The connection type. The following are valid values:</p> <p>AMDETRT: If a user is requesting information about a specific APPC/MVS conversation information for each task with an active conversation.</p> <p>AMINTSUM: If a user is requesting information about the APPC/MVS summary.</p> <p>DETAIL: If a user is requesting information about user or interval detail data stored in the main product address space.</p> <p>IDDETRT: If a user is requesting information about specific APPC/IDMS conversation information for each task with an active conversation.</p> <p>REMOTE: If a user request information about all remote connections in the main product address space.</p> <p>REMOTEGRP: If a user is requesting information about TCP/IP host name and port information.</p> <p>RRRMINFO: If a user is requesting information about Resource Recovery Services.</p> <p>SECOPT: If a user is requesting information about security optimization cache entries.</p> <p>SUMMARY: If a user is requesting information about all of the summary interval data stored in the main product address space.</p> <p>TASKS: If a user is requesting information about all tasks that run in the main product address space.</p>	Character, read-only

Descriptive name	Variable name	Contents	Data type
		<p>REMOTE: If a user request information about all remote connections in the main product address space.</p> <p>REMOTEGRP: If a user is requesting information about TCP/IP host name and port information.</p> <p>RRRMINFO: If a user is requesting information about Resource Recovery Services.</p> <p>SECOPT: If a user is requesting information about security optimization cache entries.</p> <p>SUMMARY: If a user is requesting information about all of the summary interval data stored in the main product address space.</p> <p>TASKS: If a user is requesting information about all tasks that run in the main product address space.</p>	

Command (CMD) events

Command events control client/server access to the mainframe.

When the Data Virtualization Manager server receives a command from a z/OS console, a rule is scheduled to run. The console can be a physical console or extended software, such as System Display and Search Facility (SDSF) or CA OPS/MVS Event Management and Automation. The command consists of a command verb, followed by optional operands. The command verb string is matched against enabled CMD rules to find the rule to run.

CMD rules perform the following tasks:

- Examine the command, parse the operands, and perform necessary actions, such as read and set product parameters. This allows parameters to be displayed and changed from the z/OS console.
- Access and update REXX global variables.
- Use REXX SAY statement to communicate with the console that entered the command. All output from the SAY statement is routed to the console that entered the original command. This allows ASO products to communicate with, interrogate the status, and control the Data Virtualization Manager server.

Note: Because CMD rules can access and update any part of the product, you must control who can create, enable, and disable CMD rules.

All CMD rule processing is done by IBM Data Virtualization Manager for z/OS/REXX. Processing in another programming language is not supported.

Syntax

To trigger a CMD rule, use the z/OS STOP or MODIFY command, or use a z/OS command that specifies the subsystem name. The following commands are valid:

- `MODIFY xVZy, command text`

- *xVZy command text*
- *xVZy, command text*

where *xVZy* is a specific instance of the Data Virtualization Manager server, which is identified by the subsystem name that was assigned during installation.

When the z/OS STOP command triggers a CMD rule, the rule can control or reject product shutdown. The criterion of the rule must be STOP or a less specific criterion that matches the STOP command. The z/OS STOP (P) command can also trigger a CMD rule that has the matching criterion of STOP.

Header statement

A CMD criterion is a string of 1 - 30 characters. To schedule the rule to run for all commands, use a single * (asterisk) as the criterion. Use a trailing * (asterisk) as a wildcard character.

Use the following format for the header statement:

```
/*CMD criterion
```

Process section

A REXX process section is required.

Return values

The following table lists the return values for CMD rules:

Return value	Action
None supplied	If the rule runs a RETURN command, the Data Virtualization Manager server sends a return code that indicates the successful completion of the rule.
ACCEPT	The command in the rule was successfully completed.
REJECT	The command in the rule was rejected. To specify why the command was rejected, you REXX SAY statements.

The return value for a STOP CMD rule determines how the Data Virtualization Manager server terminates. The following return values are valid:

Return value	Action
None supplied	Termination is allowed to continue.
ACCEPT	Termination is not allowed to continue.
REJECT	Termination is not allowed to continue.

CMD event variables

Values for these variables are set only when a CMD rule processes a CMD event.

Variable	Contents	Data type
CMD.TEXT	Operands that are entered after the command name at the console.	Character, read-only

Variable	Contents	Data type
CMD.VERB	The command name that is entered at the console.	Character, read-only

Exception (EXC) events

An exception event occurs when a task exceeds a specified limit.

The EXC procedure samples that are distributed with the server contain a sample for each of the exception types. Instructions in the samples explain the following information:

- The environment in which the exception is detected.
- The operational controls that affect subsequent processing by the server.
- The valid return values.

The header statement for an EXC rule is */*EXC criterion*, where *criterion* is one of strings in the following table. A process section is required.

Criterion	Description	Default action
CPULIMIT	<p>A transaction task exceeded its maximum CPU time limit. This exception is detected only when multipart messages are being transmitted and only when a new message segment is being read. A rule for this event can take one or more of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value REJECT to terminate the ODBC connection, and use the EXC.OPERXRMG variable to send an error message to the client.</p> <p>The rule can use the SDBINFO API function and pass or not pass the connection token as the second parameter.</p>	Terminate the transaction task.

Criterion	Description	Default action
CPUTIME	<p>A transaction task exceeded its maximum CPU time limit. This exception can be detected any time while the task is running. A rule for this event can take one or more of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value KILL to terminate the ODBC connection. No message is sent to the client.</p> <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	Terminate the transaction task.
IMSFAIL	<p>An IMS task detected a failing IMS operation. This exception can occur for any type of IMS processing. The rule can use the SDBINFO function without passing the connection token as the second parameter.</p>	Terminate the IMS operation, and reflect the error to the client task.
IVPD	<p>SQL execution has encountered an invalid decimal value. The rule decides to either return a correct value for the decimal field or allow the SQL statement to fail. This rule can be set to the following:</p> <ul style="list-style-type: none"> • ALLOW: IVPD rule will be allowed to execute. • REJECT: IVPD rule will be rejected and the following error message gets thrown- "invalid decimal data encountered." 	If no rule is enabled to handle the exception or no valid decimal return value is specified, the following error message is thrown: "invalid decimal data encountered"

Criterion	Description	Default action
LOCKEXCLUSIVE	<p>A transaction task exceeded its Db2 exclusive lock limit. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value KILL to terminate the ODBC connection. No message is sent to the client.</p> <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	Terminate the transaction task.
LOCKSHARE	<p>A transaction task exceeded its Db2 share lock limit. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value KILL to terminate the ODBC connection. No message is sent to the client.</p> <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	Terminate the transaction task.

Criterion	Description	Default action
LOCKUPDATE	<p>A transaction task exceeded its Db2 update lock limit. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value KILL to terminate the ODBC connection. No message is sent to the client.</p> <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	
LOGFAILURE	<p>A Db2 database exceeded a pending logging requests limit. This exception can be detected at any time. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. This action preserves the contents of the pending request queue and prevents error messages from being issued. • Use the return value CLEAR to clear the pending request queue, release all associated storage, and send an error message that contains the number of cleared requests to the system console. <p>Modify the limit so that the exception does not occur again.</p>	

Criterion	Description	Default action
PERSQLCPU	<p>A transaction task exceeded its per-SQL-statement CPU time limit. This exception is detected only by SQL operations that the server runs, for example for / *EXESQL rules. It is not detected when a user-written high-level language (HLL) program invokes long-running SQL operations. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread so that the exception does not occur again. • Use the return value KILL to terminate the ODBC connection. • Use the return value IGNORE to ignore the exception. • Modify the limit for the current thread. This action prevents the exception from occurring again. <p>Use the return value KILL to terminate the ODBC connection. No message is sent to the client.</p> <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	Terminates the transaction.
PGMDURATION	<p>An RPC stalled or was put it into an indefinitely long wait state. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Examine the problematic program name and return no value, in which case the default action is taken. • Return the value IGNORE, which allows the problematic task and the RPC task to continue. <p>Use the EXC.EXXDTMLM variable to modify the limit.</p>	If no rule is enabled to handle the exception or if no return value is specified, the default action is to cancel the problematic task and clear the RCP program.

Criterion	Description	Default action
RPCENQUEUE	<p>A transaction task detected that a client task exceeded its RPC enqueue time limit. This exception can be detected at any time. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Return the value IGNORE to ignore the exception. • Modify the time limit for the current thread. • Return the value KILL to terminate the ODBC connection. <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	
RTMONITOR	The application exceeded the client response time. This exception is detected only for ODBC connections.	None
SESSIONFAILURE	<p>A transaction task detected that a client task exceeded the session failure limit. This exception can be detected at any time. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Return the value IGNORE to ignore the exception. • Modify the time limit for the current thread. • Return the value KILL to terminate the ODBC connection. No message is sent to the client. 	Terminate the ODBC client task.
SQLFAIL	A transaction task detected that a SQL statement failed. When a failure occurs, a negative SQL code is set. Only SQL operations that the server runs, such as for /*EXECSQL rules, detect this exception. The exception is not detected when a user-written high-level language (HLL) program invokes a long-running SQL operation.	Returns the SQL error code to the transaction task.

Criterion	Description	Default action
TIMERONLIMIT	<p>A transaction task detected that a prepare returned a timer-on value that exceeds the limit. Only SQL operations that the server runs, such as for /*EXEC SQL rules, detect this exception. The exception is not detected when a user-written high-level language (HLL) program invokes a prepare. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Return the value ALLOW, which allows the exception. • Modify the limit. • Return the value REJECT, which terminates the SQL statement, and use the EXC.OPERMG variable to return an error message to the client. <p>The rule can use the SDBINFO function without passing the connection token as the second parameter.</p>	
WAITTIME	<p>A transaction task exceeded the wait time limit. This exception can be detected at any time. A rule for this event can take one of the following actions:</p> <ul style="list-style-type: none"> • Return the value IGNORE to ignore the exception. • Modify the limit. • Return the value KILL to terminate the ODBC connection. No message is sent to the client. <p>The rule can use the SDBINFO API function and must pass the connection token as the second parameter. The connection token is required to identify the task that has the exception, rather than the current task.</p>	
ZSQLALLIMSSEGMENTS	<p>SQL Solution determined that a SQL statement causes all IMS segments that are specified as tables to be read because the child segments that are being joined are not constrained. The query does not specify the CHILD_ID and PARENT_ID columns in the WHERE clause.</p>	<p>Allow or terminate the SQL statement, which is based on the value of the SLENDFLTEXCACTION parameter.</p>

Criterion	Description	Default action
ZSQLFULLDBREAD	SQL Solution determined that a SQL statement causes all database source records to be read because the subtable query is not constrained. The query does not specify the CHILD_KEY and PARENT_KEY columns in the WHERE clause.	Allow or terminate the SQL statement, which is based on the value of the SQLENGDFLTEXCACTION parameter.
ZSQLINCKEYBEGINNING	SQL Solution determined that only the beginning of an incomplete key was specified for one of the tables in a query. This situation might occur when multiple columns comprise the key and the query that is specified only the beginning columns. This situation is acceptable for VSAM access, but it might incur additional overhead for IMS access.	Allow or terminate the SQL statement, which is based on the value of the SQLENGDFLTEXCACTION parameter.
ZSQLINCKEYPARTIAL	SQL Solution determined that only part of an incomplete key was specified for one of the tables in the query and that the beginning portion of the key was not specified.	Allow or terminate the SQL statement, which is based on the value of the SQLENGDFLTEXCACTION parameter.
ZSQLNOKEYCOLUMNS	SQL Solution determined that no key columns were specified in the WHERE clause. This situation causes the entire database to be read.	Allow or terminate the SQL statement, which is based on the value of the SQLENGDFLTEXCACTION parameter.
ZSQLNOWHERECLAUSE	SQL Solution determined that no WHERE clause was provided for a table. This situation causes the entire database to be read.	Allow or terminate the SQL statement, which is based on the value of the SQLENGDFLTEXCACTION parameter.

Variables for all EXC events

You can use the variables in the following table in any EXC rule:

Variable	Contents	Data type
EXC.OPEXACSR	<p>The action string for the current exception. This string cannot be directly changed; however, the return value from some rules can change the action string. The following are valid values:</p> <ul style="list-style-type: none"> • ACCEPT: Accept the current condition • IGNORE: Ignore the current condition • KILL: Kill the current client connection • ALLOW: Allow the current exception • NOACTION: Take no action • REJECT: Reject the current exception • TERMINATE: Terminate the current client connection 	Character, read-only
EXC.OPEXCNTK	<p>The connection token that is used to obtain information about the thread where the exception occurred. You must use this field for all exceptions that the Check Limits task detects. The connection token is passed as the second parameter of the SDBINFO function. The connection token is only needed if the EXC.OPEXINFO flag is set to 0 (zero).</p>	Character, read-only
EXC.OPEXERMG	<p>The error message field. This field can be modified to send messages to the application.</p>	Character, read-write
EXC.OPEXINFO	<p>A variable that indicates whether the SDBINFO function can be used by the EXC rule. Valid values are:</p> <ul style="list-style-type: none"> • 0 (zero): SDBINFO cannot be used • 1: SDBINFO can be used 	Character, read-only
EXC.OPEXSRID	<p>The search ID field contains the criterion that triggers the current rule. The valid values are listed in the previous table.</p>	Character, read-only

Variable	Contents	Data type
EXC.OPEXWAOK	A variable that indicates whether the EXC rule is allowed to perform operations that cause the current subtask to be placed in a waiting state. An example of such a task is issuing an I/O request. Valid values are: <ul style="list-style-type: none"> • 0 (zero): WAITS are not allowed • 1: WAITS are allowed 	Character, read-only
EXC.USER	The user area is passed among all rules that are triggered for the same event.	Character, read-write

Variables for CPULIMIT events

Variable	Contents	Data type
EXC.EXCLSPLM	The CPU time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop checking the CPU time.	Character, read-write
EXC.EXCLCPVL	The CPU time value shows how much CPU time the task has used.	Character, read-only

Variables for IMSFAIL events

Variable	Contents	Data type
EXC.EXIMIMCD	The IMS code. This value is obtained from IMS.	Character, read-only

Variable for IVPD events

Variable	Contents	Data type
EXC.EXSQVDPD	This variable can be modified by the user. The user can provide a valid decimal value for this variable.	Hexadecimal
EXC.EXSQVTBN	This variable displays the virtual table's name in which the invalid decimal data is identified	Character
EXC.EXSQCLNM	This variable displays the column name where the invalid decimal data is encountered	Character
EXC.EXSQDTTY	This variable displays the data type of the field	Character

Variable	Contents	Data type
EXC.EXSQOFST	This variable displays the offset of the field in virtual table where this invalid data is identified	Integer
EXC.EXSQPRES	This variable displays the precision of the decimal field	Integer
EXC.EXSQSCAL	This variable displays the scale of the decimal field	Integer
EXC.EXSQDTLN	This variable displays the length of the decimal data	Integer
EXC.EXSQVDPD	This variable displays the invalid decimal data encountered	Hexadecimal

Variables for LOCKEXCLUSIVE events

Variable	Contents	Data type
EXC.EXXCTMLM	The exclusive lock time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop checking the CPU time.	Character, read-only
EXC.EXSHTMVL	The share lock time value shows long the current task has been holding a share lock.	Character, read-only

Variables for LOCKUPDATE events

Variable	Contents	Data type
EXC.EXUPTMLM	The update lock time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop checking the CPU time.	Character, read-only
EXC.EXUPTMVL	The update lock time value shows long the current task has been holding an update lock.	Character, read-only

Variables for LOGFAILURE events

Variable	Contents	Data type
EXC.EXLGPNLN	The pending request limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop checking the limit of all pending requests. There are two request limits: the warning limit and the failure limit. If the rule is triggered for a warning limit, only the warning limit can be changed. If the rule is triggered for a failure limit, only a failure limit can be changed.	Character, read-only
EXC.EXLGPNNL	The pending requests value shows the number of pending logging requests.	Character, read-only
EXC.EXLGSNA	The database name is the Db2 subsystem that has too many pending logging requests.	Character, read-only

Variables for PERSQLCPU events

Variable	Contents	Data type
EXC.EXPQCPLM	The per-SQL-statement CPU time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop all per-SQL-statement time checking.	Character, read-write
EXC.EXPQCPVL	The CPU time value shows the amount of CPU time that the current SQL statement used.	Character, read-only

Variables for PGMDURATION rules

Variable	Contents	Data type
EXC.EXXDTMLM	The program duration time limit, in seconds. If the PGMDURATION rule returns IGNORE, which allows the RPC program to continue, each time that the limit is checked later, an exception occurs. To avoid raising additional exceptions, change this variable to increase the program duration limit, or set the variable to 0 (zero) to prevent additional events from being recognized. If the rule puts a new limit into effect, the new limit applies only to the in-flight RPC program execution for which the current exception was raised. The new limit is not retained in memory.	Character, read-only
EXC.EXXDTMVL	The duration time value shows how long, in seconds, the RPC program has been running.	Character, read-only
EXC.EXXDPGNA	The 8-byte name of the RPC program load module that is being run. For SQL CALL statements, the full procedure name from the SQL statement is unavailable when this exception is recognized. If no RPC rule matches the SQL CALL procedure name, the value of this variable is the first 8 characters of the procedure name. If a matching RPC rule contains a PROGRAM section, the value of the variable is the 8-byte load module name from the PROGRAM section of the RPC rule. In this case, the 8 characters might not match the leading characters of the CALL statement procedure name.	Character, read-only

Variables for RPCENQUEUE rules

Variable	Contents	Data type
EXC.EXNQTMLM	The RPC enqueue time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop all RPC enqueue time checking.	Character, read-write

Variable	Contents	Data type
EXC.EXNQTML	The RPC enqueue time value, which shows how long the current task has been holding a PRC enqueue.	Character, read-only

Variables for RTMONITOR rules

Variable	Contents	Data type
EXC.EXCRTGRT	The client response time goal, which shows the acceptable response time.	Character, read-only
EXC.EXCRTMMI	The actual client response time for the transaction that produced the exception.	Character, read-only
EXC.EXCRTRTR	The total number of client response time records.	Character, read-only
EXC.EXCRSRTR	The sum of the total response time for all records.	Character, read-only
EXC.EXCRTMGR	The total number of client response time records that missed the response time goal.	Character, read-only
EXC.EXCRSMGR	The sum of the total response time for the records that missed the response time goal.	Character, read-only
EXC.EXCRIPAD	The IP address.	Character, read-only
EXC.EXCRUSID	The user ID.	Character, read-only
EXC.EXCRAPNM	The application name.	Character, read-only

Variables for SESSIONFAILURE rules

Variable	Contents	Data type
EXC.EXSETMLM	The session failure time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop all RPC enqueue time checking.	Character, read-write
EXC.EXSETMVL	The session failure time value, which shows how long the current task has been processing on behalf of a client.	Character, read-only

Variables for SQLFAIL rules

Variable	Contents	Data type
EXC.EXSQSQCA	The SQLCA is built by prepare and is provided as a single binary data area.	Character, read-only
EXC.EXSQSQCD	The SQL code that is obtained from the SQLCA.	Character, read-only
EXC.EXSQSQSR	The SQL statement that failed.	Character, read-only

Variables for TIMERONLIMIT rules

Variable	Contents	Data type
EXC.EXTMSQCA	The SQLCA is built by prepare and is provided as a single binary data area.	Character, read-only
EXC.EXTMSQSR	The SQL string that was prepared	Character, read-only
EXC.EXTMTMLM	The timer-on limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop all timer-on checking.	Character, read-only
EXC.EXTMTMVL	The timer-on value shows the timer-on value that is returned by prepare.	Character, read-only

Variables for WAITTIME rules

Variable	Contents	Data type
EXC.EXWATMLM	The wait time limit. This variable can be modified to prevent the exception from occurring again. Set the variable to 0 (zero) to stop all wait time checking.	Character, read-write
EXC.EXWATMVL	The wait time value, which shows how long the current task has been waiting for a request from a client.	Character, read-only

Global variable (GLV) events

If the SEFGLVEVENTS startup parameter is set to YES, a global variable event is generated when the value of a global variable is updated.

The default state of the SEFGLVEVENTS startup parameter is NO. To enable global variable events, set the parameter to YES. Then, if a REXX program updates a global variable, the SEF attempts to locate a rule that has the same name as the global variable. Enabling GLV events has an impact on the virtual storage that the subsystem uses and that rules are not triggered by high-level language (HLL) programs.

The following is the format for the header statement:

```
/*GLV criteria
```

where *criteria* is the name of the global variable.

A REXX process section is required. Return values are ignored. The global variable value is updated, regardless of the return value.

When a global variable event occurs, the system extracts information about the event and creates the following variables, which are instantiated when the rule is scheduled to run:

Variable	Contents	Data type
GLV.NAME	The 1- to 50-byte name of the global variable that triggered the rule.	Character, read-only
GLV.NEWVALUE	The value of the global variable, after it was changed. The standard REXX definitions apply to variables that were never previously referenced or were dropped.	Character, read-only
GLV.OLDVALUE	The value of the global variable, before it was changed. The standard REXX definitions apply to variables that were never previously referenced or were dropped.	Character, read-only
GLV.PROGRAM	The name of the program or rule that updated the global variable.	Character, read-only
GLV.TEXT	A text message that describes the event. The string, which is truncated at 100 bytes, includes the GLV.NAME, GLV.PROGRAM, GLV.OLDVALUE, and GLV.NEWNAME values.	Character, read-only
GLV.USER	An 8-byte field for communicating between rules that are triggered for a single event. Use this field to pass information between multiple rules. This field is initialized to binary zeroes.	Character, read-write

Remote procedure call (RPC) events

An RPC event occurs when the name of a remote procedure call matches the criterion of a rule.

For example, the following CALL statement could trigger an RPC rule:

```
CALL ALL ('LITERAL1', 'LITERAL2')
```

Use RPC rules for the following purposes:

Allow RPCs to be written in REXX

REXX RPCs are not as versatile as those written in C, PL/1, COBOL, and Assembler. However, REXX RPCs can write SMF records and access host information and return it to the application.

Reject the current RPC statement

To reject the current RPC statement, set the return value to REJECT. To send a message to the client, set the RPC.MESSAGE variable. If the current RPC statement is rejected, set the RPC.CODE variable to a negative value, which indicates failure.

Accept the current RPC statement

To accept the current RPC statement, set the return value to ACCEPT. To send a message to the client, set the RPC.MESSAGE variable. If the current RPC statement is accepted, set the RPC.CODE variable to a positive value, which indicates success.

When an RPC event occurs, the system extracts information about the event and creates the following rule variables. These variables are instantiated when the rule is scheduled for execution.

Criterion	Variable	Contents	Data type
ALL	RPC.CODE	Returns an RPC code to the client.	Character, read-write
ALL	RPC.MESSAGE	Returns messages to the client.	Character, read-write
ALL	RPC.SEARCHID	The stored procedure name, which is extracted from the current SQL string. For example, for the following SQL statement, the search ID is SAMPVSAM:CALL SAMPVSAM ()	Character, read-only
ALL	RPC.TEXT	The SQL source string that invoked the current stored procedure. For example, for the following SQL statement, the text is CALL SAMPVSAM ():CALL SAMPVSAM ()	Character, read-only
ALL	RPC.USER	The user area is passed among all rules that are triggered for the same event.	Character, read-write

SQL events

A SQL event occurs when a SQL statement is processed.

A SQL rule runs before the SQL source is prepared. If a SQL source is modified, it is prepared or passed to run immediately after the SQL rule runs. Use SQL rules for the following purposes:

Modify a SQL source

To modify a SQL source, add or modify a WHERE clause.

Reject a SQL statement

To reject a SQL statement, use the REJECT return value. You can also use the SQL.MESSAGE to send a message to the client. If the SQL statement is rejected, set the SQL.CODE variable to a negative value. Otherwise, the value -1 is used as the SQL code.

Accept a SQL statement

To accept a SQL statement, set the return value to ACCEPT. If the SQL statement is accepted, Db2 does not run it. Instead, the rule processes the statement. To send a warning or error message to the client, use the SQL.MESSAGE variable. For warnings, a positive value. For failures, use a negative

value. If the return code is ACCEPT and a non-zero value is set for the SQL.CODE variable, a message is sent to the client. If a message is not provided, a default message is constructed and sent.

When a SQL event occurs, the system extracts information about the event and creates the following variables. These variables are instantiated when the SQL rule is scheduled to run. You can write a SQL rule that accesses the following variables:

Criterion	Variable	Contents	Data type
ALL	SQL.CODE	The code to return to the client	Character, read-write
ALL	SQL.MESSAGE	The message to return to the client	Character, read-write
ALL	SQL.SEARCHID	The SQL verb that is extracted from the current SQL string	Character, read-only
ALL	SQL.TEXT	The actual SQL source	Character, read-only
ALL	SQL.USER	The user area that is passed among all rules	Character, read-write

Time-of-day (TOD) events

A time-of-day event occurs when the z/OS timer that is associated with a rule expires.

To specify the header statement, use the following syntax:

```
/*TOD todspec, interval, endspec, maxexecs
```

where:

- *todspec* is the date or time. You must specify either *todspec* or *interval*. Use one of the following formats to specify *todspec*:
 - *ddMMMyyyy*, where *dd* is a 2-digit integer (01 - 31) that represents the day of the month; *MMM* is a 3-character abbreviation for the month (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC); and *yyyy* is a 4-digit year.
 - *yymmday*, where *yy* is a 2-digit year; *mm* is a 2-digit month; and *day* is the full name of a day of the week, for example, SUNDAY or MONDAY.
 - *hh:mm:ss*, where *hh* is a 2-digit integer (00 - 23) for the hour; *mm* is a 2-digit integer (00 - 59) for the minute; and *ss* is a 2-digit integer (00 - 59) for the seconds after the minute. The *ss* value is optional.
- *interval* is the amount of time to wait before running the rule again. You must specify either *todspec* or *interval*. Use the following format to specify the *interval*:
 - *n units*, where *n* is an integer that represents the number of times to run the rule, and *units* is the time to wait before running the rule again. For *units*, specify one of the following: DAY, DAYS, WEEK, WEEKS, HOUR, HOURS, MINUTE, MINUTES, SECOND, SECONDS.
- *endspec* is the time or date after which the rule stops running. This parameter is optional.
- *maxexecs* is an integer that represents the maximum number of times to run the rule. This parameter is optional.

Note: If you omit any parameter, code a comma in its place.

The value that is returned from a TOD rule has no special meaning.

When a TOD event occurs, the system extracts information about the event and creates the following variables. These variables are instantiated when the rule is scheduled for execution.

Criterion	Variable	Contents	Data type
ALL	TOD.NEXTFIRE	A value that indicates the next time that the rule runs. The following are valid values: <ul style="list-style-type: none"> The date and time in <i>yyyy/mm/dd hh:mm:ss</i> format. NONE if the rule will not run again. 	Character, read-only
ALL	TOD.USER	An 8-byte field for passing information among multiple rules. This field is initialized to binary zeroes.	Character, read-write

Virtual table (VTB) events

Virtual table events are generated by the SQL Engine when a table name is found in an SQL statement. These events are only generated if the **SEFVTBEVENTS** startup parameter is set to allow them. The rules allow for creating virtual tables dynamically from a Data Mapping facility (DMF) model map and for modifying certain table values.

No keywords are defined for VTB event procedures. Only the SQL engine schedules execution of enabled VTB event procedures for each table name in an SQL statement. VTB event procedures allow you to modify information in the DMF map. VTB event procedures make it possible to access multiple data sets using one DMF map by creating alias maps using a model map. Each alias map can specify a different data set name. The model map must be a map that is created by using DMF.

Only the event procedure criterion value is allowed (and *must* be present).

To specify the header statement, use the following syntax:

```
/*VTB criterion
```

where:

- criterion* is the criterion value for VTB event procedures. This *criterion* is one of the two event types that are shown in the following table.

Each VTB event procedure has access to server-wide global variables.

In addition, VTB-specific variables are created before the VTB event procedure is invoked. The variables that are created differ depending on the criterion.

Criterion	Variable	Contents	Data type
Any criterion	VTB.USER	The user area is passed between all event procedures that fire for the same event.	Read-write

Criterion	Variable	Contents	Data type
Any criterion	VTB.OPTBSRID	The search id field contains the criterion used to fire the current event procedure. The format of the criterion is the string 'MODIFYTABLE.' followed by the table name found in the SQL statement.	Character Read-only
Any criterion	VTB.OPTBTBNA	The 1 to 128-character table name from the SQL statement.	Character Read-only
MODIFYTABLE. <i>tablename</i>	VTB.OPTBMTNA	Set the model table name. This is the 1 to 50-character name of a DMF map that will be used to create a virtual table with the alias name <i>tablename</i>	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBMRDI	Disable MapReduce. Set this value to 1 to disable map reduce. Setting this value to 0 has no effect. VTB.OPTBMRDI and VTB.OPTBMREN are mutually exclusive.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBMREN	Enable MapReduce. Set this value to 1 to enable map reduce. Setting this value to 0 has no effect. VTB.OPTBMREN and VTB.OPTBMRDI are mutually exclusive. Enabling MapReduce requires that the MapReduce feature is enabled.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBMRTC	Set the number of MapReduce threads to use.	Character, write

Criterion	Variable	Contents	Data type
MODIFYTABLE. <i>tablename</i>	VTB.OPTBFLAT	<p>Flatten this table. Set this value to 1 to flatten the table. All columns and occurrences are returned in a single table</p> <p>Setting this value to 0 has no effect.</p> <p>VTB.OPTBFLAT and VTB.OPTBSUBT are mutually exclusive.</p>	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBSUBT	<p>Create subtables. Set this value to 1 to create subtables Columns that are part of an occurs or occurs-depending-on are returned as separate tables.</p> <p>Setting this value to 0 has no effect.</p> <p>VTB.OPTBFLAT and VTB.OPTBSUBT are mutually exclusive.</p>	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLSQ	<p>Clear sequential data set map related fields. Set this value to 1 to clear the data set member name, pre-write exit name, and post read exit name.</p> <p>Setting this value to 0 has no effect.</p> <p>The fields are cleared before any other variables are processed.</p>	Character, write

Criterion	Variable	Contents	Data type
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLCI	<p>Clear VSAMCICS map related fields. Set this value to 1 to clear the pre-write exit name, post read exit name, CICS file control table entry names, CICS connection name, and CICS transaction name fields.</p> <p>Setting this value to 0 has no effect.</p> <p>The fields are cleared before any other variables are processed.</p> <p>Clearing those fields cause a VSAMCICS file to be processed as a native VSAM file.</p>	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLAD	<p>Clear Adabas map related fields. Set this value to 1 to clear the database ID, file number, and subsystem name fields.</p> <p>Setting this value to 0 has no effect.</p> <p>The fields are cleared before any other variables are processed.</p>	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLD2	<p>Clear Db2 map related fields. Set this value to 1 to clear the table name, subsystem map name, table creator name, plan name, and user ID fields.</p> <p>Setting this value to 0 has no effect.</p> <p>The fields are cleared before any other variables are processed.</p>	Character, write

Criterion	Variable	Contents	Data type
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLIM	Clear IMS DB map related fields. Set this value to 1 to clear the segment name, DBD name, and PSB name fields. Setting this value to 0 has no effect. The fields are cleared before any other variables are processed.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCLIV	Clear IMS view map related fields. Set this value to 1 to clear the segment name, DBD name, and PSB name fields. Setting this value to 0 has no effect. The fields are cleared before any other variables are processed.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBDSNA	Set the 1 to 44-character VSAM or sequential data set name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBMEMA	Set the 1 to 8-character sequential data set member name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBPRWR	Set the 1 to 8-character VSAM, VSAMCICS, or sequential data set pre-write exit name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBPSRD	Set the 1 to 8-character VSAM, VSAMCICS, or sequential data set post read exit name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBVSBF	Set the 1 to 8-character CICS file control table entry name for the base file.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCONN	Set the 1 to 4-character CICS connection name.	Character, write

Criterion	Variable	Contents	Data type
MODIFYTABLE. <i>tablename</i>	VTB.OPTBCITR	Set the 1 to 4-character CICS transaction name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBADBI	Set the Adabas database ID (DBID) number.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBAFNR	Set the Adabas file number.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBSUBS	Set the 1 to 4-character Adabas subsystem name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBD2TN	Set the 1 to 128-character Db2 table name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBD2SN	Set the 1 to 50-character Db2 subsystem map name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBD2TC	Set the 1 to 8-character Db2 table creator ID.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBD2PN	Set the 1 to 8-character Db2 plan name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBIMSN	Set the 1 to 8-character IMS DB segment name	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBIMDN	Set the 1 to 8-character IMS DB DBD name	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBPSB	Set the 1 to 8-character IMS DB PSB name	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBIVSG	Set the 1 to 8-character IMS view segment name	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBIVDB	Set the 1 to 8-character IMS view DBD name	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBIVPS	Set the 1 to 8-character IMS view PSB name	Character, write

Criterion	Variable	Contents	Data type
GETALIASES. <i>tablename</i>	VTB.OPTBLIST	Set a list of 1 to 50-character table names that are the aliases of map <i>tablename</i> . There is room for up to 637, 50-character alias names that are separated by a blank. More alias names are possible if they are shorter.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBSSD2	Set the DB2 subsystem name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBTCD2	Set the DB2 table creator ID or name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBTND2	Set the DB2 table name.	Character, write
MODIFYTABLE. <i>tablename</i>	VTB.OPTBPND2	Set the DB2 plan name.	Character, write

API functions for rules

This section defines the API functions used for certain IBM Data Virtualization Manager for z/OS rules.

AVZVALUE API function

Use the AVZVALUE function to manipulate global variables.

For example, use the AVZVALUE function to use compound symbols as a type of database. Use this function in a rule that performs special interrogation or serialization processing.

Under normal circumstances, you can use a REXX language statement to reference or set the value of a global variable. The following code shows an example of using a REXX statement:

```
SAVENAME = GLOBAL.COMPANY.NAME
GLOBAL.COMPANY.NAME = "Keiroct Software"
GLVEVENT.MYDATA = "ABC"
```

Syntax

```
val = AVZVALUE(derivedname, actioncode, newval, oldvar)
```

where:

- *derivedname* is the name of the symbol that receives the action. When you use this parameter without quotation marks, simple symbols (case sensitive) following the stem are replaced by their values.
- *actioncode* is the action to take on the symbol.
- *newval* is the new value to assign to the symbol.
- *oldval* is the value of the symbol before the action is taken.

Return values

AVZVALUE returns a value from the function call, and for some action codes, places information in the external data queue.

Action codes

The following table describes the actions that are performed for each action code and the values that are returned.

<i>Table 11. Action Codes and return values</i>			
Action code	Descriptin	Return value	Description
A (Add)	Adds a number, which is specified by increment, to the existing compound symbol given by <i>derivedname</i> . All references to the compound symbol are serialized during the add operation, so you can use this function to increment a counter that is set by concurrent tasks.	<code>val = AVZVALUE (derivedname 'A' increment)</code>	Returns 1 (true) if the comparison finds the pre-action value to be equal to the old value and the compound symbol was updated. Returns 0 (false) if the comparison finds unequal values and does not update the value of the compound symbol. Does not change the external data queue.
C (Compare and update)	Verifies the value of a compound symbol and then updates its value. Safely updates global symbols that more than one rule uses or global symbols that multiple copies of the same rule might access and update. Serializes the compare and update operations for global values.	<code>val = AVZVALUE (derivedname, 'C', newval, oldval)</code>	Returns 1 (true) if the comparison finds the pre-action value to be equal to the old value and the compound symbol was updated. Returns 0 (false) if the comparison finds unequal values and does not update the value of the compound symbol. Does not change the external data queue.

Table 11. Action Codes and return values (continued)

Action code	Description	Return value	Description
D (Drop)	<p>Drops the compound symbol that is specified by <i>derivedname</i>. Resets the compound symbol to its uninitialized value or derived name. If <i>derivedname</i> specifies a stem, all compound symbols that belong to that stem are dropped and the virtual storage that is allocated to them is released. All other references see the compound symbol as it existed before the drop operation started or as it is after the drop operations finishes.</p>	<pre>val = AVZVALUE (derivedname, 'D')</pre>	<p>Returns the value of <i>derivedname</i>. Does not change the external queue.</p>
E (Existence)	<p>Determines whether a global variable exists.</p>	<pre>val = AVZVALUE (derivedname, 'E')</pre>	<p>Returns one of the following values for the status of the global variable:</p> <ul style="list-style-type: none"> • I: Initialized • U: Uninitialized. The variable exists in storage, but it is uninitialized so it is set to the value of its name. • N: Does not exist. The variable does not exist in storage. <p>Does not change the external data queue.</p>

Table 11. Action Codes and return values (continued)

Action code	Description	Return value	Description
F (Find)	Determines whether a global variable exists. The maximum length for a string pulled from the external data queue is 350 bytes. Longer strings are truncated.	<pre>val = AVZVALUE (derivedname, 'F')</pre>	<p>Returns one of the following values for the status of the global variable:</p> <ul style="list-style-type: none"> • I: Initialized • U: Uninitialized. The variable exists in storage, but it is uninitialized so it is set to the value of its name. • N: Does not exist. The variable does not exist in storage. <p>When the return value is I or U, the value of the node is returned in the external data queue.</p>

Table 11. Action Codes and return values (continued)

Action code	Description	Return value	Description
I (Information)	Returns information about all of the immediate subnodes of the <i>derivedname</i> .	<code>val = AVZVALUE (<i>derivedname</i>, 'I')</code>	<p>For each subnode, places two lines in the external data queue. The first line contains the next segment of the <i>derivedname</i>. The second line contains the following information about the <i>derivedname</i>:</p> <ul style="list-style-type: none"> • Word 1, length 8: Number of subnodes under this node. • Word 2, length 8: Create date, in the form yy/mm/dd. • Word 3, length 8: Create time, in the form hh:mm:ss. • Word 4, length 17: Create rule or program name. • Word 5, length 8: Create job name, task name, or TSO ID. • Word 6, length 8: Last modification date. • Word 7, length 8: Last modification time. • Word 8, length 17: Last modification rule or program name. <p>Does not return partially updated symbol names.</p>
L (List)	Lists the derived name of each subnode of the <i>derivedname</i> .	<code>val= AVZVALUE (<i>derivedname</i>, 'L')</code>	Returns the number of subnodes that are listed in the external data queue. Returns dropped symbols, but does not return removed symbols.

Table 11. Action Codes and return values (continued)

Action code	Descriptin	Return value	Description
O (Obtain)	Obtains the value of a global variable.	<code>val = AVZVALUE (derivedname, 'O')</code>	Returns the value of a global variable. If the global variable does not exist, returns an error. Does not change the external data queue.
R (Remove)	Removes the specified node and all of its subnodes. After a node is removed, it ceases to exist.	<code>val = AVZVALUE (derivedname, 'R')</code>	Returns the number of subnodes that were removed. Does not change the external data queue. Does not allow other accessories of compound symbols to see partially updated symbols.
S (Subtree)	Lists the entire global variable name of all subnodes of the <i>derivedname</i> .	<code>val = AVZVALUE (derivedname, 'S')</code>	Returns the entire global variable name of all of the subnodes in the external data queue. Returns the number of subnodes that exist, as listed in the external data queue. Does not return partially updated symbol names.
T (Subtree and information)	Lists the entire global variable name and all subnodes of the <i>derivedname</i> .	<code>val = AVZVALUE (derivedname, 'S')</code>	Returns the entire global variable name and two lines for each subnode in the external data queue. The first line contains the next segment of the <i>derivedname</i> . The second line contains the information., as described for the Information code, for each <i>derivedname</i> . Does not return partially updated symbol names.

Table 11. Action Codes and return values (continued)

Action code	Descriptin	Return value	Description
U (Update)	Assigns <i>newval</i> as the value of the compound symbol that is specified by <i>derivedname</i> . If the compound does not exist, the compound is created and assigned the new value. Use Update to prevent others who access compound symbols from seeing partially updated symbols.	<i>val</i> = AVZVALUE(<i>derivedname</i> , 'U', <i>newval</i>)	Returns the variable that is specified by <i>newval</i> . Does not change the external data queue.
V (Value)	Returns the value of the specified compound symbol. Use Value to prevent the issuer of SDVALUE from seeing partially updated symbols.	<i>val</i> = AVZVALUE(<i>derivedname</i> , 'V')	Returns the current value of the node. If the node does not exist, it is created but it is not assigned a value. Instead, it is given the same value as its name. Does not change the external data queue.

AVZINFO API function

The AVZINFO function retrieves information about the Data Virtualization Manager server subsystem.

The syntax for the AVZINFO function is the following:

```
var=AVZINFO(arg1[,arg2])
```

where *arg1* is a parameter from the following table, and *arg2* is the connection token, which is optional.

The function always returns a return value. If the value requested is not valid for the environment, a NULL string is returned.

Parameter	Return value
ASID	Returns the address space identifier (ASID) as a 2-byte binary value when invoked using the program API. Returns the ASIDD as a 4-byte value when invoked from REXX.
BYTES	Returns the number of saved bytes.
CLOCK	Returns the current time-of-day (TOD) clock value as an 8-byte binary value. This is the unadjusted STCK value.
CONNECTID	Returns the unique connection ID value.
CPUDELT	Returns the 8-byte task CPU time delta value.
CPUTIME	Returns the 8-byte task CPU time value.
DB2PLAN	Returns the name of the Db2 plan.

Parameter	Return value
DB2SUBSYS	Returns the name of the Db2 subsystem.
EVENTTYPE	Returns the type of event that is associated with the rule or program.
HOSTDOMAIN	Returns the host (server) domain that is associated with the current request.
HOSTNAME	Returns the host name (client) associated with the current request.
IPADDRESS	Returns the fully formatted IP address for the current request in the form 10.17.16.164.
JOBNAME	Returns the z/OS job name that is related to the current primary address space.
LASTCONNECTID	Returns the last connection ID used on the current link.
LASTUSERID	Returns the last user ID used on the current link.
LINKTYPE	Returns the link type for the current request.
LU	Returns the LU name for the current request.
MAINPGM	Returns the name of the main REXX program or rule.
MODE	Returns the mode name for the current request.
ODBCDATE	Returns the compile date of the .NET Client (ODBC).
ODBCVERSION	Returns the version of the .NET Client (ODBC).
PRODUCT	Returns the product identification string.
PRODUCTSTATUS	Returns the current product status.
PROGRAM	Returns the name of the REXX program or rule.
ROWS	Returns the number of source rows.
SEFFEATURE	Returns a single blank if the Server Event Facility (SEF) is not enabled.
SUBSYS	Returns the accessed subsystem ID from the current OPMS image.
SUBSYSASID	Returns the ASID of the active subsystem from the real OPMS as a 2-byte binary value when invoked by using the program API and as a 4-byte value when invoked from REXX.
SMFID	Returns the SMF ID.
TASKTYPE	Returns the task type.
TRANSTYPE	Returns the transaction program type.
USERID	Returns the user ID value.
USERPARM	Returns the user parameter string from the client.

Parameter	Return value
VERSION	Returns, as a string, the version of the product subsystem under which the rule or program is running.

Examples

The following call sets the REXX variable, IPA, to the fully formatted TCP/IP address of the client program:

```
IPA = AVZINFO(IPADDRESS)
```

The following call sets the variable *USER* to the user ID value of the connection that caused the exception. In this example, EXC.OPEXCNTK, which contains the connection token, is used to obtain the user ID because the exception rule runs under the OPCKLM (check limits) task, not the user connection task:

```
USER = AVZINFO(USERID,EXC.OPEXCNTK)
```

AVZECURE API function

The AVZECURE function performs security-authorization processing.

Verify data set access

To verify that the current user has authorization to access a data set, use the following syntax:

```
var = AVZECURE('D', 'dsname', 'accesstype', 'volser')
```

where:

- *dsname* is the name of the data set.
- *accesstype* is the type of data set access to verify. If you do not specify a type, READ access is the default. Valid values are:
 - A: Verify ALTER access.
 - C: Verify CONTROL access.
 - R: Verify READ access.
 - U: Verify UPDATE access.
- *volser* is the volume serial number to validate. If you do not specify a volser, the parameter is blank, by default.

The function returns a message that indicates whether access is allowed.

Retrieve logon ID field data

To retrieve security subsystem information from the current user's ACEE, use the following syntax:

```
var = AVZECURE('F', 'fieldname')
```

where *fieldname* is one of the fields in the following table:

Field	Description	Field format
ALTER	Alter authority flag	Bit
APPLICATION	Application name	Character
APPLICATIONDATA	Application data	Character
APPLICATIONLEVEL	Application level	Binary

Field	Description	Field format
AUDITOR	Auditor attribute	Bit
AUTOMATIC	Automatic attribute	Bit
CLASSAUTHORIZATIONS	Class authorizations	Binary
CONTROL	Control authority flag	Bit
DATE	Date	RACINT date
DEFINEUSERS	Authorized to define users	Bit
GROUP	Contents of the ACEE group field	Character
GROUPLIST	A list of groups	Character
GROUPLISTCONTAINS	Group list contents flag	Bit
INSTALLATIONDATA	Contents of the installation data field	Character
LOG	Logging on for most operations	Bit
NONE	None authority flag	Bit
OPERATIONS	Operations attribute	Bit
PORTOFENTRYDATA	Port of entry data	Character
PORTOFENTRYLEVEL	Port of entry level	Binary
PRIVILEGED	Started task with privileged flag	Bit
PROTECTDASD	Authorized to protect DASD	Bit
PROTECTTAPE	Authorized to protect tape	Bit
PROTECTTERMINALS	Authorized to protect terminals	Bit
RACF®	RACF-defined user flag	Bit
READ	Read authority flag	Bit
SPECIAL	Special attribute	Bit
STCNAME	Started task name	Character
SURROGATEUSERID	Surrogate user ID	Character
TERMINAL	Terminal ID	Character
UPDATE	Update authority flag	Bit
USERDATA	Contents of the user data field	Character
USERID	Contents of the ACEE user ID field	Character
USERNAME	User name field	Character
VERSION	ACEE version code	Binary

The following conversions occur, based on the field format:

- Binary fields are converted to signed decimal values without leading zeroes or blanks. The number zero is returned as 0.

- Character fields are returned as is. If a character field name exceeds the maximum allowed string length, it is truncated to the Data Virtualization Manager configuration/REXX-defined maximum string length.
- Date fields are converted to the format *yyyy/mm/dd*. Leading zeros are retained so that the result is always 10 non-blank characters. A date field that contains zero is returned as *****/**/***.
- Bit fields are converted to 0 (false or off) or 1 (true or on).
- The GROUPLIST field inquiry returns an integer that represents the number of entries in the group list. Each group name is returned as a separate entry in the external data queue.

Request security product information

To retrieve information about the security product, use the following syntax:

```
var = AVZECURE('i', 'name')
```

where *name* is one of the values in the following table:

Value	Return value
MODE (Valid only for systems that run ACF2)	Returns one of the following ACF2 operating modes: ABORT, LOG, OFF, WARN, QUIET.
PRODUCT	Returns the name of the security product or the message UNKNOWN SECURITY PRODUCT.
RELEASE	Returns the release and version number for the security product.

If the information cannot be obtained, a NULL string is returned.

Verify access to a generalized resource

To verify that the current user has access to a generalized resource, use the following syntax:

```
var = AVZECURE('R', class, resource, requestcode)
```

where:

- *class* is the generalized resource class name or for ACF2, the type name.
 - Note:** Rules that verify access to resources use SAF processing. If you use ACF2, you must define the ACF2 resource type as a SAF class name.
- *resource* is the 1- to 39-byte resource entity name.
- *requestcode* is the type of access to verify. If you do not specify a request code, READ access is the default. The following are valid values:
 - A: Verify ALTER access..
 - C: Verify CONTROL access.
 - R: Verify READ access.
 - U: Verify UPDATE access.

If access to the resource is allowed, the string ALLOW is returned. Otherwise, an error message is returned.

Verify a user ID and password

Use the following syntax to verify the user ID and password. If the password is valid, the user is logged on to the system. This API call is valid only for ATH events.

```
var = AVZECURE('P', 'userid', 'password', 'newpassword')
```

where:

- *userid* is the user ID to validate.
- *password* is the password that is associated with the user ID.
- *newpassword* is the new password to associate with the user ID.

If you omit the *newpassword* parameter, the user ID and password are validated. If you specify the *newpassword* parameter, the password is changed.

If the password is correct, the return value is the string ALLOW. If the password is incorrect, an error message is returned. For ACF2, the counter for invalid password violation for the specified user ID is incremented for each failed attempt.

Use an implied password to validate a user ID

This request causes the specified user ID to be validated. If the password is valid, the user is logged on to the system. The password is not specified on the function call. Instead, the initial inbound transaction request transmits the password. Use this function to perform custom security checks without making the clear text password available to the procedure. This API call is valid only for ATH events.

Use the following syntax to use an implied password to validate a user ID:

```
var = AVZECURE('PI', 'userid', 'newpassword')
```

where:

- *userid* is the user ID to validate.
- *newpassword* is the new password to associate with the user ID.

If you omit the *newpassword* parameter, the function uses the implied password to validate the user ID. If you specify *newpassword*, the function changes the password. If the password is correct, the return value is the string ALLOW. If the password is incorrect, an error message is returned. For ACF2, the counter for invalid password violation for the specified user ID is incremented for each failed attempt.

AVZSUBMIT API function

Use the AVZSUBMIT function to submit JCL to the internal reader and return the JES2 or JES3 job ID for each submitted job.

The AVZSUBMIT function can be invoked as a function reference, which returns its result to the point of invocation, or as a REXX CALL statement. There is no corresponding TSO/E REXX or high-level language (HLL) API interface.

- The JCL statements read from the input stream can be any size; however, each individual statement is extended or truncated to be 80 bytes when submitted through the internal reader.
- In cases where the JCL input stream is ASCII or UTF-8 encoded, for example, for POSTED input, the function converts the JCL stream to IBM-1047 EBCDIC. Only rudimentary UTF-8 support is available, so avoid including double-byte characters and ASCII characters above code point 0x7F.
- The function provides no editing and imposes no restrictions on the content and format of JOB statement names in the JCL that is submitted.
- To detect job boundaries, the function scans each JCL statement. The following situations indicate a job boundary:
 - The JCL statement begins with “//”, followed by an uppercase EBCDIC Latin letter or one of the IBM 1047 EBCDIC characters “@”, “\$”, or “#”.
 - The prefix is followed by 0 - 7 Latin letters or numbers or the IBM 1047 EBCDIC characters “@”, “\$”, or “#”.
 - The next blank-delimited word is JOB. After this word is found, the scan stops parsing the statement.

- The scan does not take into account quoted string boundaries that enclose continued PARM= operands and does not detect, honor, and process JCL statement continuations.
- Jobs that are submitted while a client user ID logon are in effect are given a USER attribute that matches the logon ID of the client subtask. If the JCL USER= operand of the JOB statement is present and differs from the client task logon ID and PASSWORD= is not present, RACF surrogate user attribute assignment and authorization restrictions might be imposed.
- The AVZSUBMT function can be used only in REXX language rules. The function cannot be used in a rule that runs in cross-memory mode or one for which waiting for system services is inhibited. Areas where AVZSUBMT cannot be used or can be used only conditionally include the following:
 - AVZSUBMT cannot be used during enabling or disabling a rule, which occurs when the PHASE variable is not set to PROC.
 - AVZSUBMT cannot be used in CMD, GLV, and TYP rules.
 - To determine when AVZSUBMT can be used, an ATH rule can check the value of the ATH.OPAU13WA variable, and an EXC rule can check the value of the EXC.OPEXWAOK variable. If AVZSUBMT can be used, the variable is preset to 1.

Use the following syntax:

```
AVZSUBMIT(arg1, arg2, arg3, arg4 )
```

or

```
CALL AVZSUBMIT(arg1, arg2, arg3, arg4 )
```

where:

- *arg1* and *arg2* specify the location of the input JCL stream.
- *arg3* specifies the 1-character JES class to which the internal reader is allocated.
reader is allocated.
- *arg4* is a string that specifies the type of tracing.

The following table lists the valid values for *arg1* and *arg2*:

Value	arg1: Location of the JCL input stream	arg2
STEM	The JCL is in a REXX stem variable array. The 0th entry in the array contains the count of entries. Entries 1 - <i>n</i> contain individual JCL statements.	The REXX variable stem name. The name must end with a period. Length 1- 12 character.
DSN	The JCL is in a z/OS data set.	A fully qualified z/OS data set name. The name can include a PDS(E) member name. Length 1- 54 bytes.
DDN	The JCL is in a z/OS data set that is preallocated to a DD name.	The DD name. Length 1- 8 bytes.
PATH	The JCL is in a USS HFS file.	The fully qualified HFS path name of the file. Length 1 - 256 bytes.
POSTED	The JCL is received as a posted file entity over HTTP.	The index number, 1 to <i>n</i> , of the posted file entity in the received HTTP request. If this argument is omitted, the default value is 1.

arg3 is the 1-character JES class to which the internal reader is allocated. The character A - Z, 0 - 9, and * (asterisk) are valid. Use * to request the default job class. If you do not specify this parameter, * is the default.

arg4 is a string that is 1 - 5 bytes. Each character of the string must be Y or N to specify whether the corresponding trace function for that byte is enabled. The following table describes the byte positions and trace functions:

Byte position	Default	Trace function
1	Y	Trace JOB IDs that JES returns.
2	Y	Trace input source JCL.
3	Y	Trace the dynamic-allocation activity of the internal reader.
4	N	Trace writes to the internal reader.
5	N	Trace the decoding of posted data (conversion to EBCDIC).

Unless a REXX ERROR or FAILURE signal is generated because of a fault condition, *arg4* returns one of the following numeric results:

- 0: Successful completion
- 4: Parameterization error
- 8: Environmental error
- 12: System service error
- 16: ABEND condition that is trapped
- +100: If one or more jobs are submitted before a failure, the value +100 is added to a result. To determine the failure code, subtract 100.

JOBID. stem variables

The function uses a REXX DROP on all JOBID. stem variables during entry-processing and presets variables to the values shown in the following table. This reset operation occurs after initial parameter validation but before JCL processing. If the reset fails, the REXX invalid symbol signal is generated. After setup, unless a REXX signal is thrown, the JOBID.RC, JOBID.REASON, JOBID.0, and JOBID.*n* variables are set as described. All other JOBID. stem variables are undefined.

Variable	Description
JOBID.RC	Contains the same value as the evaluated RESULT of the function call or if a problem is detected before all other JOBID. stem variables are correctly set, contains a NULL string. JOBID.RC is set to a NULL string at entry, and setting this variable to the RESULT is the last action that the function takes before exit.
JOBID.REASON	When the function call ends with a non-zero RESULT, contains error text. This variable is set to a NULL string when the RESULT is zero.

Variable	Description
JOBID.0	Contains an integer that indicates the number of jobs that were found in the input JCL stream and successfully submitted to the internal reader. If no jobs were successfully submitted or if a system failure prevented the return of any job IDs during processing, this variable contains 0 (zero). If one or more jobs are submitted before a failure, this variable contains the number of submitted jobs for which IDs were returned.
JOBID. <i>n</i>	Contains the job ID that is assigned to the first through <i>n</i> th job in the submitted JCL stream. Valid job IDs are in the format JOBxxxxx or Jxxxxx, where xxxxx is a system-assigned sequence number. Only the variables JOBID.1 through JOBID. <i>n</i> , where <i>n</i> is the numeric value that is assigned to JOBID.0 are set.

DISPLAY command

Use the DISPLAY command to display information about all connected users.

Displaying basic information

Use the following syntax to display basic information about all connected users:

```
"DISPLAY REMOTE USERS(*)"
```

This command displays the following information about each connected user:

- ACTUAL BLOCK ADDRESS
- APPLICATION NAME
- CONNECTION ID
- DB2 SUBSYSTEM
- HOST NAME
- ICUV PATH ID
- IP ADDRESS
- LINK TYPE
- LOCAL IP PORT NUMBER
- REMOTE IP PORT NUMBER
- SOCKET NUMBER
- TRUSTED HOST
- USER ID
- TASK TCB ADDRESS
- TRUSTED HOST
- USER ID

Displaying additional information

Use the following syntax to display additional information about all connected users:

```
"DISPLAY REMOTE USERS(*) VERBOSE"
```

This command provides the following additional information about each connected user:

- ACEE SOURCE
- BUFFER FUNCTION CODE
- COMPRESSED SEND AMOUNT
- COMPRESSED TOTAL BYTES RECEIVED
- CPU TIME
- CUMULATIVE COMPRESSION
- CUMULATIVE RECEIVED COMPRESSION
- CURRENT COMPRESSED RECEIVED AMOUNT
- CURRENT RAW RECEIVED AMOUNT
- CURRENT STATE
- DB2 PLAN NAME
- DB2 REQUESTING SITE
- DB2 THREAD TOKEN
- DOMAIN NAME
- ELAPSED TASK TIME
- EXTENDED USER ID
- GENERIC USER ID
- HOST TIME
- INTERNAL NAME
- LAN USER ID
- LOCKS HELD
- MODULE NAME
- ODBC DRIVER DATE
- ODBC DRIVER VERSION
- PROGRAM NAME
- RAW BYTES RECEIVED
- RAW BYTES SENT
- RAW RECEIVED COMPRESSION
- RAW SEND AMOUNT
- RAW SEND COMPRESSION FACTOR
- SQL CODE
- SQL COUNT
- SQL CURSOR NUMBER
- SQL REASON CODE
- SQL RETURN CODE
- SQL STATEMENT NUMBER
- SQL STATEMENT TYPE
- STATE DURATION
- TELEPROCESSING TIME
- TELEPROCESSING TIME PERCENTAGE

- TOTAL RAW BYTES SENT
- USER PARAMETER
- WLM ENCLAVE COUNT
- WLM ENCLAVE CPU TIME

Chapter 5. Configuring access to data sources

Configure the Data Virtualization Manager server to enable access to mainframe data sources.

Configuring access to Adabas

To access Adabas, you need to configure the started task JCL and the Data Virtualization Manager configuration member.

Before you begin

The server must be installed.

About this task

The SQL interface for Adabas provides seamless, real-time controlled access to Adabas data. It allows ODBC, JDBC, and web clients to access Adabas data in a relational model using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to Adabas. The result is a relational result set, with no host programming required.

Procedure

To configure access to data in Adabas, perform the following tasks:

- a) Configure the server started task. See [“Configuring the server started task JCL”](#) on page 101.
- b) Modify the server configuration member. See [“Modifying the Data Virtualization Manager configuration member”](#) on page 102.
- c) Configure security access to Adabas data. See [“Configuring security access to Adabas data”](#) on page 103.

Configuring the server started task JCL

Make the ADALNKR module available.

Before you begin

All LOAD library data sets allocated to the Data Virtualization Manager server in the server started task JCL must be APF-authorized.

About this task

Note: You can skip this task if the ADALNKR module is in the z/OS linklist.

Procedure

1. Add the Adabas LOAD library to the server started task JCL. Uncomment the ADALOAD parameter and set it to the correct Adabas load library name.

```
ADALOAD=' ADABAS . LOAD '
```

2. Uncomment the reference to the LOAD library in the STEPLIB.

Modifying the Data Virtualization Manager configuration member

Enable the Adabas parameters in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment “ENABLE ADABAS DATABASE SERVER SUPPORT.”
2. Enable the Adabas parameters by changing the syntax `if DontDoThis` to `if DoThis`.

Set the ADABAS parameter to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
do
  "MODIFY PARM NAME(ADABAS)VALUE(YES)"
  "MODIFY PARM NAME(ADABASUBINFOSIZE)VALUE(256)"
  "MODIFY PARM NAME(ADABASAUTO COMMITBIND)VALUE(YES)"

  "MODIFY PARM NAME(ACIMAPREDUCEADAB)VALUE(64000)"
  "MODIFY PARM NAME(ACIMAPREDUCEADAISN)VALUE(64000)"
end
```

The following table lists the parameters for configuring support for Adabas data stores:

Parameter	Description	Valid values
ACIMAPREDUCEADAB	Map Reduce Adabas Record Buffer Size - Allows Adabas Multi-Fetch used to read records via L1 commands. If the Adabas ADARUN limits are exceeded, an Adabas response code 53 is issued.	Buffer size in bytes. 64000 (default value)
ACIMAPREDUCEADAISN	Map Reduce Adabas ISN Buffer Size - When a Key Descriptor is used in a Search query, an Adabas S1 search is performed. The resulting internal sequence number (ISN) Record number list is divided up into separate Map Reduce threads.	Buffer size in bytes. 64000 (default value)
ADABAS	Activates support for Adabas data stores.	NO Support is not active. (default value) YES Activate support.
ADABASAUTO COMMITBIND	Activates support for the AUTO COMMIT BIND option.	YES Activate support. (default value) NO Support is not active.

Parameter	Description	Valid values
ADABASUBINFOSIZE	Specifies the total amount of space to allocate for user information and review information in the Adabas user block. Review the maximum user information size in the ADALNKR, and increase the value of this parameter to be equal to or greater than the maximum user information size.	256 KB (default value)

Configuring security access to Adabas data

Configure security to provide access to Adabas.

Procedure

See "Configuring Adabas security" in the *Administration Guide*.

Configuring access to RDBMS

You can access data on DB2 for z/OS and distributed databases IBM Big SQL, IBM dashDB, DB2 LUW (Linux, UNIX, and Windows), Microsoft SQL Server, Oracle, and QMF DRDA.

Before you begin

The server and relational database management system (RDBMS) must already be installed.

About this task

The SQL interface for DB2 provides seamless, real-time controlled access to RDBMS data. It allows ODBC, JDBC, and web clients to access RDBMS data in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to RDBMS. The result is a relational result set, with no host programming required.

To configure and verify access to data in a RDBMS, perform the following tasks.

Procedure

1. Enable the RDBMS access method in the Data Virtualization Manager configuration member.

See [“Modifying the DRDA configuration member” on page 105](#).

2. Configure access to the database.

- IBM DB2 for z/OS

Configure DB2 to use either the Distributed Relational Database Architecture (DRDA) access method or the Resource Recovery Services attachment facility (RRSAF) access method.

See [“Configuring access to IBM Db2 for z/OS” on page 104](#).

- Distributed databases, including Big SQL, dashDB, DB2 LUW, Microsoft SQL Server, Oracle, and QMF DRDA.

Configure the RDBMS to use the Distributed Relational Database Architecture (DRDA) access method.

See [“Configuring access to distributed databases” on page 119](#).

Configuring access to IBM Db2 for z/OS

Configure Db2 to use either the Distributed Relational Database Architecture (DRDA) access method or the Resource Recovery Services attachment facility (RRSAF) access method.

About this task

Using DRDA might yield a lower total cost of ownership than RRSAF because DRDA allows a higher percentage of Db2 work to run in SRB mode and offloaded to a zIIP specialty engine.

If you have a zIIP specialty engine, use DRDA. If you do not have a zIIP specialty engine, use RRSAF.

Before you issue Db2 requests, you must bind DRDA, RRSAF, or both into packages within each Db2 subsystem. Binding both access methods is recommended.

Configure access to Db2 for z/OS databases as follows.

Procedure

1. [“Configuring security” on page 104](#)
2. Configure for DRDA (Distributed Relational Database Architecture) or for RRSAF (Resource Recovery Services attachment facility) access method.
 - If you are using a zIIP specialty engine, enable the RDBMS access method for DRDA:
 - a. [“Modifying the DRDA configuration member” on page 105](#)
 - b. [“Configuring DB2 for DRDA” on page 111](#)
 - If you are not using a zIIP specialty engine, enable the RDBMS access method for RRSAF:
 - a. [“Modifying the RRSAF configuration member” on page 111](#)
 - b. [“Configuring Db2 for RRSAF” on page 113](#)

Configuring security

Configure security to provide user access to DB2.

About this task

If the DB2 being accessed does not have the DSNZPARM DDF option TCPALVER set to either YES or CLIENT, then a passticket is needed for certain DB2 on z/OS DRDA operations. These operations may include:

- Refreshing in-memory metadata catalog information at server startup for DB2 on z/OS defined virtual tables. Catalog information is refreshed at every server startup by the Data Virtualization Manager server connecting to each DB2 where virtual tables have been defined.
- Any SQL statement coming from the dsClient interface, dsSpufi or application APIs using the dsClient interface. This may also include running administrative tasks in batch using dsClient that accesses DB2 on z/OS such as updating MapReduce information using the DRDARange command.

Procedure

1. This step only applies to DB2 for z/OS. To grant users access to the DB2 subsystem and to enable passticket logon processing, you must define one RACF PTKTDATA resource for each unique DRDA APPLNAME. To define each PTKTDATA resource, customize and run the appropriate job.
 - AVZRADB2 is for IBM Resource Access Control Facility (RACF) security.
 - AVZA2DB2 is for CA ACF2 (Access Control Facility) security.
 - AVZTSDB2 is for CA Top Secret Security (TSS).

2. Assign users READ authority.

- For DRDA, assign users READ authority to the *ssid*.DIST profile.
- For RRSAF, assign users READ authority to the *ssid*.RRSAF profile.

Configuring the server started task JCL

If you use Db2 z/OS, add the Db2 load library to the server started task JCL.

Before you begin

All LOAD library data sets allocated to the Data Virtualization Manager server in the server started task JCL must be APF-authorized.

About this task

This task is not applicable for Db2 UDB (Linux, UNIX, and Windows).

Note: Skip this task if the Db2 interface modules, DSNALI and DSNHLI, are in the z/OS linklist.

Procedure

Edit the JCL in the *hlq*.SAVZCNTL (AVZ1PROC) member to include in the PROC statement the DB2LIB parameter with the Db2 library name assigned, as shown in the following example:

```
DB2LIB= ' DSNX10 '
```

The Db2 library must contain the Db2 interface modules, such as DSNALI and DSNHLI, and must be in uppercase and enclosed in quotation marks.

Modifying the DRDA configuration member

If you are using a zIIP specialty engine, enable the RDBMS access method for Distributed Relational Database Architecture (DRDA) in the Data Virtualization Manager configuration member.

About this task

Configure the server to use Distributed Relational Database Architecture (DRDA) when accessing a RDBMS.

Modify the Data Virtualization Manager configuration member in data set *hlq*.AVZS.SAVZEXEC(AVZSIN00). The Data Virtualization Manager configuration member is shipped in data set member *hlq*.SAVZEXEC(AVZSIN00) and copied to *hlq*.AVZS.SAVZEXEC(AVZSIN00) by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the DEFINE DATABASE statement and the CCSID used by the target RDBMS are defined for your z/OS environment.

a) You should identify the CCSID of the RDBMS.

For example, Oracle may use *ccsid1*. In your DEFINE DATABASE statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **D UNI,ALL**.

b) If the entry is not present, you need to add the entry to your Unicode translation table and refresh.

Please refer to the IBM z/OS documentation on how to add the entry.

Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

```
"DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE) "
"DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE) "
```

2. In the AVZSIN00 member, locate the section that contains the comment Enable DRDA access to DB2 database subsystems.
3. Enable the DRDA parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the DRDASKIPZSERVICES parameter to YES. The following example shows the section in the configuration member to enable:

```
/*-----*/
/* Enable DRDA access to DB2 database subsystems */
/*-----*/
if DoThis then
do
    "MODIFY PARM NAME(TRACEOEDRDARW) VALUE(YES)"
    "MODIFY PARM NAME(CLIENTMUSTELECTDRDA) VALUE(NO)"
    "MODIFY PARM NAME(DRDASKIPWLMSETUP) VALUE(NO)"
    "MODIFY PARM NAME(DRDAFORLOGGINGTASK) VALUE(NO)"
    "MODIFY PARM NAME(DRDASKIPZSERVICES) VALUE(YES)"
```

The following table describes these parameters:

Parameter	Description	Valid values
TRACEOEDRDARW	<p>If set to YES (recommended), TCP/IP communications via DRDA are traced.</p> <p>If set to NO, DRDA receive and send operations are not traced.</p>	<p>YES</p> <p>NO</p> <p>Default value.</p>
CLIENTMUSTELECTDRDA	<p>If set to YES, the ODBC and JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'.</p> <p>Note: ODBC and JDBC clients can always opt out of DRDA processing by setting the user parameter to 'NODRDA'.</p> <p>If set to NO, DRDA processing is used for access all configured RDBMSs.</p>	<p>YES</p> <p>NO</p> <p>Default value.</p>

Parameter	Description	Valid values
DRDASKIPWLMSETUP	<p>If set to YES, WLM information is not collected and sent to DRDA during ODBC/JDBC logon processing. If captured, the DRDA equivalent to SET_CLIENT_ID calls is issued after logon to establish these values on the DRDA connection. If not captured, the transmission that is used to set these WLM-related values is bypassed.</p> <p>If set to NO, the client user ID, application name, workstation name, and accounting token that were sent in the initial client buffer are collected and sent separately after logon processing to DRDA.</p>	<p>YES NO Default value.</p>
DRDAFORLOGGINGTASK	<p>If set to YES, DRDA processing is used for the Db2 on z/OS logging subtask.</p> <p>If set to NO, SAF or RRSF connections are used.</p> <p>Note: Passticket support must be enabled for the target DDF server. If passticket support is not configured, set the parameter to NO.</p>	<p>YES NO Default value.</p>
DRDASKIPZSERVICES	<p>Prevents DRDA from being used for z/Service Db2 processing.</p> <p>If set to YES, z/Services client tasks do not use DRDA processing for Db2 requests.</p> <p>If set to NO, DRDA will be used when configured for a particular Db2 connection.</p> <p>Note: Passticket support must be enabled for all target DDF servers.</p>	<p>YES NO Default value.</p>

4. Define DRDA RDBMSs by entering a definition statement. Provide your local environment values for all the parameters. The following example shows the section in the configuration member to enable:

```
"DEFINE DATABASE TYPE(type_selection) "      ,
"NAME(name) "                                ,
"LOCATION(location) "                          ,
"DDFSTATUS(ENABLE) "                         ,
"DOMAIN(your.domain.name) "                 ,
"PORT(port) "                                ,
"IPADDR(1.1.1.1) "                           ,
"CCSID(37) "                                  ,
```

```
"APPLNAME (DSN1LU) "
"IDLETIME (110) " ,
```

Where *type_selection* is either GROUP, MEMBER, or ZOSDRDA.

The previous example shows only a subset of the available parameters. The following table lists all available parameters for defining DDF endpoints:

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. <i>(Optional)</i>	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. Note: APPLNAME is not required when connecting from the ODBC/JDBC driver.
AUTHTYPE	Authentication type. This can be either DES (Diffie Hellman Encryption Standard) or AES (Advanced Encryption Standard). When AUTHTYPE is not supplied, the default is DES. To force AES, the option must be added to the DEFINE DATABASE statement. Each server can be different in what is supported as to AES/DES. For this setting to have effect, you must specify a security mechanism (SECMEC) that requests encryption.	DES Diffie Hellman Encryption Standard (default value) AES Advanced Encryption Standard.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. <i>(Optional)</i>	Refer to the RDBMS vendor documentation for a list of valid CCSID.
DDFSTATUS	The DDF activation status can be altered online by using the ISPF 4-Db2 dialog panels. <i>(Required)</i>	ENABLE To make this DDF definition active within Data Virtualization Manager server. DISABLE DDF endpoint is not used.
DOMAIN	The part of a network address that identifies it as belonging to a particular domain.	No default value.

Parameter	Description	Valid values
IDLETIME	If Db2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. <i>(Db2 for z/OS only)</i>	0-9999 seconds.
IPADDR	Specify the dot-notation IPV4 address of the DDF endpoint. <i>(Optional)</i>	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.
LOCATION	For Db2: The Db2 location name. For LUW: The LUW database. For Oracle: The Oracle SSID as defined to the Oracle Database Provider (Gateway) <i>(Required)</i>	A valid value is a string 1 - 16 characters.
NAME	The database name as known to the server. <i>(Required)</i>	A valid value consists of 1 - 4 characters. Clients use this ID when they request access to a specific Db2 subsystem.
PORT	The TCP/IP port at which the server is listening. <i>(Required)</i>	If this keyword is not entered, the default DRDA port number 443 is used.

Parameter	Description	Valid values
SECMEC	The DRDA security mechanism in force. (For <i>GROUP</i> and <i>MEMBER</i> types.)	<p>USRIDPWD User ID and password are sent as is. No encryption is used.</p> <p>USRIDONL User ID is sent as is. No encryption is used for the user ID only (client security).</p> <p>USRENCPWD Encrypt the password only.</p> <p>EUSRIDPWD Encrypt the user ID and password.</p> <p>USRSBSPWD User ID and password substitute. A RACF pass ticket is generated.</p> <p>Note: SECMEC 6 is added in the DEFINE DATABASE. When used DRDA ignores the password and generates a substitute password for connection to the DB2 database. In ACCSEC and SECCHK SECMEC 3, USRIDPWD is used.</p>
SYSTEMVCAT	The VCATNAME for the Db2 system catalog tables (in the DSNDB06 database). The VCATNAME for system catalog tables is a system bootstrap value and not available using the data discovery query. Use this parameter if you intend to access the system catalog tables using Db2 Direct or if the VCATNAME for database DSNDB06 is different from the subsystem name.	<p>A valid value is 1 - 8 characters.</p> <p>If this parameter is not specified, the 4-character Db2 subsystem name is used by default as the high-level qualifier for Db2 data sets.</p>

Parameter	Description	Valid values
TYPE	<p>For Db2 for z/OS:</p> <p>GROUP DDF endpoint is a Db2 group director.</p> <p>MEMBER DDF endpoint is a Db2 instance or group member for z/OS.</p> <p>ZOSDRDA DDF endpoint is a remote z/OS Db2 on another LPAR.</p> <p>This setting allows you to use SEF ATH rules when z/OS Pass Ticket passwords cannot be used or the server administrator has the requirement to manage the authentication credentials for remote z/OS systems.</p>	<p>For Db2 for z/OS:</p> <p>GROUP</p> <p>MEMBER</p> <p>ZOSDRDA</p>

Configuring DB2 for DRDA

If you are using a zIIP specialty engine, configure DB2 to use DRDA.

About this task

Before you can successfully issue DRDA requests, you must bind IBM Data Virtualization Manager for z/OS DBRMs into packages within each target DB2 subsystem.

Procedure

1. Set the DEFAULTDB2SUBSYS parameter in the Data Virtualization Manager configuration member to a valid DB2 subsystem name.
2. Edit the AVZBINDD job that is supplied in the *hlq.SAVZCNTL* data set.
Follow the instructions that are provided in the JCL.
3. Run the AVZBINDD job.

Modifying the RRSF configuration member

If you are not using a zIIP specialty engine, enable the RDBMS access method for Resource Recovery Services attachment facility (RRSAF) in the Data Virtualization Manager configuration member.

About this task

This task is only applicable for Db2 for z/OS.

Modify the Data Virtualization Manager configuration member in data set *hlq.AVZS.SAVZEXEC(AVZSIN00)*. The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the DEFINE DATABASE statement and the CCSID used by the target RDBMS are defined for your z/OS environment.

- a) You should identify the CCSID of the RDBMS.

For example, Oracle may use *ccsid1*. In your DEFINE DATABASE statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **D UNI,ALL**.

- b) If the entry is not present, you need to add the entry to your Unicode translation table and refresh.

Please refer to the IBM z/OS documentation on how to add the entry.

Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

```
"DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE) "
"DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE) "
```

2. Set the DEFAULTDB2SUBSYS parameter in the Data Virtualization Manager configuration member AVZSIN00 to a valid Db2 subsystem name.
3. In the AVZSIN00 member, locate the comment ENABLE DB2 RRSF SUPPORT section.
4. Enable the RRSF parameters by changing the syntax `if DontDoThis` to `if DoThis`. The following example shows the section in the configuration member to enable:

```
if DoThis then
do
"MODIFY PARM NAME(RRS) VALUE(YES) "
"MODIFY PARM NAME(DB2ATTACHFACILIT) VALUE(RRS) "
"MODIFY PARM NAME(TRACERSSDATA) VALUE(NO) "
"MODIFY PARM NAME(TRACERSSEVENTS) VALUE(YES) "
"MODIFY PARM NAME(TRACERSSAF) VALUE(YES) "
end
```

The following table lists the parameters for configuring support for RRSF:

Parameter	Description	Valid values
DB2ATTACHFACILITY	Specifies the Db2 attach facility. The Resource Recovery Services attachment facility (RRSAF) uses the DSNRLI interface module and allows for 2-phase commit actions. The Call Attach Facility (CAF) uses the DSNALI interface module.	The default value is RRS. Valid values are RRS and CAF.
RRS	Activates RRS support. This parameter must be set to YES to activate RRS.	YES Default value. NO
TRACERSSDATA	Specifies whether to trace RRS data.	YES Default value. NO
TRACERSSEVENTS	Specifies whether to trace RRS events.	YES Default value. NO

Parameter	Description	Valid values
TRACERSSAF	Creates an entry in the server trace for each call to DSNRLI for RRSAF requests.	YES Default value. NO

Configuring Db2 for RRSAF

If you are not using a zIIP specialty engine, configure RRSAF for access to local Db2.

About this task

This task only applies to Db2 for z/OS.

Procedure

1. Run the AVZBINDC member of the *hlq.SAVZCNTL* data set to bind the following server product plans:
 - AVZC1010 is bound using cursor stability.
 - AVZR1010 is bound using repeatable read.
 - AVZS1010 is bound using read stability.
 - AVZU1010 is bound using uncommitted read.

Use AVZC1010 as the default server plan, and use the other product plans for operations that require those levels of isolation. To change the default plans, edit the BIND member and replace the default plan names with new names. You must run the BIND job of the *hlq.SAVZCNTL* data set against each Db2 subsystem that you want to access. Use the instructions in the JCL to customize the job.

2. Install the DSN3@SGN exit in the Db2 master task (normally placed in the SDSNEXIT data set). Installing this exit enables the server to use Db2 authority that was granted through secondary Db2 authorization IDs.

Disabling query acceleration

You can use a Data Virtualization Manager Server Event Facility (SEF) rule to disable the SET CURRENT QUERY ACCELERATION command when you do not want to use the accelerator for certain DB2 virtual tables.

About this task

By default, the server sends the command **SET CURRENT QUERY ACCELERATION = ENABLE WITH FAILBACK** to a DRDA server if it is DB2 for z/OS. This setting allows access to accelerator tables but does not prevent access to non-accelerator tables. Sending the command can be suppressed using the virtual table rule AVZMDTBL by setting the field **OPTBNOQA** to **1** in the rule. If sending the command is suppressed and the table is an accelerator only table, the query will fail. This setting has effect only when the table is owned by a DB2 for z/OS subsystem and the table is an accelerator table; otherwise, there is no impact to the processing.

Use the following procedure to set up the rule.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

2. Access the VTB rules, as follows:
 - a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**

- b) Specify option 2, **SEF Rule Management**.
 - c) Enter VTB for **Display Only the Ruleset Named**.
3. Customize the AVZMDTBL rule, as follows:
 - a) Specify S next to AVZMDTBL to edit the rule.
 - b) Find the **VTB.OPTBNOQA** variable and set to 1 to turn query acceleration off.
 - c) Save your changes and exit the editor.
 4. Enable the rule by specifying E next to AVZMDTBL and pressing Enter.
 5. Set the rule to Auto-enable by specifying A next to AVZMDTBL and pressing Enter.
- Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring access to Db2 unload data sets

To be able to access a Db2 unload data set directly with an SQL query, you must configure a virtual table rule to define the Db2 unload data set name to the Db2 virtual table.

About this task

To configure access to a Db2 unload data set, you must add the Db2 unload data set name to the Db2 virtual table in a Data Virtualization Manager Server Event Facility (SEF) virtual table rule. With this access, you can issue SQL queries directly against Db2 unload data sets using existing Db2 virtual tables.

Switching a Db2 virtual table to read an unload data set is done by assigning a data set name to the table in a virtual table rule. The VTB variable **vtb.optbdsna** is used to redirect access from Db2 to reading the sequential file named in the variable. The named sequential file must contain unload data created by the Db2 UNLOAD utility. A model VTB rule, AVZMDLDU, is provided to demonstrate redirecting a Db2 virtual table to a Db2 unload data set.

As an example, consider a virtual table named DSNA_EMPLOYEES that maps the EMPLOYEES table in Db2 subsystem DSNA. By activating the model rule AVZMDLDU, you can query an unload sequential dataset named EMPLOYEE.UNLOAD.SEQ by issuing the following query:

```
SELECT * FROM MDLDU_DSNA_EMPLOYEES__EMPLOYEE_UNLOAD_SEQ
```

The AVZMDLDU rule performs the following steps:

1. Extracts the table name DSNA_EMPLOYEES and sets the VTB variable **vtb.optbmtna**.
2. Extracts the data set name EMPLOYEE_UNLOAD_SEQ, converts the underscores to periods, and sets the VTB variable **vtb.optbdsna**.

The following restrictions and considerations apply when using this feature:

- SQL access to Db2 unload files is limited to SQL queries only.
- The columns in Db2 virtual table definition must exactly match the table unloaded in Db2.

Use the following procedure to configure the sample rule AVZMDLDU.

Note: Sample rule AVZMDLDU is intended to be used as a model and may require customization. When customizing this rule, additional logic may need to be added if different unload data sets require different VTB variable settings for CCSID or internal/external format.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

2. Access the VTB rules, as follows:
 - a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt**.

- b) Specify option 2, **SEF Rule Management**.
 - c) Enter VTB for **Display Only the Ruleset Named**.
3. Customize the AVZMDLDU rule, as follows:
- a) Specify S next to AVZMDLDU to edit the rule.
 - b) Find the **vtb.optbdsna** variable and specify the name of the Db2 unload data set to process.
 - c) Update additional rule options as needed. The following table describes the VTB rule options that support Db2 unload data set access.

VTB variable	Description
vtb.optbd1cv	If the data was unloaded with a DELIMITED statement, set vtb.optbd1cv to 1 to declare the data is in delimited format. It may also be necessary to declare the delimiters if the default column delimiter (,) and character string delimiter (“”) were overridden when the data was unloaded.
vtb.optbdsna	Specifies the name of the sequential unload data set created by the Db2 UNLOAD utility to access.
vtb.optbduif	By default, the Db2 unload utility writes data in external format. If FORMAT INTERNAL is used when unloading data, vtb.optbduif must be set to 1 to declare that the data was unloaded in internal format.
vtb.optbmtna	Specifies the map name of the Db2 virtual table describing the unload file.
vtb.optbtbcc	If the table CCSID is not compatible with the CCSID defined for the SQL engine (AVZSIN00 SQLENGDFLTCCSID parameter), vtb.optbtbcc can be used to declare the CCSID of the data. This is particularly important for Unicode tables and tables containing GRAPHIC columns.

- d) Save your changes and exit the editor.
4. Enable the rule by specifying E next to AVZMDLDU and pressing Enter.
5. Set the rule to Auto-enable by specifying A next to AVZMDLDU and pressing Enter.
Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Db2 for z/OS data access methods

Db2 for z/OS data can be accessed by the Data Virtualization Manager server using different data access methods.

The following Db2 for z/OS data access methods are available:

- Traditional Db2 access. This method accesses Db2 data through traditional Db2 APIs. This access method allows for reading and writing of the data and provides transactional integrity.
- Db2 Direct. This method accesses Db2 data by reading the underlying Db2 VSAM linear data sets directly. This access method allows read-only access to the data and provides high performance, bulk data access.

The Db2 data access method is specified when creating virtual tables in the IBM Data Virtualization Manager studio for access to Db2 data.

The following topics provide more information about the Db2 for z/OS data access methods.

Using traditional Db2 access

Traditional Db2 access methods access Db2 data through APIs such as Distributed Relational Database Architecture (DRDA), Call Attachment Facility (CAF), and Resource Recovery Services attachment

facility (RRSAF). Using traditional Db2 access allows for reading and writing of the data and provides transactional integrity.

Traditional Db2 access methods provide MapReduce and Virtual Parallel Data support. MapReduce is an algorithm that enables the Data Virtualization Manager server to streamline how it accesses Db2 data, thereby reducing the processing time required to virtualize Db2 data. Statistics about the Db2 database are collected and stored within a metadata repository from which the SQL engine optimizes the MapReduce process.

In order to exploit MapReduce for Db2 when using traditional Db2 access, the Data Virtualization Manager server must collect information about the Db2 database. This information is collected using the **DRDARange** command and is stored within the Data Virtualization Manager server metadata repository.

Traditional Db2 access is used automatically when Db2 Direct access is not available.

Using Db2 Direct

Db2 Direct is a Data Virtualization Manager server access method that reads the data in the Db2 VSAM linear data sets directly instead of accessing the data through traditional Db2 APIs. Using Db2 Direct, large data pulls can be performed in service request block (SRB) mode, and MapReduce and Virtual Parallel Data features can be exploited without any prerequisite processing, such as the collection of statistics using the **DRDARange** command. Db2 Direct access provides a significant increase in performance and reduced elapsed time in processing analytical type queries.

Db2 Direct allows read-only access to the data. When using Db2 Direct, there is no locking involved when accessing the data, so updates may not be captured and deleted records may have been captured. Results from Db2 Direct queries may be out of sync with the current state of a Db2 table due to recent table updates not being flushed to the linear data sets.

Security is managed using Db2 table authorization.

Restrictions and considerations:

Consider the following points when using Db2 Direct:

- The Db2 subsystem hosting a Db2 table must be active when Db2 Direct-enabled tables are loaded or refreshed in the data server. The map build process requires Db2 system access to identify data set information in the Db2 system catalog.
- The Data Virtualization Manager server requires read access to the Db2 VSAM linear data sets. The linear data sets containing the Db2 rows must be available to the data server processing SQL requests for Db2 data. If the data sets are unavailable or archived, Db2 Direct will be disabled during map load or refresh for the virtual table.
- Virtual tables enabled for Db2 Direct must include all the columns defined in the base Db2 table. This is necessary because the columns describe the internal format of the Db2 data.
- If Db2 is not available or some other error occurs during map build or map refresh processing, Db2 Direct is automatically disabled for the table and a message is written to the trace log:

```
DB2 direct processing disabled for map map-name
```

- If Db2 Direct processing is disabled, processing will continue with traditional Db2 APIs when possible.
- To determine if Db2 Direct is active, the following messages appear in the server trace:
 - At startup and map refresh, the following message is issued:

```
DB2 direct processing enabled for map map-name
```

- When Db2 Direct is used in a query, the following message is issued:

```
Processing table map-name using DB2 direct
```

- If Db2 Direct table security is enabled, the Db2 subsystem must be available to check security at SQL query time.

- If Db2 Direct table security is disabled, unauthorized users who would normally receive a -551 SQLCODE attempting to access data through traditional APIs may gain access to Db2 data.
- Db2 Direct does not support tables with edit procedures or SQL statements containing joins, LOB columns, or key columns.
- If Db2 Direct security is disabled, the CCSIDs of table columns will be assumed based on the ENCODING_SCHEME (EBCDIC, Unicode, ASCII) of the table.

Configuring Db2 Direct

Configure Db2 Direct options or disable Db2 Direct.

Before you begin

Review the restrictions and considerations when using Db2 Direct. See [“Using Db2 Direct” on page 116](#).

About this task

By default, Db2 Direct is enabled in the Data Virtualization Manager server. Use the information in this topic to perform the following optional tasks:

- Disable the Db2 Direct feature for a virtual table by using a Virtual Table (VTB) rule.
- Define the VCATNAME for the Db2 system catalog tables (in the DSNDB06 database) by modifying the DEFINE DATABASE statement. The VCATNAME for system catalog tables is a system bootstrap value and is not available using the data discovery query. This task is required only in the following situations:
 - Access to system catalog tables using Db2 Direct is intended.
 - The VCATNAME for database DSNDB06 is different from the subsystem name.
- Configure Db2 Direct options, such as the number of pages to allocate for Db2 segment information, whether to enforce Db2 SQL table security authorizations, and disabling Db2 Direct for the server, by modifying server parameters.
- Specify what Db2 Direct information to display in the server trace by modifying server parameters.

Procedure

1. To disable the Db2 Direct feature for a virtual table, in a VTB rule, set the variable **OPTBDIDD** to 1. For additional information, see the generic sample rule AVZMDTBL.
2. To define the VCATNAME for the Db2 system catalog tables, perform the following steps:
 - a) Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization.
 - b) In the DEFINE DATABASE statement, use the SYSTEMVCAT parameter to define the VCATNAME for the system catalog tables, as shown in the following example:

```
"DEFINE DATABASE TYPE(MEMBER) "
      "NAME(DBA9) "
      "LOCATION(RSABCDE9) "
      "DDFSTATUS(ENABLE) "
      "PORT(3725) "
      "IPADDR(127.0.0.1) "
      "CCSID(37) "
      "APPLNAME(DBABCD2) "
      "SYSTEMVCAT(DXX9) "
      "IDLETIME(110) "
```

3. To modify server parameters, perform the following steps:
 - a) Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization.
 - b) Use the **MODIFY PARM** command to change a parameter value. For example, the following command disables Db2 Direct for the Data Virtualization Manager server:

"MODIFY PARM NAME(DISABLEDB2DIRECT) VALUE(YES) "

The parameters in the following tables are available for use with Db2 Direct.

Table 12. SQL parameters in group PRODSQL

Parameter name	Parameter description	Default value
DB2DIRECTSEGTLBPAGES	DB2-DIRECT SEGMENT TABLE PAGES Defines the number of 4K pages to be allocated for Db2 segment information. The default value is 8, which should be enough for most Db2 Direct queries. This parameter should only be changed if a query fails because the Db2 Direct segment table was exhausted.	8
DISABLEDB2DIRECT	DISABLE DB2-DIRECT PROCESSING Disables Db2 Direct processing in the server.	NO
DISABLEDB2DIRSEC	DISABLE DB2-DIRECT TABLE SECURITY Disables SQL table security checking when Db2 Direct is selected to process Db2 data. Disabling table security checking will allow access to Db2 data when the target Db2 subsystem is not active. Important: Unauthorized users who would normally receive a -551 SQLCODE attempting to access data through traditional APIs like DRDA may gain access to Db2 data.	NO

Table 13. SQL parameters in group PRODTRACE

Parameter name	Parameter description	Default value
TRACEDB2DIRSTATS	TRACE DB2-DIRECT STATISTICS Enables tracing of a summary report to the system trace after each Db2 Direct query. Included in the trace are statistics about read and point operation in the Db2 linear data set(s) processed.	NO
TRACEDB2DIOPEN	TRACE DB2-DIRECT OPEN CONTROL BLOCKS Enables tracing of control blocks created at the open of each linear data set for Db2 Direct processing.	NO

Table 13. SQL parameters in group PRODTRACE (continued)

Parameter name	Parameter description	Default value
TRACEDB2DIRSEGP	TRACE DB2-DIRECT SEGMENT PAGES Enables tracing if Db2 pages containing segmented map information.	NO
TRACEDB2DIRDICTP	TRACE DB2-DIRECT DICTIONARY PAGES Enables tracing of the compression dictionary used to compress and expand rows stored in Db2 linear data sets.	NO
TRACEDB2DIRDATAP	TRACE DB2-DIRECT DATA PAGES Enables tracing of data pages in a linear data set containing Db2 rows.	NO
TRACEDB2DIRROWS	TRACE DB2-DIRECT ROWS Enables tracing of rows extracted from data pages in a Db2 linear data set. If rows are compressed, an additional trace is created of the uncompressed row data.	NO

Configuring access to distributed databases

You can configure access to data on Big SQL, dashDB, DB2 LUW (Linux, UNIX, and Windows), Microsoft SQL Server, Oracle, and QMF DRDA.

Before you begin

If you are connecting to a Big SQL or DB2 LUW database, then you must install and configure the IBM DB2 Federated Server. For additional information, refer to the documentation.

If you are connecting to an Oracle database, then you must install and configure the Oracle Database Provider for DRDA. For additional information, refer to the documentation on the Oracle website.

If you are connecting to a 2016 Microsoft SQL Server database, then you must install and configure the Host Integration Server for HIS DRDA Service. Then install SYSIBM Views from Microsoft, use SQL Server IDE or command line to execute the script from Sample.txt file attached for SYSIBM Views in Microsoft. For additional information, refer to the documentation on the Microsoft website.

About this task

Configure access to distributed databases by modifying the configuration member, configuring Data Virtualization Manager Server Event Facility (SEF) rules, and optionally setting up alternate authentication information.

Procedure

1. [“Modifying the Data Virtualization Manager configuration member”](#) on page 120.
2. Configure the Data Virtualization Manager Server Event Facility rules and set up authentication for the appropriate database.

- [“Configuring rules and authentication for Big SQL” on page 125.](#)
- [“Configuring rules and authentication for dashDB” on page 126.](#)
- [“Configuring rules and authentication for LUW databases” on page 127.](#)
- [“Configuring rules and authentication for Microsoft SQL Server” on page 128.](#)
- [“Configuring rules and authentication for Oracle DRDA” on page 130.](#)
- [“Configuring rules and authentication for QMF DRDA Server” on page 131.](#)

Modifying the Data Virtualization Manager configuration member

Enable the RDBMS access method in the Data Virtualization Manager Data Virtualization Manager configuration member.

About this task

Configure the server to use Distributed Relational Database Architecture (DRDA) when accessing a RDBMS.

Modify the Data Virtualization Manager configuration member in data set *hlq.AVZS.SAVZEXEC(AVZSIN00)*. The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. Verify that the Unicode translation of the Coded Character Set Identifier (CCSID) used in the DEFINE DATABASE statement and the CCSID used by the target RDBMS are defined for your z/OS environment.
 - a) Identify the CCSID of the RDBMS.

For example, Oracle may use *ccsid1*. In your DEFINE DATABASE statement in the configuration member for the RDBMS you have *ccsid2*. For this example, where Oracle is using *ccsid1*, you need to verify that you have *ccsid1-ccsid2* and *ccsid2-ccsid1* defined in your Unicode translation table on z/OS using the command **D UNI,ALL**.

- b) If the entry is not present, add the entry to your Unicode translation table and refresh.

Refer to the IBM z/OS documentation on how to add the entry.

Note: As an alternative, the Unicode table can be appended within the server by using the following statement examples in the server configuration member:

```
"DEFINE CONV SOURCE(ccsid1) TARGET(ccsid2) TECH(RE)"
"DEFINE CONV SOURCE(ccsid2) TARGET(ccsid1) TECH(RE)"
```

2. In the AVZSIN00 member, locate the section that contains the comment Enable DRDA access to DB2 database subsystems.
3. Enable the DRDA parameters by changing the syntax `if DontDoThis` to `if DoThis` and then set the DRDASKIPZSERVICES parameter to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
do
  "MODIFY PARM NAME(TRACEOEDRDARW) VALUE(YES)"
  "MODIFY PARM NAME(CLIENTMUSTELECTDRDA) VALUE(NO)"
  "MODIFY PARM NAME(DRDASKIPWLMSETUP) VALUE(NO)"
  "MODIFY PARM NAME(DRDAFORLOGGINGTASK) VALUE(NO)"
  "MODIFY PARM NAME(DRDASKIPZSERVICES) VALUE(YES)"
```

The following table lists the parameters for configuring support for DRDA:

Parameter	Description	Valid values
CLIENTMUSTELECTDRDA	<p>If set to YES, the ODBC and JDBC clients must explicitly opt in for DRDA to be used by setting the user parameter connection variable to 'DRDA'.</p> <p>Note: ODBC and JDBC clients can always opt out of DRDA processing by setting the user parameter to 'NODRDA'.</p> <p>If set to NO, DRDA processing is used for access to all configured RDBMSs.</p>	<p>YES NO Default value.</p>
DRDAFORLOGGINGTASK	<p>If set to YES, DRDA processing is used for the Db2 on z/OS logging subtask.</p> <p>If set to NO, SAF or RRSF connections are used.</p> <p>Note: Passticket support must be enabled for the target DDF server. If passticket support is not configured, set the parameter to NO.</p>	<p>YES NO Default value.</p>
DRDASKIPWLMSETUP	<p>If set to YES, WLM information is not collected and sent to DRDA during ODBC/JDBC logon processing. If captured, the DRDA equivalent to SET_CLIENT_ID calls is issued after logon to establish these values on the DRDA connection. If not captured, the transmission that is used to set these WLM-related values is bypassed.</p> <p>If set to NO, the client user ID, application name, workstation name, and accounting token that were sent in the initial client buffer are collected and sent separately after logon processing to DRDA.</p>	<p>YES NO Default value.</p>

Parameter	Description	Valid values
DRDASKIPZSERVICES	Prevents DRDA from being used for z/Service Db2 processing. If set to YES, z/Services client tasks do not use DRDA processing for Db2 requests. If set to NO, DRDA will be used when configured for a particular Db2 connection. Note: Passticket support must be enabled for all target DDF servers.	YES NO Default value.
TRACEOEDRDARW	If set to YES (recommended), TCP/IP communications via DRDA are traced. If set to NO, DRDA receive and send operations are not traced.	YES NO Default value.

4. Define DRDA RDBMSs by entering a definition statement. Provide your local environment values for all the parameters. The following example shows the section in the configuration member to enable:

```
"DEFINE DATABASE TYPE(type_selection)"
      "NAME(name)"
      "LOCATION(location)"
      "DDFSTATUS(ENABLE)"
      "DOMAIN(your.domain.name)"
      "PORT(port)"
      "IPADDR(1.1.1.1)"
      "CCSID(37)"
      "APPLNAME(DSN1LU)"
      "IDLETIME(110)"
```

This is an example for dashDB:

```
"DEFINE DATABASE TYPE(DASHDB)"
      "NAME(name)"
      "LOCATION(location)"
      "AUTHTYPE(AES)"
      "SECMEC(EUSRIDPWD)"
      "DDFSTATUS(ENABLE)"
      "DOMAIN(your.domain.name)"
      "PORT(port)"
      "CCSID(37)"
```

The following table lists the parameters for defining DDF endpoints:

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. <i>(Optional)</i>	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. Note: APPLNAME is not required when connecting from the ODBC/JDBC driver.

Parameter	Description	Valid values
AUTHTYPE	<p>Authentication type. This can be either DES for Diffie Hellman Encryption Standard or AES for Advanced Encryption Standard.</p> <p>When AUTHTYPE is not supplied, the default is DES. To force AES, the option must be added to the DEFINE DATABASE statement. Each server can be different in what is supported as to AES/DES.</p>	<p>DES Diffie Hellman Encryption Standard (default value)</p> <p>AES Advanced Encryption Standard.</p>
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. (<i>Optional</i>)	Refer to the RDBMS vendor documentation for a list of valid CCSIDs.
DDFSTATUS	The DDF activation status can be altered online by using the ISPF 4-Db2 dialog panels. (<i>Required</i>)	<p>ENABLE Make this DDF definition active.</p> <p>DISABLE DDF endpoint is not used.</p>
DOMAIN	The part of a network address that identifies it as belonging to a particular domain.	No default value.
IDLETIME	If Db2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. (<i>Db2 for z/OS only</i>)	0-9999 seconds.
IPADDR	Specify the dot-notation IPV4 address of the DDF endpoint. (<i>Optional</i>)	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.

Parameter	Description	Valid values
LOCATION	<p>For Db2: The Db2 location name.</p> <p>For dashDB: This is the database name of the dashDB database or alias name for the database.</p> <p>For LUW: The LUW database.</p> <p>For Oracle: The Oracle SSID as defined to the Oracle Database Provider (Gateway).</p> <p><i>(Required)</i></p>	A valid value is a string 1 - 16 characters.
NAME	The database name as known to the server. <i>(Required)</i>	A valid value consists of 1 - 4 characters. Clients use this ID when they request access to a specific Db2 subsystem.
PORT	The TCP/IP port at which the server is listening. <i>(Required)</i>	A valid 1-5 numeric string.
SECMEC	<p>The DRDA security mechanism in force for standard dashDB services requires an authentication method setting. Define as either USRENCPWD, which informs the server to encrypt the PASSWORD or EUSRIDPWD, which informs the server to encrypt the USERID and PASSWORD during the initial connection to dashDB.</p> <p><i>(Except QMFDRDA)</i></p>	<p>USRENCPWD Encrypt password only.</p> <p>EUSRIDPWD Encrypt userid and password.</p>

Parameter	Description	Valid values
TYPE	<p>For distributed databases:</p> <p>BIGSQL DDF endpoint is a Big SQL engine.</p> <p>DASHDB DDF endpoint is a dashDB database.</p> <p>LUW DDF endpoint is a Db2 instance or group member for Linux, UNIX, or Windows.</p> <p>MSSQL DDF endpoint is a Db2 instance or group member for Microsoft SQL Server.</p> <p>ORACLE DDF endpoint is an Oracle instance. The parameter informs DRDA AR and supportive tooling that the remote server is an Oracle Database Provider which supports DRDA AS. The Oracle DRDA AS must be in z/OS simulation mode.</p> <p>QMFDRDA DDF endpoint is a QMF DRDA AS Object Server instance.</p>	<p>For distributed databases:</p> <p>BIGSQL</p> <p>DASHDB</p> <p>LUW</p> <p>MSSQL</p> <p>ORACLE</p> <p>QMFDRDA</p>

Configuring rules and authentication for Big SQL

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to Big SQL databases.

About this task

To complete configuration for access to Big SQL databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to Big SQL. By default, the server will attempt to log on to Big SQL with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing Big SQL.

This task uses the following tools:

AVZSBIGC

An SQL rule that allows Meta discovery on Big SQL databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEBIGG

An ATH rule that switches credentials when connecting to a Big SQL database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSBIGG) to allow IBM Data Virtualization Manager studio Meta discovery on Big SQL databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your Big SQL database, as instructed in the JCL, make sure to specify the correct DBTYPE. For Big SQL, specify DBTYPE=BIGSQL.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEBIGG) to provide the logon credentials to each Big SQL instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for dashDB

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to IBM dashDB databases.

About this task

To complete configuration for access to dashDB databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to dashDB. By default, the server will attempt to log on to dashDB with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing dashDB.

This task uses the following tools:

AVZSDDBC

An SQL rule that allows Meta discovery on dashDB databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEDDBG

An ATH rule that switches credentials when connecting to a dashDB database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSDDBC) to allow IBM Data Virtualization Manager studio Meta discovery on dashDB databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your dashDB database, as instructed in the JCL, make sure to specify the correct DBTYPE. For dashDB, specify DBTYPE=DASHDB.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEDDBG) to provide the logon credentials to each dashDB instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for LUW databases

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to LUW (Linux, UNIX, and Windows) databases, including databases connected via IBM Federated Server.

About this task

To complete configuration for access to LUW databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to LUW databases. By default, the server will attempt to log on to the LUW database with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing an LUW database.

This task uses the following tools:

AVZSLUWC

An SQL rule that allows Meta discovery on LUW databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZELUWG

An ATH rule that switches credentials when connecting to an LUW database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSLUWC) to allow IBM Data Virtualization Manager studio Meta discovery on LUW databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your LUW database, as instructed in the JCL, make sure to specify the correct DBTYPE. For LUW databases, specify DBTYPE=LUW.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZELUWG) to provide the logon credentials to each LUW instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for Microsoft SQL Server

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to Microsoft SQL Server via the 2016 Host Integration Server for HIS DRDA Service.

About this task

To complete configuration for access to Microsoft SQL Server, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to SQL Server. By default, the Data Virtualization Manager server will attempt to log on to SQL Server with the same user ID that was presented for logon to z/OS. A facility is provided in the Data Virtualization Manager server to optionally change the logon credentials for a user when accessing SQL Server.

This task uses the following tools:

AVZSMSSC

An SQL rule that allows Meta discovery on SQL Server databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEMSSG

An ATH rule that switches credentials when connecting to a SQL Server database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSMSSC) to allow IBM Data Virtualization Manager studio Meta discovery on SQL Server databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your Microsoft SQL Server database, as instructed in the JCL, make sure to specify the correct DBTYPE. For SQL Server databases, specify DBTYPE=MSSQL.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEMSSG) to provide the logon credentials to each SQL Server instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for Oracle DRDA

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to Oracle databases via the Oracle Database Provider for DRDA.

About this task

To complete the configuration for access to Oracle databases via the Oracle Database Provider for DRDA, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to Oracle AS. By default, the Data Virtualization Manager server will attempt to log on to Oracle with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing Oracle.

This task uses the following tools:

AVZSORAC

An SQL rule that allows Meta discovery on Oracle databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEORAG

An ATH rule that switches credentials when connecting to an Oracle database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSORAC) to allow IBM Data Virtualization Manager studio Meta discovery on Oracle databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your Oracle database, as instructed in the JCL, make sure to specify the correct DBTYPE. For Oracle, specify DBTYPE=ORACLE.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEORAG) to provide the logon credentials to each Oracle instance. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.

e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Configuring rules and authentication for QMF DRDA Server

Configure Data Virtualization Manager Server Event Facility (SEF) rules and set up authentication to provide access to QMF DRDA Server databases.

About this task

To complete the configuration for access to QMF DRDA Server databases, you must activate SEF rules and optionally set up authentication.

It is common for data centers to assign different user IDs for access to z/OS and for access to QMF DRDA Server. By default, the Data Virtualization Manager server will attempt to log on to QMF DRDA Server with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing QMF DRDA Server.

This task uses the following tools:

AVZSQMFC

An SQL rule that allows Meta discovery on Oracle databases.

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEQMFG

An ATH rule that switches credentials when connecting to a QMF DRDA Server database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

Procedure

1. Auto-enable the SQL rule SAVZXSQ(AVZSQMFC) to allow IBM Data Virtualization Manager studio Meta discovery on QMF DRDA Server databases.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or SQL to display only SQL rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is restarted.
2. Optional: To define alternate authentication information, use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq*.SAVZCNTL data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your QMF DRDA Server database, as instructed in the JCL, make sure to specify the correct DBTYPE. For QMF DRDA Server databases, specify DBTYPE=QMFRDA.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
3. Optional: If using alternate authentication information, auto-enable the SEF ATH rule SAVZXATH(AVZEQMFG) to provide the logon credentials to each QMF DRDA Server database. Global variables are used to define alternate authentication credential mapping for the SEF ATH Rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.

- c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Enable the rule by specifying E and pressing Enter.
 - e) Set the rule to Auto-Enable by specifying A and pressing Enter.
- Setting the rule to Auto-enable activates the rule automatically when the server is restarted.

Controlling display and access for native Db2 subsystems

You can control whether native Db2 database subsystems appear in ISPF and the Data Virtualization Manager studio and if attempts to connect to native Db2 subsystems are allowed.

About this task

The server parameter **DISABLEATTACH** controls whether native Db2 database subsystems appear in the ISPF and Data Virtualization Manager studio applications and if attempts to connect to native Db2 subsystems are allowed.

The following table describes the settings for this parameter:

Parameter	Description	Valid values
DISABLEATTACH	<p>Controls whether native Db2 database subsystems appear in the ISPF and Data Virtualization Manager studio applications and if attempts to connect to native Db2 subsystems are allowed.</p> <p>YES</p> <p>Only data sources defined as DRDA endpoints appear in the ISPF DB2 Interface Facility (Database Control) and the Data Virtualization Manager studio interface.</p> <p>An attempt to connect to a subsystem that does not have a DRDA configuration will be rejected. Trace Browse will show the following message:</p> <pre>DB SUBSYSTEM xxxx IS NOT DEFINED</pre> <p>For an attempt to connect to a DRDA data source that is disabled, Trace Browse will show the following message:</p> <pre>DB SUBSYSTEM xxxx IS NOT OPERATIONAL</pre> <p>NO</p> <p>(Default) All Db2 subsystems appear in the ISPF and Data Virtualization Manager studio interfaces.</p>	<p>YES</p> <p>NO</p>

The default setting for server parameter **DISABLEATTACH** is NO; however, the following statement is included in the server configuration file, which changes the setting to YES:

```
"MODIFY PARM NAME(DISABLEATTACH) VALUE(YES)"
```

If this override is omitted from the server configuration file, the setting will default to NO.

To review or update the **DISABLEATTACH** parameter setting, use the following procedure:

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization in the step Copying target libraries section.
2. Review the following statement in your AVZSIN00 member, and update the setting if necessary:

“MODIFY PARM NAME(DISABLEATTACH) VALUE(YES)”

Copying Alternate IDs for DRDA Data Sources Between DVM Servers

Configure the Data Virtualization Manager server to copy the alternate credentials for DRDA sources from one Data Virtualization Manager server to another.

About this task

When a Data Virtualization Manager server is cloned, the GLOBALU variables defined in the original server are not automatically transferred to the new server. To address this, batch jobs are available to unload and load the GLOBALU variables between Data Virtualization Manager server. These jobs first unload the GLOBALU variables from the original Data Virtualization Manager server into a sequential data set, and then load them into the new server. This process automates the transfer of user mappings between Data Virtualization Manager server.

If you need to copy the alternate USERID and passwords for DRDA sources from one Data Virtualization Manager server to another, you must execute these batch jobs to transfer the credentials.

To accomplish this, the following utilities are provided:

AVZUMUNL

This utility transfers the user mappings from the GLOBALU table of the original Data Virtualization Manager server to a sequential file.

The following table lists the parameters:

Parameters	Description
SS	Indicates the system ID of the Data Virtualization Manager server.
USRDATA	Indicates the sequential file where the data will be unloaded.
DBTYPE	Database type. Indicates the type of the DRDA database to copy. Use asterisks (*) to copy all.
DBNAME	Indicates the DRDA sub-system identifier. Use asterisks (*) to copy all.
OPT	Indicates the trace option for debugging.

AVZUMCPY

This utility transfers the user mappings from an unloaded sequential file to the cloned Data Virtualization Manager server.

The following table lists the parameters:

Parameters	Description
SS	Indicates the system ID of the Data Virtualization Manager server.
USRDATA	Indicates the sequential file where the data is unloaded.

OPT	Indicates the trace option for debugging.
-----	---

Configuring access to IMS databases

To access an IMS database, you need to configure the server started task JCL and the Data Virtualization Manager configuration member.

Before you begin

The server must already be installed.

About this task

IBM Data Virtualization Manager for z/OS provides seamless, real-time controlled access to IMS database data. It allows ODBC, JDBC, Data Virtualization Manager client, and web clients to access IMS database data in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to an IMS database. The result is a relational result set with no host programming required.

For a key field that allows using either an index (HIDAM) or a randomizer (HDAM), IBM Data Virtualization Manager for z/OS uses DBCTL. For unkeyed fields, IBM Data Virtualization Manager for z/OS uses IMS Direct as both IMS Direct and IMS DBCTL need to scan the data.

Procedure

To configure and verify access to data in an IMS database, complete the following tasks.

Configuring the server started task JCL

Add IMS.SDFSRESL to the server started task JCL.

Before you begin

All LOAD library data sets allocated to the Data Virtualization Manager server in the server started task JCL must be APF-authorized.

About this task

You can omit this task if the IMS resident library (SDFSRESL) module is in the z/OS linklist.

Procedure

Modify the server started task JCL. If the IMS SDFSRESL is not already in the link pack area or linklist, add it to the STEPLIB.

Installing virtual table and virtual view maps for IMS catalog access

Install virtual table and virtual view maps for IMS catalog access.

About this task

The IBM Data Virtualization Manager for z/OS studio reads the IMS catalog using the virtual tables and views installed as part of server set up. The following maps are distributed in XMIT format in the SAVZSAMP member AVZICMPD:

ZIMCDX00-ZIMXD000

IMS Catalog DBD maps for DBDs DFSCX000 and DFSCD000.

ZIMCP000

IMS Catalog PSB map for PSB DFSCP000.

ZIMCT001-ZIMCT015

The IMS virtual tables that map segments in the IMS catalog database DFSCD000.

ZIMCV001-ZIMCV010

The virtual views on the IMS catalog virtual tables used by the studio interface to the IMS catalog.

These maps are not installed by default. Use the following procedure to install these maps.

Procedure

1. Locate the AVZICMPS member in the *hlq.SAVZCNTL* data set.
2. Modify the JCL according to the instructions provided in the AVZIAMPS member.
3. Submit the job.

The virtual tables and virtual target maps are installed.

Configuring ODBA Support

You can configure the IBM Data Virtualization Manager for z/OS to support the Open Database Access (ODBA) interface. Perform the following tasks to enable ODBA support.

Procedure

1. Turn off IMS DBCTL support in the IN00.
2. Under **Enable IMS CCTL/DBCTL support**, choose **DontDoThis**. This sets the value of the parameter DBCTL to **NO**.
3. Add the following to the IN00:

```
/*-----*/
/* Enable IMS ODBA support                               */
/*-----*/
if DoThis then
do
  "MODIFY PARM NAME(IMSODBA)           VALUE(YES) "
  "MODIFY PARM NAME(RRS)              VALUE(YES) "
  "MODIFY PARM NAME(MAXODBACONNECT)   VALUE(12) "
  "MODIFY PARM NAME(TRACEIMSDLIEVENTS) VALUE(NO) "
  "MODIFY PARM NAME(IMSID)            VALUE(IMS1) "

  "DEFINE IMSODBA",
    "NAME(IMS1) ",
    "CONTROLREGIONID(IMS1) ",
    "MAXTHREADS(10) ",
    "MINTHREADS(3) ",
    "USERID(AVZ9) "
end
```

4. Confirm that the virtual table uses **IMS/DBCTL** on Studio
5. If IMS is a FASTPATH DB, add the following.

```
"FASTPATHNBA(100) ",
"FASTPATHOVERFLOW(100) ",
"FASTPATHALLOCATED(100) ",
```

Modifying the DBCTL configuration member

Enable the IMS database control (DBCTL) parameters in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment “Enable IMS CCTL/DBCTL support.”
2. Enable the IMS DB parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the parameter DBCTL to YES. The following example shows the section in the configuration member to enable:

```
if DoThis then
do
"MODIFY PARM NAME(DBCTL)           VALUE(YES) "
"MODIFY PARM NAME(IMSID)           VALUE(IVP1) "
"MODIFY PARM NAME(IMSIDSNAME)      VALUE(IMSX10.SFDSRESL) "
"MODIFY PARM NAME(IMSINTHREADS)    VALUE(5) "
"MODIFY PARM NAME(IMSMAKTHREADS)   VALUE(10) "
"MODIFY PARM NAME(IMSMBABUFFERS)   VALUE(0) "
"MODIFY PARM NAME(IMSFPBUFFERS)    VALUE(0) "
"MODIFY PARM NAME(IMSFPFLOW)       VALUE(0) "
"MODIFY PARM NAME(TRACEIMSDLIEVENTS) VALUE(NO) "
```

The following table lists the parameters for configuring support for IMS DB data stores:

Parameter	Description	Valid values
DBCTL	Initialize DBCTL support.	YES NO (default value)
IMSID	IMS SSID of the DBCTL region.	Four-character name
IMSIDSNAME	The name of the data set for the IMS residence library.	Data set name
IMSINTHREADS	Minimum number of threads.	Numeric value. Default is 5.
IMSMAKTHREADS	Maximum number of threads.	Numeric value. Default is 10.
IMSMBABUFFERS	Total number of NBA buffers.	Numeric value. Default is 0.
IMSFPBUFFERS	Fast path buffers per thread.	Numeric value. Default is 0.
IMSFPFLOW	Fast path overflow buffers.	Numeric value. Default is 0.
TRACEIMSDLIEVENTS	Trace IMS DLI events.	YES NO (default value)

Modifying the IMS Direct configuration member

Enable and configure the IMS Direct parameters in the Data Virtualization Manager configuration member.

About this task

The IMS Direct feature provides map reduce and parallelism support for accessing native IMS files. This support bypasses the requirement of having to use native IMS API calls by reading the IMS database files directly, similar to how an unload utility may work. This method provides a significant improvement in performance and reduced elapsed time in processing analytical type queries.

When an IMS SQL query is run, the SQL engine for the server will determine if the request is best executed using IMS Direct (native file support) or if IMS APIs are required. The determination is based on database and file types supported as well as the size of the database. Virtual tables of the IMS segments are required.

The following types of IMS databases are currently supported by IMS Direct:

- Hierarchical direct access method (HDAM) - VSAM and OSAM
- Hierarchical indexed direct access method (HIDAM) - VSAM and OSAM
- Partitioned HDAM (PHDAM) - VSAM and OSAM
- Partitioned HIDAM (PHIDAM) - VSAM and OSAM
- Fast Path data entry database (DEDB)

When using IMS Direct, there is no locking involved when accessing the data, so updates may not be captured and deleted records may have been captured. Security is managed on the IMS native data set itself when IMS Direct is used. The user ID of the client connection must have the necessary security permissions for reading the IMS database data set(s).

IMS Direct supports access to multiple IMS subsystems and calls to compression exits and Guardium encryption and decryption exits.

Using exits

If you use compression exits or Guardium encryption and decryption exits, you can configure the server to call these exits, providing optimization.

For compression exits, the default mode of operation is to call them in TCB mode with a serialization latch held and a PST address of 0. This can be inefficient since most of the IMS Direct processing takes place in SRB mode on a zIIP. If you know enough about your compression exit, you can optimize performance of the exit by specifying it in either the `IMSDIRCMPTCB n` , or `IMSDIRCMPSRB n` statements, which are described in the procedure below. All exits are called for INIT and TERM in TCB mode.

- Decompression calls may be made in TCB mode, without serialization by specifying the name in an `IMSDIRCMPTCB n` statement. This will allow parallel threads to run without serialization, improving performance.
- Decompression calls may also be made in SRB mode, without serialization, by specifying the name in an `IMSDIRCMPSRB n` statement. This will avoid a task switch for each compressed segment, improving performance. Note that the supplied IMS compression `DFSCMPX0` exits and `DFSKMPX0` will run in SRB mode.

Guardium decryption exits require a PST and PST work area. A dummy PST with a PST work area is passed to these exits when they are specified in an `IMSDIRDECXIT n` statement, which is described in the procedure. Guardium decryption exits can run in SRB mode, without serialization.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member `hlq.SAVZEXEC(AVZSIN00)` and may have been copied to a new data set for customization.
2. In the AVZSIN00 member, locate the comment "Enable IMS Direct Map Reduce."

3. Enable the IMS Direct parameters by changing the syntax `if DontDoThis` to `if DoThis`, and then set the parameter `IMSDIRECTENABLED` to `YES`. The following example shows the section in the configuration member to enable:

```

if DoThis then
do
"MODIFY PARM NAME(IMSDIRECTENABLED) VALUE(YES)"
"MODIFY PARM NAME(IMSDIRECTBUFFERSIZE) VALUE(1024)"
"MODIFY PARM NAME(ACIINTSEGMP256) VALUE(200)"
"MODIFY PARM NAME(TRACEIMSDDBREFRESH) VALUE(YES)"
"MODIFY PARM NAME(TRACEIMSDIRSTATS) VALUE(YES)"

"DEFINE IMSDBINFO",
.
.
.
end

```

The following table lists the parameters for configuring support for IMS Direct:

Parameter	Description	Valid values
ACIINTSEGMP256	The 256K ACI buffer pool. Required for IMS Direct.	Numeric value. Default is 200.
IMSDIRECTBUFFERSIZE	Specified in KB, and should be greater than the size of the largest complete IMS database record (root + all dependent segments).	Numeric value.
IMSDIRECTENABLED	Enable IMS Direct support.	YES NO (default value)
TRACEIMSDDBREFRESH	Generate trace message when IMS Direct map reduce discovery processing is performed.	YES NO (default value)
TRACEIMSDIRSTATS	Produce runtime statistics at the end of IMS Direct processing of a data set.	YES NO (default value)

4. Define your IMS subsystem using the `DEFINE IMSDBINFO` statement. Provide one statement for each IMS subsystem that will be used by IMS Direct.

```

"DEFINE IMSDBINFO",
"IMSID(xxxx)",
"SUFFIX(x)",
"MODBLKS(your.MODBLKS)",
"ACBLIB(your.ACBLIB)",
"DFSRESLB(your.SDFSRESL)",
"IMSDALIB(your.dynamic.allocation.lib)",
"RECON1(your.RECON1)",
"RECON2(your.RECON2)",
"RECON3(your.RECON3)"
end

```

The following table lists the parameters used to define the IMS database:

Parameter	Description	Valid values
IMSID	The IMS subsystem identification.	Up to 4-character ID.

Parameter	Description	Valid values
SUFFIX	The setting of the SUF= keyword used in the IMS Control Region.	One character. Default value is I.
ACBLIB	ACBLIB data sets contain the application control blocks (ACBs), which describe IMS applications, and data management blocks (DMBs), which describe databases and the applications that can access them.	your.ACBLIB
DFSRESLB	Load library that contains the major IMS modules.	your.SDFSRESL
IMSDALIB	Dynamic Allocation Library for IMSDBs and RECONS.	your.dynamic.allocation.lib
MODBLKS	Used to support dynamic resource definition. Contains the APPLCTN, DATABASE, RTCODE, and TRANSACT macros.	your.MODBLKS
RECON1	Primary RECONciliation dataset, which holds all of the resource information and event tracking information that is used by IMS.	your.RECON1
RECON2	An active copy of RECON1.	your.RECON2
RECON3	Spare RECON to be used when RECON1 or RECON2 are not useable.	your.RECON3

5. (Optional) Add the following statements to configure additional IMS Direct parameters:

```
"MODIFY PARM NAME(IMSDIRECTCYLBUF) VALUE(3)"
"MODIFY PARM NAME(IMSDIRECTOSAMRECSR) VALUE(2)"
```

Parameter	Description	Valid values
IMSDIRECTCYLBUF	Specifies the number of cylinders of data to buffer for each file processed in an IMS Direct task.	1-50. Default value is 3.
IMSDIRECTOSAMRECSR	Specifies the number of records to read in each OSAM I/O operation. For random reads, a large number may lead to unnecessary blocks read. For sequential reads, small numbers may give decreased performance.	1-50. Default value is 2.

6. To call a compression exit, perform one of the following steps as appropriate:

- If your compression exit must be called in TCB mode but can run properly without serialization, specify your exit name in the following statement:

```
"MODIFY PARM NAME(IMSDIRCMPIITTCBn) VALUE(exitname)"
```

where n is a number from 1 to 10 and *exitname* is the name of the compression exit routine.

- If your exit can run properly in SRB mode without serialization, specify your exit name in the following statement:

```
"MODIFY PARM NAME(IMSDIRCMPXITSRBn) VALUE(exitname)"
```

where n is a number from 1 to 10 and *exitname* is the name of the compression exit routine.

If neither of these conditions apply, do not specify the name of your compression exit.

Note: Review ["Using exits"](#) for more information about configuring calls to compression exits.

Parameter	Description	Valid values
IMSDIRCMPXITTCBn	Specifies the name of a compression exit that can be safely called without serialization. Up to 10 exit names can be specified, where n is a number from 1 to 10. Since the server runs multiple threads in parallel, this feature provides optimization by eliminating the possible serialization conflicts between threads.	Name of compression exit routine
IMSDIRCMPXITSRBn	Specifies the name of a compression exit that can be safely called without serialization and in SRB mode. Up to 10 exit names can be specified, where n is a number from 1 to 10. Since multiple exit names can be called without serialization and without switching off the zIIP (SRB mode) into TCB mode (GP processor), this feature provides optimization by eliminating the need to switch tasks for each exit call. The IBM supplied compression exits DFSCMPX0 and DFSKMPX0 will run safely in SRB mode. They can be specified in IMSDIRCMPXITSRB1 and IMSDIRCMPXITSRB2.	Name of compression exit routine

7. To call Guardium encryption and decryption exits, add the following statement:

```
"MODIFY PARM NAME(IMSDIRDECXITSRBn) VALUE(exitname)"
```

where n is a number from 1 to 20 and *exitname* is the name of the Guardium exit routine.

Note: Review ["Using exits"](#) for more information about configuring calls to Guardium encryption and decryption exits.

Parameter	Description	Valid values
IMSDIRDECXITSRBnn	Specifies the name of the Guardium encryption and decryption exit routine. Up to 20 exit names can be specified, where <i>nn</i> is a value from 1 to 20.	Name of Guardium exit routine

8. To change the default security environment to the server address space id from the connected user id, perform the following:

- Set the **IMSDIRDSAUTHUSER** parameter to SERVER.
- Set the **IMSPSBSECURITY** parameter to YES.
- Set the **IMSRCLASS** parameter to IMS.

Parameter	Description	Valid values
IMSDIRDSAUTHUSER	IMS-DIRECT DATASET AUTHORIZATION.	<ul style="list-style-type: none"> • User: IMS direct datasets are opened for IMS direct processing under user context. • SERVER: Setting this value opens the datasets under the server address space. • CLIENT: Opens the datasets under the user id of the connected client
IMSRCLASS	IMS RCLASS VALUE. The IMSRCLASS parameter passes the IMS resource class to the product. This parameter value is specified using the RCLASS keyword of the SECURITY macro in the stage 1 gen input. This value must be one to seven alphanumeric characters. Note that this value can also be specified using the DFSDCxxx member of PROCLIB. Any value specified using the DFSDCxxx member of PROCLIB will override the SECURITY macro value.	IMS
IMSPSBSECURITY	CHECK IMS PSB USER ACCESS. For DBCTL, the IMSPSBSECURITY parameter controls whether a SAF call is made before each PSB is scheduled. For IMS direct, the SAF call will be issued for the PSB name. This parameter does not apply to ODBA.	YES: SAF call is executed to verify that the current user is authorized to access the PSB. NO: no SAF call is executed.

Configuring access to IBM MQ

For access to IBM MQ (MQ) data, you must modify the server started task, configure the Data Virtualization Manager configuration member, and set virtual table options.

Data Virtualization Manager provides SQL-only query access to MQ queues using virtual tables. Data in MQ queues is described using COBOL or PLI data descriptions taken from copybooks or programs.

IBM MQ for z/OS Versions 7.5 and newer are supported.

Note: Server configuration parameters control MQ tracing and can be modified if necessary.

Configuring the server started task JCL

Modify the server started task JCL to access IBM MQ data.

Before you begin

All data sets that you add to the server started task JCL STEPLIB must be APF-authorized.

About this task

Modify the server started task JCL to access IBM MQ data. You can skip this task if the IBM MQ load module is in the z/OS linklist or link pack area.

Procedure

Add the IBM MQ load library to the server started task JCL STEPLIB.

Modifying the Data Virtualization Manager configuration member for IBM MQ

To enable support for MQ data, you must update your Data Virtualization Manager server configuration file.

About this task

To be able to access MQ data in virtual tables, enable the feature in the server configuration file, as described in the following procedure.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization.
2. Add the following statement to your AVZSIN00 member:

The following table describes this parameter:

Parameter	Description	Valid values
MQACTIVE	Initialize IBM MQ support. This parameter must be set to YES to access MQ queues.	YES NO (default value)

Configuring virtual table rules for IBM MQ

Configure Data Virtualization Manager Server Event Facility (SEF) rules to support IBM MQ data.

About this task

You can configure VTB rule options to control the MQ data access feature. These options control inclusion of the MQ message descriptor meta data fields in the virtual tables, how to handle truncated messages, and whether to perform destructive reads. Sample VTB rule AVZMDLMQ documents these settings.

When accessing MQ data with sample rule AVZMDLMQ (or equivalent options) enabled, tables prefixed with MDLMQ_* are filtered, and the map name is extracted by removing the MDLMQ_ prefix. For example, the following query will execute the rule and query virtual table MQ_CSQ7_TRADE:

```
SELECT * FROM MDLMQ_MQ_CSQ7_TRADE
```

Use the following procedure to configure the sample rule AVZMDLMQ.

Note: Sample rule AVZMDLMQ is intended to be used as a model and may require customization. When customizing this rule, additional logic may need to be added if different VTB variable settings are required for different MQ queues.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

2. Access the VTB rules, as follows:
 - a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
 - b) Specify option 2, **SEF Rule Management.**
 - c) Enter VTB for **Display Only the Ruleset Named.**
3. Customize the AVZMDLMQ rule, as follows:
 - a) Specify S next to AVZMDLMQ to edit the rule.
 - b) Update the rule options as needed. The following table describes the VTB rule options that support MQ data access.

VTB variable	Description	Valid values
vtb.optbmqdg	Delete messages during retrieval. When set to 1, SQL queries will remove messages from the queue if ALL messages in the queue are successfully retrieved by the server. Retrieval of MQ messages will use non-browse (destructive) MQGET calls with syncpoint control. Once all messages are delivered to the server, they will be deleted from the queue. If a failure occurs before all messages are retrieved, an MQBACK call will be issued to restore messages to the queue that have been retrieved so far. Note that an MQCMIT will be issued and messages deleted if the IBM MQ syncpoint limit is reached. A failure after MQCMIT will not be able to restore messages as they have been permanently deleted.	0 (Default) 1
vtb.optbmqim	When set to 1 for an MQ virtual table, the MQ Series Message Descriptor (MQMD) meta data fields will be added to the virtual table as columns and returned with each result row. These columns are prefixed with the value MQMD_.	0 (Default) 1
vtb.optbmqt c	By default, a truncation error reading an IBM MQ message will result in a query failure. When set to 1, MQ Series access ignores truncated message warnings and returns data received.	0 (Default) 1

- c) Save your changes and exit the editor.
4. Enable the rule by specifying E next to AVZMDLMQ and pressing Enter.
 5. Set the rule to Auto-enable by specifying A next to AVZMDLMQ and pressing Enter.

Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring access to native VSAM

No modifications are required to configure the SQL interface for native VSAM. However, you should verify that the server has access to native VSAM. Optionally, you can control the data buffer (BUFND) and the index buffer (BUFNI) values for VSAM files either globally or for individual requests.

Before you begin

The server must already be installed.

About this task

The SQL interface for native VSAM provides seamless, real-time controlled access to native VSAM data. It allows ODBC, JDBC, and web clients to access native VSAM data in a relational model using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to native VSAM. The result is a relational result set, with no host programming required.

You can use a post-read exit routine in the member AVZGNASB in the hlq.SAVZSAMP module to improve performance by limiting the number of records processed by IBM Data Virtualization Manager for z/OS.

Verifying access to native VSAM

Verify native VSAM data access by creating a sample VSAM file and a corresponding virtual table and running a query that accesses the VSAM data.

Procedure

1. Create the sample VSAM file on the mainframe that hosts the Data Virtualization Manager server.
Run the AVZGNSTF member in the *hlq.SAVZCNTL* data set to allocate and load the sample VSAM file.
The job should complete with a condition code of 0.
2. Create the *staffvs* virtual table, and run a query that returns a result set.
Run the AVZIVVS1 member in the *hlq.SAVZCNTL* data set to perform a batch extract of the sample VSAM file listing and create a virtual table that is used to format the result set that is returned from the VSAM file.
The job should complete with a condition code of 0.
3. Verify that the SQL results contained in the AVZIVVS1 member are valid.

Modifying the data and index buffers for VSAM files

You can change the data and index buffers for VSAM files.

About this task

To control the settings of the data and index buffers for VSAM files, you must add the required parameters to your Data Virtualization Manager server configuration file.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization in the step Copying target libraries section.
2. Add the following statements to your AVZSIN00 member:

```
"MODIFY PARM NAME(SQLENGVSAMDATABUFF) VALUE(20)"
"MODIFY PARM NAME(SQLENGVSAMINDEXBUFF) VALUE(30)"
```

The following table lists these parameters:

Parameter	Description	Valid values
SQLENGVSAMDATABUFF	Specifies the number of data buffers for VSAM files. Default: 20	Numeric value.
SQLENGVSAMINDEXBUFF	Specifies the number of index buffer for VSAM files. Default: 30	Numeric value.

VSAM record-level sharing

VSAM Record-Level Sharing (RLS) enables data sharing across multiple systems in a parallel Sysplex. RLS is an access mode for VSAM files to enable the VSAM data to be shared, with full update capability, between many applications.

If RLS is not enabled for a VSAM file, you can only invoke the SELECT statement on the VSAM file. To enable you to invoke INSERT, UPDATE, and DELETE statements for VSAM data, you need to enable RLS on the VSAM file. For more information on VSAM RLS, see [Administering VSAM record-level sharing and Using VSAM Record-Level Sharing](#).

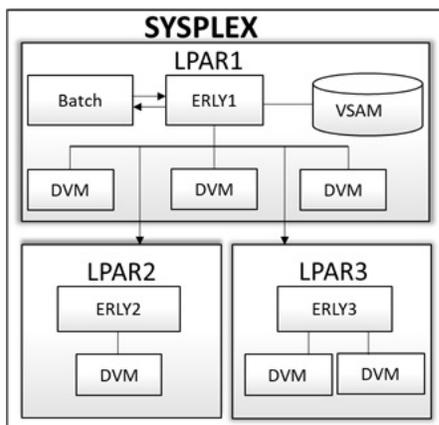
When you enable VSAM RLS, configure the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

Parameter name	Parameter description
VSAMCONTROLFILE	Specifies the data set name of the control file for VSAM files. This file is shared among all copies of the product in the Sysplex. If this parameter is enabled, VSAMCONTROLAIX parameter must also be specified.
VSAMCONTROLAIX	Specifies the path name of the control file for VSAM files. This file is shared among all copies of the product in the Sysplex. This parameter must be enabled if the VSAMCONTROLFILE parameter is specified.

VSAM File Control and ERLY Servers

VSAM File Control is an administration facility. VSAM File Control enables the users to manage the VSAM datasets accessed by the Data Virtualization Manager server and only applies to the Data Virtualization Manager servers using VSAM Record Level Sharing (VSAMRLS). This facility uses the cross-memory services and TCP/IP services to administer and manage the VSAM datasets within and across the SYSPLEX.

The user must define an ERLY server, which acts like as a Manager server, to cater to all the file management requests triggered from the Data Virtualization Manager data servers within the LPAR. The following diagram shows a high-level architecture for VSAM File Control. The Figure here illustrates a SYSPLEX with three logical partitions (LPAR1, LPAR2, and LPAR3).



Here,

- The Batch utility runs on LPAR1 and calls ERLY1 using cross-memory and passes commands.
 - Note:** The batch utility can run on any LPAR in the sysplex.
- ERLY1 updates the VSAM file used for local File Management (FM) and communicates the FM commands to each active Data Virtualization Manager copy on the LPAR1.
- ERLY1 then reads the File Management (FM) file to identify other LPARs in the SYSPLEX and forwards the commands via TCP/IP to the LPAR Listeners.
- ERLY2 and ERLY3 communicates the FM commands using cross-memory to each active Data Virtualization Manager copy on LPAR2 and LPAR3. Finally, all the responses flow back to ERLY1 and are consumed by the Batch utility.

The ERLY Server reserves two TCP/IP ports for its functions, the ports are namely Sysplex Listener Port and LPAR Listener Port. Sysplex Listener Port receives all the File Management commands across the Sysplex, while the LPAR Listener Port receives all the File Management commands from other LPARs within the Sysplex.

Configuring VSAM Control Dataset

This task provides step-by-step guidelines to define the VSAM control dataset.

About this task

The VSAM File Control consists of several parts. The ERLY servers are defined to handle the communications between the LPARs. A control file is shared with all the LPARs in the SYSPLEX. Each ERLY server performs only File Control requests. It does not process customer transactions. You must start the server during or near in time to IPL of the LPAR and before any customer servers start. One ERLY subsystem must be defined on each LPAR. The TCP configuration will define an IP name and address for each ERLY subsystem. One LPAR is defined as the primary SYSPLEX server

Procedure

1. To define cluster, define the following statements:

```
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER(NAME(<<VSAM CONTROL DATASET NAME>>) -
  TRACKS(15 15) SHR(2 3) LOG(NONE)-
  STORCLAS(<<SMS STORAGE CLASS>>) -
  DATACLAS(<<SMS DATA CLASS>>) -
  INDEXED RECSZ(100 32700) KEYS(57 0) )
/*
```

Use DEFINE CLUSTER to define attributes for the cluster as a whole and for the components of the cluster. The following table defines the command's parameters:

Parameters	Description
NAME	Specifies the name of the cluster
TRACKS	Specifies the number of tracks allocated for the cluster
STORCLAS	Specifies an installation-defined name of a storage class
DATACLAS	Specifies an installation-defined name of a data class
INDEXED	Specifies that the cluster is a VSAM KSDS (key sequenced)

2. To define an alternate index, define the following statements:

```
//DEFAIX EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE ALTERNATEINDEX (NAME(<<ALTERNATE INDEX>>) -
    RELATE(<<VSAM CONTROL DATASET>>) -
    CYLINDERS(2 2) -
    VOLUMES(*) -
    KEYS(10 47) -
    RECSZ(200 32700) -
    DATACLAS(<<DATACLAS>>) -
    NONUNIQUEKEY -
    UPGRADE)
  DEFINE PATH (NAME(<<ALTERNATE INDEX PATH>>) -
    PATHENTRY(<<ALTERNATE INDEX>>))
  BLDINDEX INDATASET(<<VSAM CONTROL DATASET>>) -
    OUTDATASET(<<ALTERNATE INDEX PATH>>)
/*
```

Use DEFINE ALTERNATEINDEX command to create an alternate index entry and an index entry to define the alternate index. The following table defines the command's parameters:

Parameters	Description
RELATE	Identifies the alternate index's base cluster.
CYLINDERS	Allocates two cylinders for the alternate index's space. The alternate index is extended in increments of two cylinder.
VOLUMES	Indicates that the alternate index is to reside on volume .
KEYS	Specifies the length and location of the alternate key field in each of the base cluster's data records.
DATACLAS	Specifies an installation-defined name of a data class.

Configuring the SYSPLEX ERLY Server

This task provides guidelines to configure the SYSPLEX ERLY Server.

About this task

After defining the cluster and alternate index, you can configure the SYSPLEX ERLY Server. One ERLY subsystem must be defined on each LPAR.

Procedure

1. Edit PROC JCL in h1q. .SAVZCNTL (AVZPEDBY) and replace EDBY with xVZy.

2. Enter the following statements to define ERLY server in IN00 file:

```

/* ENABLING VSAM FILE CONTROL UTILITY                                */
/*-----*/
  if 1 == 1 then
  do
  "MODIFY PARM NAME(ERLY) VALUE(YES)"
  "MODIFY PARM NAME(VSAMCONTROLFILE) VALUE(<Control_Dataset_Name>)"
  "MODIFY PARM NAME(VSAMCONTROLAIX) VALUE(<Control_Path_Name>)"
  end

```

For Non-ERLY server in IN00, enter the following statements:

```

/*-----*/
/* UTILIZE VSAM FILE CONTROL                                        */
/*-----*/
"MODIFY PARM NAME(VSAMCONTROLFILE) VALUE (<<VSAM CONTROL DATASET>>)"
"MODIFY PARM NAME(VSAMCONTROLAIX)
  VALUE (<<VSAM PATH FOR THE ALTERNATE INDEX >>)"
/*-----*/

```

The following table lists these parameters:

Parameter	Description	Valid values
ERLY	Defines the server as an ERLY server	<ul style="list-style-type: none"> • Yes for ERLY and • No for Non-ERLY
VSAMCONTROLFILE	Specifies the data set name of the control file for VSAM files. This file is shared among all copies of the product in the Sysplex.	
VSAMCONTROLAIX	Specifies the path name of the control file for VSAM files. This file is shared among all copies of the product in the Sysplex.	

3. Run the security job –

- RACF: h1q..SAVZCNTL (AVZRAEDB)

4. Start the Data Virtualization Manager ERLY Server.

The following message displays to indicate that the server is a Data Virtualization Manager ERLY Server:

ERL3260I SubSystem <AVZ SSID> is an ERLY SubSystem.

VSAM File Manager Commands

A single VSAM file is shared between the Manager LPAR and the Worker LPAR. You must define all the jobs in the VSAM control file and provide a sample of the ALT index.

Commands to administer VSAM dataset

To define, modify, and delete LPAR and SYSPLEX dataset in h1q.SAVZLOAD (AVZVSFCT) use the following syntaxes:

Syntax to define the SYSPLEX name:

```
DEFINE | SYSPLEX NAME(sysplex) | LPAR (lpar-name) | - IP(IP-address) | PORT(port-name)
```

Keyword	Description
SYSPLEX NAME	Defines the name of the SYSPLEX. This keyword is mandatory.
LPAR	Name of the LPAR(s) that resides on this SYSPLEX. You can define 4000 LPARs. This keyword is mandatory. For multiple LPARs, the LPAR names are delimited by one or more spaces (For example- LPAR (LP1 LP2 LP3))
IP	This is the TCP/IP name of the SYSPLEX being defined. This keyword is mandatory.
PORT	Defines the port number used by the ERLY subsystem on which the SYSPLEX is defined. This keyword is mandatory.

Syntax to define the LPAR name:

```
DEFINE | LPAR NAME(lpar-name) | - IP(IP-address) | PORT(port-name) | - GROUP(group-name) |
```

Keyword	Description
LPAR	Name of the LPAR(s) that resides on this SYSPLEX. You can define to the limit of 4000 LPARs within the SYSPLEX. This keyword is mandatory.
IPADDRESS	This is the TCP/IP name of the SYSPLEX that you want to define. This keyword is mandatory.
PORT	Defines the port number used by the ERLY subsystem on which the SYSPLEX is defined. This keyword is mandatory.
GROUPS	The names of all the group(s) that resides on this LPAR. You can define up to 4000 groups. This keyword is mandatory. For multiple GROUPs, the GROUP names are delimited by one or more spaces (For example- GROUP (G1 G2 G3))

Note:

- If DEFINE LPAR statement has an invalid LPAR name, the following message appears:
<LPAR> NOT FOUND IN A SYSPLEX RECORD
- On defining a dataset or LPAR record in the control file, the following message appears:
<n> RECORD(S) ADDED TO CONTROL FILE

Syntax to define the DATASET name

```
DEFINE | DATASET NAME(dataset-name) | - GROUP(group-name) | STATUS(status) |
```

Keyword	Description
DATASET NAME	Defines the name of the data set. This keyword is mandatory.

Keyword	Description
GROUP	Defines the names of the group(s) this data is a member of. This keyword is mandatory. You can define up to 4000. For multiple GROUPs, the GROUP names are delimited by one or more spaces (For example- GROUP (G1 G2 G3)).
STATUS	The value entered enables or disables the dataset defined. The default value is ENABLE. This keyword is optional. Valid values are ENABLE and DISABLE.

Note:

- If DEFINE DATASET statement do not include a GROUP statement, the following message appears:
AT LEAST ONE GROUP IS REQUIRED FOR DEFINE DATASET
- If the GROUP statement is not defined in the LPAR records, the following message appears:
<GROUP> NOT FOUND IN AN LPAR RECORD

Syntax to modify the DATASET name:

```
MODIFY DATASET NAME|(dataset-name)| - GROUP|(group-name)| STATUS|(status)
```

Keyword	Description
DATASET NAME	This is the name of the data set to be modified. This keyword is mandatory.
GROUP	This is the name of the group this data set is a member. The record for only this group is modified. If the data set is a member of some other group, that record is not modified. This keyword is mandatory. For multiple GROUPs, the GROUP names are delimited by spaces (For example- GROUP (G1 G2 G3)).
STATUS	The value entered enables or disables the dataset defined. The default value is ENABLE. This keyword is optional. Valid values are ENABLE and DISABLE.

Note:

- Whenever a Dataset or an LPAR record is modified in the control file, the following message appears:
<n> RECORD(S) MODIFIED IN CONTROL FILE

Syntax to modify the SYSPLEX:

```
MODIFY SYSPLEX NAME|(sysplex-name)|LPAR(lpar-name)| - IP |(IP-address)| PORT|(port)|
```

Keyword	Description
SYSPLEX NAME	Is the name of the SYSPLEX to be modified. This keyword is mandatory.
LPAR	These are the names of all the LPAR(s) that resides on this SYSPLEX.
IP	This is the TCP/IP name of the SYSPLEX being modified. This keyword is optional.

Keyword	Description
PORT	This is the port number used by the ERLY subsystem on the SYSPLEX being modified. This keyword is optional.

Syntax to delete the DATASET name:

```
DELETE | DATASET NAME(VSAM-dataset-name) |
```

Note:

- When there is more than one group in the DELETE statement, the following message appears:
ONLY ONE GROUP NAME ALLOWED FOR DELETE DATASET
- When a Dataset or an LPAR record is deleted from the Control file, the following message appears:
<n> RECORD(S) DELETED FROM CONTROL FILE

Syntax to delete the SYSPLEX name:

```
DELETE | SYSPLEX NAME(sysplex) |
```

Enable or disable VSAM dataset

To either enable or disable a dataset or a group in h1q.SAVZLOAD(AVZVSED) use the following syntaxes:

Use the following syntax to enable the DATASET name:

```
ENABLE | DATASET(dataset-name) |  
+ GROUP|(group-name) +
```

Keyword	Description
DATASET	Defines the name of the data set. This keyword is mandatory.
GROUP	Defines the name of the group to be enabled.

Use the following syntax to disable the DATASET name:

```
DISABLE | DATASET(dataset-name) | {FORCE}  
+ GROUP(group-name) +
```

Keyword	Description
DATASET	Defines the name of the data set. This keyword is mandatory.
FORCE	Here FORCE will first try and disable the dataset normally and will then attempt a force disable if the DISABLE times out.
GROUP	Defines the name of the group to be disabled.

To disable using force use the following syntax:

```
FORCE | DATASET(dataset-name) |  
+ GROUP(group-name) +
```

Keyword	Description
DATASET	Defines the name of the data set. This keyword is mandatory.
FORCE	The FORCE DATASET forcefully disables the dataset immediately.
GROUP	Defines the name of the group to be force disabled.

Configuring access to sequential files

No modifications are needed to configure the SQL interface to access sequential files. However, you should verify access to sequential files. Optionally, you can specify the number of tracks to read ahead when reading sequential data sets for individual requests.

Before you begin

The server must already be installed.

About this task

The SQL interface for sequential files provides seamless, real-time controlled access to sequential files. It allows ODBC, JDBC, Data Virtualization Manager client, and web clients to access sequential files in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, JDBC, and two-tier and three-tier web implementations. Using the interface, applications can use standard ODBC or JDBC facilities to make SQL requests directly to native VSAM. The result is a relational result set, with no host programming required.

Reading ahead tracks for sequential file access

You can use a Data Virtualization Manager Server Event Facility (SEF) rule to specify the number of tracks to read ahead (MULTACC) when reading sequential data sets for individual requests.

About this task

Using a virtual table (VTB) rule, you can specify the number of tracks to read ahead (the MULTACC parameter value) for MapReduce sequential file access for individual requests. This support overrides the value in the server parameter **ACIMAPREDUCETRACKS (NUMBER OF MAP REDUCE TRACKS TO READ)** for individual requests. Sample VTB rule AVZMLTAC is provided.

To override the MULTACC value, you must enable the AVZMLTAC rule and use the `MACC_nn_` prefix for table names in your SQL statement.

The AVZMLTAC rule is invoked every time a table with the prefix `MACC_nn_` is found in the SQL statement. The following format is expected:

```
MACC_nn_virtualtablename
```

Where:

- `nn` is the number of tracks to read ahead (the MULTACC value) when reading sequential data sets
- `virtualtablename` is the name of the virtual table

For example:

```
SELECT * from MACC_15_STAFF_SSEQ ;
```

The following message is displayed in the Server Trace:

```
AVZ1000I VTB.OPTBMACC set to 15
```

Use the following procedure to set up the rule.

Procedure

1. Customize the Data Virtualization Manager configuration member (AVZSIN00) to enable virtual table rule events by configuring the SEFVTBEVENTS parameter in the member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

2. Access the VTB rules, as follows:
 - a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
 - b) Specify option 2, **SEF Rule Management.**
 - c) Enter VTB for **Display Only the Ruleset Named.**
3. Enable the rule by specifying E next to AVZMLTAC and pressing Enter.
4. Set the rule to Auto-enable by specifying A next to AVZMLTAC and pressing Enter.

Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring access to IBM CICS

For VSAM data access via the CICS Transaction Server (TS), you need to configure the Data Virtualization Manager configuration member and CICS TS.

Before you begin

The server must already be installed.

To activate the AVZ/CICS interface, the EXCI modules that are located in CICS SDFHEXCI are required by Data Virtualization Manager at startup. You can either APF authorize the CICS SDFHEXCI and add it to AVZ STEPLIB or you can copy the modules from CICS SDFHEXCI to AVZ load library because it is already in STEPLIB and is APF authorized.

About this task

The server connects to CICS TS, via the IBM EXCI (External CICS Interface).

CICS provides logging and recovery facilities that are required if VSAM updates are being applied. When accessing VSAM files that are owned by CICS TS, recovery is provided by CICS TS.

Modifying the Data Virtualization Manager configuration member

Enable the VSAM data access via CICS TM parameters in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment "ENABLE CICS TRANSACTION SERVER SUPPORT."
2. Enable the CICS TS parameters by changing `if DontDoThis` to `if DoThis`.

```
if DoThis then
do
  "MODIFY PARM NAME(EXCI) VALUE(YES) "
  "MODIFY PARM NAME(EXCICONNECTIONNAME) VALUE(CICA) "
  "MODIFY PARM NAME(TRACEEXCIDPLEVENTS) VALUE(YES) "
```

```
"MODIFY PARM NAME(CICSENDABCODE) VALUE(YES)"
"MODIFY PARM NAME(RRSCICS) VALUE(YES)"
```

The following table lists the parameters for configuring support for a CICS TS:

Parameter	Description	Valid values
EXCI	Initialize EXCI support.	NO YES Default value.
EXCICONNECTIONNAME	EXCI Default Connection Name. Specifies the default CICS Connection Name for EXCI support.	EXCW
TRACEEXCIDPLEVENTS	Trace EXCI DPL Events	NO Default value. YES
CICSENDABCODE	Send ABEND Code to Clients. Controls the sending of the CICS ABEND code to the client. If set to YES, the ABEND code is returned to the client as part of the error message.	NO Default value. YES
RRSCICS	Specifies whether RRS CICS support is active.	NO Default value. YES

3. Create a DEFINE CONNECTION statement for each CICS region. Include the following parameters in the statement:

```
"DEFINE CONNECTION NAME(CICA)",
    "GROUP(CICA)",
    "ACCESSMETHOD(IRC)",
    "NETNAME(CICADBVS)",
    "INSERVICE(YES)",
    "PROTOCOL(EXCI)",
    "APPLID(XXXXXXXX)",
    "LOADBALGROUP(LBG1)",
    "SECURITYNAME( )",
end
```

Parameter	Description	Valid values
NAME	Specify a four-character name for the connection to the CICS region.	Four-character name
GROUP	Specify the same name as the connection name.	Eight-character name
ACCESSMETHOD	Specify IRC.	IRC

Parameter	Description	Valid values
NETNAME	Specify the network name of the remote system. To identify these connections in CICS, use a name that is a combination of the connection name and the server subsystem name.	Eight-character name
INSERVICE	Specify YES to open the connection at server startup. Specify NO to open the connection manually.	NO Default value. YES
PROTOCOL	Specify EXCI.	EXCI
APPLID	Specify the VTAM APPLID of the target CICS.	No restriction on the APPLID name
LOADBALGROUP	Specify the name of the group that is used to balance the CICS workload across multiple CICS regions. Specify the same group name in each DEFINE CONNECTION statement that you create. (<i>Optional</i>)	Four-character name
SECURITYNAME	Specify a valid security name from the remote system.	Eight-character name

4. Create a DEFINE SESSION statement for each CICS region. Include the following parameters in each statement:

```

"DEFINE SESSION  NAME(CICA) ",
                  "GROUP(CICA) ",
                  "CONNECTION(CICA) ",
                  "PROTOCOL(EXCI) ",
                  "RECEIVERFX(XD) ",
                  "RECEIVERCOUNT(0) ",
                  "SENDPFX(SD) ",
                  "SENDCOUNT(20) ",
                  "IOAREALEN(4096) ",
end

```

Parameter	Description	Valid values
NAME	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Four-character name
GROUP	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Eight-character name
CONNECTION NAME	Specify the same name that you specified for the NAME when you defined the connection for this CICS region.	Four-character name
PROTOCOL	Specify EXCI.	EXCI

Parameter	Description	Valid values
RECEIVEPFX	Not applicable.	This field should be blank its included to provide complete list of connection parameters.
RECEIVECOUNT	Not applicable.	This field should be blank its included to provide complete list of connection parameters.
SENDPFX	Specify a one- or two-character prefix for the session name. The session name, which is limited to four characters, is composed of the prefix and the session number. Therefore, if you define more than 99 sessions, specify a one-character prefix.	A one- or two-character prefix
SENDCOUNT	Specify the maximum number of concurrent transactions. This value should match the RECEIVECOUNT value minus one set in the DEFINE SESSIONS definition in the AVZCICSD job.	Any number up to 255
IOAREALEN	Specify the length, in bytes, of the terminal input/output area to use to process transmitted messages.	Any value up to 4096K

Configuring CICS

Configure CICS by modifying the CICS started tasks JCL, the System Initialization Table (SIT), and the DFHCSD file.

Procedure

1. Add the *hlq.SAVZCLOD* library to the DFHRPL concatenation in each CICS region that you want to connect to server.
2. Use the CEMT INQUIRE IRC command to verify that the CICS interregion communication (IRC) facility is open.

To start IRC at CICS system startup, ensure that the IRCSTRT=YES parameter is in the SITPARM for the CICS region.

3. Update the DFHCSD file by performing the following steps:
 - a) For each CICS region, modify and submit the AVZCICSD job that is in *hlq.SAVZCNTL* data set:
 - Update the DEFINE CONNECTION and DEFINE SESSION values to match the definitions that you specified in the Data Virtualization Manager configuration member. The GROUP value is the CICS GROUPNAME and does not need to match the GROUP name that is defined for the server. By default, 21 sessions are defined. Set this value to the maximum number of concurrent transactions for a single instance of server. The maximum value is 250.
 - Change the name of the *hlq.FILEA* data set to the FILEA VSAM data set name. This VSAM file is used when you verify access to CICS data.

The member contains additional information about modifying the job.

- b) Update LIST(*YOURLIST*) to match the startup group list for the CICS region.

- c) Review more comments in the JCL notes section for additional considerations. Define all of the definitions in the *hlq.SAVZCNTL(AVZCICSD)* member.

Configuring security

Configure security to provide user access to CICS TS.

About this task

See "CICS security" in the *Administrator's Guide*.

Configuring access to zFS files

The Data Virtualization Manager server is already configured to support zFS files. No modifications are needed to configure access to zFS files. However, you should verify access to zFS files.

Configuring access to data for IT Operational Analytics

IT Operational Analytics (ITOA) allows you to retrieve, analyze, and report data for IT operations. System information can be logged using the IBM System Management Facility (SMF) and the native Data Virtualization Manager server logging feature. Logging allows you to collect various system and operations-related information.

Before you begin

Verify that the following IBM APARs have been applied:

- [APAR OA49263](#). This APAR provides real-time SMF support and is a requirement for the configuration of real-time SMF data access. (The closed date for this APAR is 2016-08-31.)
- [APAR OA48933](#). This APAR is required to address accessing log streams. SMF log stream configuration is required for in-memory resource support. (The closed date for this APAR 2015-11-24.)

About this task

Virtual tables for SMF are provided in the *hlq.SAVZSMAP* data set.

The following options are available to access the SMF data:

- Reading data from SMF data sets - SMF information is recorded in MANx data sets. When a data set gets full, the data is processed via IFASMFDP. When defining global variables for accessing SMF data in data sets, the output of IFASMFDP is used.
- Reading data from log streams - SMF information is recorded in multiple log streams and data can be read directly from the log streams. Log stream recording is determined by the data set name beginning with IFASMF that is used in the VTB rule for SMF.
- Reading SMF data from in-memory (real-time) - SMF information is read directly from the system buffer. SMF information is read in real time. There are two interfaces to real-time SMF data, which connect to the in-memory resource at different times, as follows:
 - At product initialization. This interface connects to the in-memory resource at product initialization and continuously reads from the API to maintain a buffer of recent SMF activity. This buffer can be queried, and its contents will be returned, followed by an end-of-data indication.
 - At the time of the request. This interface connects to the in-memory resource at the time of the request and streams the SMF data to the requester in real time. A request to this named stream is considered non-ending, and data will continue to flow until the request is canceled or the server is stopped.

When defining the global variables for SMF, the data set can be either a log stream or a SMF dump data set from IFASMFDP. The log stream data set is recommended for access to near real-time data.

To configure access to IT Operational Analytics data, see the following topics:

- [“Configuring access to System Management Facility \(SMF\) files” on page 158](#)
- [“Configuring access to SYSLOG files” on page 161](#)
- [“Configuring access to OPERLOG files” on page 162](#)

Configuring access to System Management Facility (SMF) files

To configure access to System Management Facility (SMF) files, you need to configure the server started task JCL, the server configuration member, and the server virtual table member. To enable reading SMF data real-time using log streams, you must have the **SMFPRMxx** member in the system PARMLIB data set configured to use both log streams and in-memory resources. Follow the steps in this section to use SMF GDG data set names, or to use dynamic data set names.

About this task

SMF data set names are dynamic in local environments and require SEF rules enablement and optionally Global Variables set to specific values to provide data set names to the virtual tables and views when using SMF data set or log stream configurations.

You can choose either GDG data set name support or dynamic data set name support, or both, to quickly access your SMF data. These two options are provided for your convenience to help you start accessing your SMF data. Custom rules may need to be developed to use your local naming convention to access your SMF files.

Procedure

1. Configure the server started task JCL by concatenating the *hlq.SAVZSMAP* data set to the AVZMAPP DD statement to add all maps for SMF.
2. Customize the server configuration member.
To enable virtual table rule events, configure the SEFVTBEVENTS parameter in the AVZSIN00 member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

Verify the VTB ruleset name:

```
"DEFINE RULESET NAME(VTB) "  
"RULETYPE(VTB) "  
"DSNAME(' || SHLQ2 || '.SAVZXVTB' ) "
```

If there were any changes to AVZSIN00, recycle the server started task.

3. To enable real-time access to SMF data, add the following statements to the AVZSIN00 member after the GLOBAL PRODUCT OPTIONS statement.

```
IF DoThis  
  THEN DO  
    "DEFINE SMF NAME(IFASMF.INMEM) ",  
    "STREAM(IFASMF.INMEM.STREAM) ",  
    "BUFSIZE(500) ",  
    "TIME(0) "  
END
```

Note: You must have the **SMFPRMxx** member in the system PARMLIB data set configured to use log streams and in-memory resources.

Parameter	Description	Valid values
NAME	Specifies the name of the in-memory resource. This value must match the name of a resource defined to SMF with the INMEM parameter. If this parameter is included, the in-memory API will be read continuously and a buffer of the most recent records will be maintained. Either this parameter or the STREAM parameter, or both, must be specified.	This parameter must contain the name of an in-memory resource defined to SMF with the INMEM statement. The format of the name is defined by SMF configuration, which is 1-26 characters and must begin with IFASMF.
STREAM	Specifies the name of the streaming in-memory feature. If this name is specified on a SELECT statement, a dynamic connection will be made to the SMF in-memory API and records will be streamed to the caller in real time. Either this parameter or the NAME parameter, or both, must be specified.	If a NAME parameter is also supplied, the in-memory resource named in that parameter will be connected to and the value of this parameter can be any name, 1-26 characters. If the NAME parameter is not supplied, this parameter must contain the name of an in-memory resource defined to SMF with the INMEM parameter. If both NAME and STREAM are provided, the names must be different.
BUFSIZE	Indicates how much SMF data (megabytes) will be retained in memory for queries. If the buffer fills up, the oldest data will be discarded. In parallel, SMF is recording these records to a log stream. This parameter applies to the resource named in the NAME parameter.	1-10,000
TIME	Indicates how long (in minutes) to keep SMF data in memory. Older data will be discarded. Specifying 0 indicates no time limit and data will be retained until the buffer fills up. This parameter applies to the resource named in the NAME parameter.	0-1440

4. To use SMF data in compressed log streams, add the following statement to the AVZSIN00 member:

```
"MODIFY PARM NAME(ZEDCCOMPRESSION) VALUE(YES) "
```

Note: You must have the **SMFPRMxx** member in the system PARMLIB data set configured to use compressed log streams, and the zEDC Express hardware feature must be installed.

5. To use SMF_1100P* maps, add the following statements to the AVZSIN00 member:

```
"MODIFY PARM NAME(ACIMAPREDUCEBUFF) VALUE(16383K)"
"MODIFY PARM NAME(ACIMAPREDUCESPACE) VALUE(64)"
```

6. Enable reading SMF data from GDG data sets and access to SMF data using dynamic data set names by enabling Data Virtualization Manager Server Event Facility rule AVZSMFT1 in the VTB ruleset. You can select from a GDG data set, any SMF dump data set, a log stream data set, or the in-memory stream. Activate your options by customizing the rule.

a) Use the following steps to enable rule AVZSMFT1 in the VTB ruleset:

- i) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
- ii) Specify option 2, **SEF Rule Management.**
- iii) Enter VTB for **Display Only the Ruleset Named.**
- iv) Enable the rule by specifying E and pressing Enter.
- v) Set the rule to Auto-enable by specifying A and pressing Enter.

Setting the rule to Auto-enable activates the rule automatically when the server is re-started.

b) Configure the access method using one or more of the following methods:

- Review the information in the rule for the instructions on setting Global Variables that will be used by the rule. Navigate one screen back on the ISPF panel, or start over by going to option E, **Rules Mgmt.**, and then option 1, **Global Variables.** In the Global Variables display, perform the following steps:
 - i) Change Global Prefix to GLOBAL2.
 - ii) Select SMFTBL2 by entering S next to the SMFTBL2 data set.
 - iii) Configure the SMF data access option. DEFAULT should have corresponding SMF dump data set names if used. This option can be used to specify the source SMF, such as GDGBASE, INMEM, and LOGSTREAM.

Note:

VTB rules and global variables may be used to reference a GDG data set, any SMF dump data set, a log stream data set, or the in-memory stream. For example:

```
GLOBAL2.SMFGBL2.YESTERDAY = "YOUR.DATASET.SMFDUMP(-1)"
GLOBAL2.SMFGBL2.M2 = "YOUR.DATASET.SMFDUMP(-2)"
GLOBAL2.SMFGBL2.M3 = "YOUR.DATASET.SMFDUMP(-3)"
GLOBAL2.SMFGBL2.M4 = "YOUR.DATASET.SMFDUMP(-4)"
GLOBAL2.SMFGBL2.M5 = "YOUR.DATASET.SMFDUMP(-5)"
GLOBAL2.SMFGBL2.IM = "IFASMF.INMEM"
GLOBAL2.SMFGBL2.IM2 = "IFASMF.INMEM2"
GLOBAL2.SMFGBL2.LOG = "LOGSTREAM.dataset.name"
```

- Pass a dynamic data set name for SMF tables using the following format for the table name in the SQL statement:

```
TableMapName__DataSetName
```

Where DataSetName is prefixed by two underscores (__) and the periods in the data set name are replaced with single underscores (_).

For example, SELECT * FROM SMF_01400__DATA_SET_NAME would translate into an SQL query of SELECT * FROM SMF_14000 and access the data set DATA.SET.NAME.

- Pass a dynamic data set name for SMF virtual views using the following format for the virtual view name in the SQL statement:

```
ViewMapName__DataSetName
```

Where DataSetName is prefixed by two underscores (__) and the periods in the data set name are replaced with single underscores (_).

For example, `SELECT * FROM SMFV_01400__DATA_SET_NAME` would translate into an SQL query of `SELECT * FROM SMFV_01400` and access the data set `DATA.SET.NAME`.

Configuring access to SYSLOG files

To configure access to system log (SYSLOG) files, you need to configure the Data Virtualization Manager configuration member and the server virtual table rules.

About this task

Virtual table rules are provided that support the processing of SYSLOG files and vary based on the type of file name used for your SYSLOG data sets. Each of the rules for SYSLOG processing requires that the table names in the SQL begin with SYSLOG. The following rules are provided:

AVZSYSLG

This rule uses a global variable to specify the name of the data set to use for the SYSLOG data.

AVZSYSL2

This rule supports the use of generation data group (GDG) data set names. One of the following formats is expected:

- `SYSLOG_GDG_nnnn`

Where *nnnn* is a relative GDG number (between 0 and 9999) that is appended to the GDG base name value that is obtained from the `GLOBAL2.SYSLOG.GDGBASE` variable. For example, if the table name as specified in the SQL statement is `SYSLOG_GDG_1`, then the data set name returned by this rule is `HLQ.SYSLOG(-1)`, depending on the value in `GLOBAL2.SYSLOG.GDGBASE`.

- `SYSLOG_DSN_suffix`

Where *suffix* is used as the last part of a global variable of the form `GLOBAL2.SYSLOG.suffix` in order to look up the name of the data set to be used. If this variable does not exist, the data set name specified in `GLOBAL2.SYSLOG.DEFAULT` is used to read the SYSLOG records.

By using global variables, you do not need to modify the code in the rule. The following are some examples of global variables that can be set up to be used in conjunction with this rule:

```
Global Prefix: GLOBAL2.SYSLOG
S Subnode Name   Nodes          Subnode Value
-----
GDGBASE          0 HLQ.SYSLOG
DEFAULT          0 HLQ.SYSLOG(0)
TODAY            0 HLQ.SYSLOG(0)
YESTERDAY        0 HLQ.SYSLOG(-1)
```

AVZSYSL3

This rule lets you dynamically specify in your SQL the name of the data set to use when processing SYSLOG files. In the SQL, the table name must begin with the prefix SYSLOG; the rest of the table name is used by the rule to determine the actual data set name to use for processing the SYSLOG records.

The following format is expected:

```
SYSLOG__DataSetName
```

Where *DataSetName* is preceded by two underscores (__) and the periods in the data set name are replaced with single underscores (_). For example, `SELECT * FROM SYSLOG__DATA_SET_NAME` would translate into an SQL query of `SELECT * FROM SYSLOG` and access the data set `DATA.SET.NAME`.

To use one of the rules, you must enable the rule and use the prefix SYSLOG for table names in your SQL statement. The enabled rules are invoked every time a table with the prefix SYSLOG is found in the SQL statement.

Use the following procedure to set up the rules.

Procedure

1. Customize the Data Virtualization Manager configuration member.

To enable virtual table rule events, configure the SEFVTBEVENTS parameter in the AVZSIN00 member, as follows:

```
"MODIFY PARM NAME(SEFVTBEVENTS) VALUE(YES) "
```

2. Access the VTB rules, as follows:
 - a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
 - b) Specify option 2, **SEF Rule Management.**
 - c) Enter VTB for **Display Only the Ruleset Named.**

3. For AVZSYSLG, customize the rule, as follows:

- a) Specify S next to AVZSYSLG to edit the rule.
- b) Customize the rule with the SYSLOG data set name.
- c) Save your changes and exit the editor.

Note: For AVZSYSL2 and AVZSYSL3, no customization of the rule is needed.

4. Enable each rule by specifying E next to the member name and pressing Enter.
5. Set each rule to Auto-enable by specifying A next to the member name and pressing Enter.
Setting a rule to Auto-enable activates the rule automatically when the server is re-started.
6. If global variables are needed, set up the SYSLOG global variable.

Configuring access to OPERLOG files

No modifications are needed to configure the Data Virtualization Manager server to access OPERLOG data; however, OPERLOG must be active in a system logger log stream.

About this task

Use the following procedure to verify that OPERLOG is active in a system logger log stream.

Procedure

To display the active medium where messages are recorded, enter the following command:

```
D C,HC
```

The following results are expected:

```
CNZ4100I 15.19.16 CONSOLE DISPLAY 056
CONSOLES MATCHING COMMAND: D C,HC
MSG:CURR=0    LIM=9000 RPLY:CURR=0    LIM=9999  SYS=P02    PFK=00
HARDCOPY LOG=(SYSLOG,OPERLOG)  CMDLEVEL=CMDS
          ROUT=(ALL)
LOG BUFFERS IN USE: 0          LOG BUFFER LIMIT: 9999
```

Configuring access to CA IDMS

To access CA IDMS data, you must configure the Data Virtualization Manager server started task JCL. You can then optionally verify access to the data.

Data Virtualization Manager server started task JCL changes are required to access CA IDMS software and define default CA IDMS settings.

Restrictions

The following restrictions and considerations apply when accessing CA IDMS data:

- SELECT-only support is provided.
- CA IDMS Logical Record Facility (LRF) is not supported. Virtual views provide many of the same capabilities as LRF and can be used in place of LRF.
- Data access uses CA IDMS network DML only. The CA IDMS SQL product is not required.

Note:

Server configuration parameters control the following behaviors and can be modified if necessary:

- CA IDMS run-unit management, specifically maximum run-units and a timeout value for inactive run-units
- CA IDMS access tracing

Configuring the server started task JCL

Modify the server started task JCL to access CA IDMS and define default CA IDMS settings.

Before you begin

All LOAD library data sets allocated to the Data Virtualization Manager server in the server started task JCL must be APF-authorized.

About this task

Modify the server started task JCL to access CA IDMS and define default IDMS settings.

Procedure

1. Add the CA IDMS load libraries to the STEPLIB, which are required for CA IDMS central version access.
2. Add the SYSTCL DD statement identifying the CA IDMS central version to access.
3. Add the SYSIDMS statement with additional environment parameters. Minimally, this data set should include a CVRETRY=OFF statement to prevent an WTOR message when the CA IDMS central version is not active.
4. Add the CA IDMS system message data set to DCMSG.

Modifying the Data Virtualization Manager configuration member for CA IDMS

To optionally configure server parameters for CA IDMS, you can update your Data Virtualization Manager server configuration file.

About this task

The CA IDMS server parameters can assist you in configuring CA IDMS data access. In most typical environments, the default settings for these parameters will not need modification.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization.
2. Add the following statements to your AVZSIN00 member:
The following table lists the parameters for configuring CA IDMS data access:

Parameter	Description	Valid values
MAXIDMSRUNUNITS	<p>MAXIMUM IDMS RUN UNITS</p> <p>This parameter limits the number of concurrent IDMS run units that a server will start to access a CA IDMS central version. Limiting concurrent IDMS run units will prevent storage related user 3134 abends when creating run units with CA IDMS.</p>	Positive numeric value. Default value is 4.
SQLENGIDMSRUTIMOUT	<p>IDMS RUN UNIT INACTIVITY TIMEOUT</p> <p>Specifies the length of time in seconds to keep a run unit active for reuse by subsequent SQL queries in a client connection.</p>	Positive numeric value. Default value is 60 seconds.

Verifying access to CA IDMS data

To verify access to CA IDMS data, you can optionally install a set of maps to the sample database EMPDEMO and run queries using the installed maps.

Before you begin

The CA IDMS sample database EMPDEMO must be installed in the central version you plan to access.

About this task

You can customize and run the provided IVP job AVZISIV1 to install maps to the EMPDEMO database and network schema maps to the SYSTEM database.

The following maps are installed for verification testing using the sample EMPDEMO database:

Map	Description
EMPSS01_EMPLOYEE	Enables SQL access to EMPLOYEE record.
EMPSS01_OFFICE	Enables SQL access to the OFFICE record.
EMPSS01_DEPARTMENT	Enables SQL access to the DEPARTMENT record.
EMPSS01_OFFICE_EMPLOYEE	Enables SQL access to the OFFICE-EMPLOYEE set for joining the EMPSS01_OFFICE and EMPSS01_EMPLOYEE tables.
EMPSS01_DEPT_EMPLOYEE	Enables SQL access to the DEPT-EMPLOYEE set for joining the EMPSS01_DEPARTMENT and EMPSS01_EMPLOYEE tables.

The network schema maps can be used for verification purposes if the EMPDEMO database is not installed in your central version. These maps access records and sets in the CA IDMS network schema IDMSNTWK, providing SQL access to application metadata. The following table provides a subset of the installed network schema maps that can be used for verification purposes:

Table 15. CA IDMS network schema IDMSNTWK maps

Map	Description
IDMSNWKA_S_010	Enables SQL access to the S-010 network schema record. S-010 records describe application schemas defined to your IDMS central version.
IDMSNWKA_SS_026	Enables SQL access to the SS-026 network schema record. SS-026 records describe application subschemas defined to your IDMS central version.
IDMSNWKA_SSR_032	Enables SQL access to the SSR-032 network schema record. SSR-32 records describe application subschema records defined to your IDMS central version.
IDMSNWKA_S_SS	Enables SQL access to the S-SS set for joining the IDMSNWKA_S_010 and IDMSNWKA_SS_026 tables.
IDMSNWKA_SS_SSR	Enables SQL access to the SS-SSR set for joining the IDMSNWKA_SS_026 and IDMSNWKA_SSR_032 tables.

Procedure

1. Locate the AVZISIV1 member in the *hlq.SAVZCNTL* data set.
2. Modify the JCL according to the instructions provided in the AVZISIV1 member.
3. Submit the job.
4. If the server is active, use the following instructions to refresh maps and make the maps available for use:
 - a) From the Primary Option Menu, specify option D, **Data Mapping**, and press Enter.
 - b) From the Data Mapping Facility menu, specify option 3, **Map Refresh**, and press Enter.

Results

AVZISIV1 installs CA IDMS EMPDEMO and network schema maps into the server map data set.

Configuring access to ADDI

To use IBM Application Discovery and Delivery Intelligence (ADDI) information for creating virtual maps that access VSAM and sequential data, you must configure the server for ADDI access.

System requirements

The following system requirements apply:

- IBM Application Discovery Suite Version 5.0 or newer
- Microsoft Host Integration Server (HIS) 2016 or higher. The SYSIBM views that are part of the Microsoft HIS Software Development Kit must be installed as part of the HIS installation.
- Microsoft SQL Server 2012 Enterprise or Express or higher

Restrictions

The following restrictions and considerations apply when using ADDI to access VSAM and sequential data sets:

- Virtual table creation is restricted to data sets in the ADDI project that are processed by COBOL programs using JCL. Data sets accessed using CICS as well as other databases (such as IMS, CA IDMS, or Adabas) are not supported.
- Virtual table mapping is only supported through the Data Virtualization Manager studio. No batch utilities or ISPF interfaces are provided to map tables.

Configuration steps

The following configuration steps are required to use ADDI to access VSAM and sequential data:

1. Install virtual tables. See [“Installing virtual tables and virtual target maps for ADDI access”](#) on page 166.
2. Define ADDI project in the server configuration member. See [“Modifying the configuration member for ADDI access”](#) on page 166.
3. Activate virtual table rules. See [“Configuring virtual table rules for ADDI”](#) on page 170.
4. Define credentials for target database(s). See [“Configuring authentication for ADDI”](#) on page 170.

Installing virtual tables and virtual target maps for ADDI access

Install virtual tables and virtual target maps for IBM Application Discovery and Delivery Intelligence (ADDI) access.

About this task

The Data Virtualization Manager studio reads the ADDI project using virtual tables and views installed as part of server set up. The following maps are distributed in XMIT format in the SAVZSAMP member AVZIAMPD:

ZIADTSR

Virtual target system TSIAD_PROJECT1 for external subsystem named IAD1.

ZIADT001-ZIADT021

Virtual tables that map tables in the ADDI project. Each virtual table uses the name of the corresponding ADDI project table with the added prefix IAD_. For example, SQL Server table `dbo.Variables` has a virtual table name of `IAD_VARIABLES`.

ZIADV001-ZIADV002

Virtual views on the IAD_ virtual tables used by the Data Virtualization Manager studio to read ADDI data. These views are all prefixed with IADV_ (for example, `IADV_DATASETS`). All data access from the studio is performed using virtual views.

These maps are not installed by default. Use the following procedure to install these maps.

Procedure

1. Locate the AVZIAMPD member in the `hlq.SAVZCNTL` data set.
2. Modify the JCL according to the instructions provided in the AVZIAMPD member.
3. Submit the job.

The virtual tables and virtual target maps are installed.

Modifying the configuration member for ADDI access

Enable and configure the parameters for IBM Application Discovery and Delivery Intelligence (ADDI) in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member contains a sample DATABASE definition that defines the first ADDI project. The initial definition is named IAD1 and is disabled.

When enabling the database definition for the first ADDI project, the LOCATION and IPADDR parameters must be set to the correct project name and IP address of the Microsoft HIS DRDA Provider Service for SQL Server. The LOCATION provides the name of the SQL Server project, and IPADDR(...) PORT(...) provide the TCP/IP information for the HIS DRDA Service. DOMAIN(...) can be used instead of IPADDR to provide the DNS of the HIS DRDA Service. The subsystem NAME(IAD1) should not be changed because a target subsystem map is configured to use this name for the virtual tables accessing the ADDI project.

For multiple ADDI projects, see [“Adding an ADDI project” on page 169](#).

The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment “Sample IBM Application Discovery configuration”.
2. Enable the ADDI parameters by changing the syntax `if DontDoThis` to `if DoThis`. The following example shows the section in the configuration member to enable:

```

/*-----*/
/* Sample IBM Application Discovery configuration using DRDA to */
/* communicate with a Microsoft SQLServer database.          */
/*-----*/
if DoThis then do
"DEFINE DATABASE TYPE(MSSQL) "
"NAME(IAD1) "
"LOCATION(EZ_Project1) "
"DDFSTATUS(ENABLE) "
"SECMEC(USRIDPWD) "
"IPADDR(::FFFF:0.0.0.0) "
"PORT(446) "
"CCSID(37) "
"IDLETIME(0) "
end

```

The following table lists the parameters for configuring support for ADDI:

Parameter	Description	Valid values
TYPE	Database type. Because ADDI stores information in Microsoft SQL Server, this value must be MSSQL.	MSSQL
NAME	<p>The database name as known to the server.</p> <p>The first definition must be IAD1 because the target system map names this as the subsystem to access for ADDI.</p> <p>For additional ADDI projects, subsystems can have any name since you must also create a virtual target system to point to it; however, it recommended that the name start with IAD.</p> <p><i>(Required)</i></p>	A valid value consists of 1 - 4 characters. For example, IAD1.

Parameter	Description	Valid values
LOCATION	<p>Name of the database for the ADDI project.</p> <p>The LOCATION parameter must be set to the correct database name of the target MSSQL server.</p> <p><i>(Required)</i></p>	<p>A valid value is a string 1 - 16 characters.</p>
DDFSTATUS	<p>The DDF activation status</p> <p><i>(Required)</i></p>	<p>ENABLE Make this DDF definition active within Data Virtualization Manager server. DDFSTATUS should always be ENABLE for TYPE(MSSQL).</p> <p>DISABLE DDF endpoint is not used. This value disables the MSSQL database. This value should only be used if the database is off-line or otherwise not available for access.</p>
SECMEC	<p>Security mechanism. The DRDA security mechanism for authentication with the HIS DRDA Service for SQL Server.</p> <p>The SECMEC setting for TYPE(MSSQL) must match the HIS DRDA Service configuration.</p>	<p>USRIDPWD User ID and password</p> <p>USRIDONL User ID only</p> <p>USRENCPWD Encrypt the password only</p> <p>EUSRIDPWD Encrypt the user ID and password</p>
IPADDR	<p>Specify the IPV4 or IVP6 address of the target MSSQL server.</p> <p>Use DOMAIN instead of IPADDR to supply the DNS of the target HIS DRDA Server for SQL Server. Use DOMAIN if the IPADDR or the HIS DRDA Service Provider can change.</p> <p>Either DOMAIN or IPADDR is required, but not both.</p>	<p>A valid IPV4 or IVP6 address set to the correct remote IP address for the system running Microsoft SQL Server.</p>

Parameter	Description	Valid values
DOMAIN	The part of a network address that identifies it as belonging to a particular domain. Use DOMAIN instead of IPADDR to supply the DNS of the target HIS DRDA Server for SQL Server. Use DOMAIN if the IPADDR or the HIS DRDA Service Provider can change. Either DOMAIN or IPADDR is required, but not both.	No default value.
PORT	The TCP/IP port defined for Microsoft HIS DRDA Service Provider. For TYPE(MSSQL), the standard HIS default is 446. <i>(Required)</i>	A valid 1-5 numeric string.
CCSID	Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier). <i>(Required)</i>	Refer to the Microsoft SQL Server documentation for a list of valid CCSIDs. Refer to the ISV documentation on HIS DRDA Service to SQL Server. For USA, this value is 037.
IDLETIME	This setting is not used for TYPE(MSSQL).	0

Adding an ADDI project

Perform required configuration steps to add an ADDI project.

About this task

For multiple ADDI projects, you must perform configuration steps to define each additional ADDI project. The following requirements apply when maintaining multiple ADDI projects:

- For the first instance of an ADDI project:
 - The database name in the must be IAD1.
 - The target system for the name IAD1 is automatically installed with the ADDI maps, as described in [“Installing virtual tables and virtual target maps for ADDI access”](#) on page 166.
- For subsequent ADDI projects:
 - It is recommended that the database name start with IAD.
 - The target system must start with TSIAD.

Perform the following procedure for each additional ADDI project.

Procedure

1. Repeat the database definition in the configuration member and make the following changes:
 - a) Change the NAME value to a unique name (for example, IAD2).

b) Change the LOCATION value to match the Microsoft SQL Server project name containing the ADDI project you need to access.

For information about the database definition parameters, see [“Modifying the configuration member for ADDI access”](#) on page 166.

2. Define a new virtual target system using the studio. The name of the virtual target system must start with TSIAD. This can be done in the Data Virtualization Manager studio by selecting the **Create Virtual Target System** in the **Server** tab under the **SQL > Target Systems > DBMS** node of the tree. The connection value in each definition must match the NAME value defined in the DATABASE definition in the configuration member.
3. If required, create authentication information using the AVZDRATH batch utility.

Configuring virtual table rules for ADDI

Configure Data Virtualization Manager Server Event Facility (SEF) rules to support multiple projects using common virtual table and view definitions.

About this task

To support multiple projects using common virtual table and view definitions, VTB rules AVZIADTB and AVZIADVW provide support to process tables starting with IAD_ and views starting with IADV_.

AVZIADTB

This table rule looks at the base view of a query for double underscores “__” and uses the data after the underscores to update the target subsystem for the query.

AVZIADVW

This view rule looks for the double underscores and removes them from the view name to process.

With the rules activated, the Data Virtualization Manager studio can suffix the view names with __SSID for all calls and process multiple ADDI projects using a single set of maps.

These rules must be activated regardless of the number of ADDI projects to be enabled.

Use the following procedure to set up these rules.

Procedure

Use the following steps to enable rules AVZIADTB and AVZIADVW in the VTB ruleset:

- a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
- b) Specify option 2, **SEF Rule Management.**
- c) Enter VTB for **Display Only the Ruleset Named.**
- d) Enable the rules by specifying E and pressing Enter.
- e) Set the rules to Auto-enable by specifying A and pressing Enter.

Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring authentication for ADDI

Configure authentication for communicating with the IBM Application Discovery and Delivery Intelligence (ADDI) project.

About this task

It is common for data centers to assign different user IDs for access to z/OS and for access to SQL Server. By default, the server will attempt to log on to SQL Server with the same user ID that was presented for logon to z/OS. A facility is provided in the server to optionally change the logon credentials for a user when accessing SQL Server.

When communicating between the Data Virtualization Manager server and the ADDI project, you must define what credentials to use in MSSQL connections if z/OS users are not defined as users to SQL Server. To accomplish this, the following tools are provided:

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. Use this utility to define alternate logon information for the Data Virtualization Manager server started task and z/OS users. This utility places SQL Server authentication information in GLOBALU system variables for connecting to ADDI projects. You can also use this utility to list existing credential information.

AVZEMSSG

An ATH rule that swaps z/OS user information with SQL Server authentication information defined using the AVZDRATH utility. This rule uses AES encrypted passwords stored as GLOBALU system variables.

You can use any of the following options for authentication:

- Use z/OS IDs for authentication
- Add a global default user definition using sample job AVZDRATH and enable ATH rule AVZEMSSG
- Add authentication information for specific mainframe users using sample job AVZDRATH and enable ATH rule AVZEMSSG

Network administrators may need to open ports for DRDA communication between the z/OS host and the Microsoft SQL Server machine(s) hosting ADDI projects. The default port for Microsoft SQL Server access is 446.

If z/OS user IDs are not defined to Microsoft SQL Server, use the following procedure to define alternate authentication information for the started task and z/OS users requiring access to this feature:

Procedure

1. Use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your ADDI project, as instructed in the JCL, make sure to specify the correct DBTYPE. For ADDI projects, specify DBTYPE=MSSQL.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
2. Auto-enable the SEF ATH rule SAVZXATH(AVZEMSSG) to switch credentials when connecting to ADDI using DRDA. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Set Auto-Enable for the AVZEMSSG rule member by entering A and pressing Enter.

Configuring access to RAA

To use IBM Rational Asset Analyzer (RAA) information for creating virtual maps that access VSAM and sequential data, you must configure the server for RAA access.

System requirements

The following system requirement applies:

- IBM Rational Asset Analyzer for System z 6.1 PID5655-W57

Restrictions

The following restrictions and considerations apply when using RAA to access VSAM and sequential data sets:

- Virtual table creation is restricted to data sets in the RAA database that are processed by COBOL programs using JCL. Data sets accessed using CICS as well as other databases (such as IMS, CA IDMS, or Adabas) are not supported.
- Virtual table mapping is only supported through the Data Virtualization Manager studio. No batch utilities or ISPF interfaces are provided to map tables.

Configuration steps

The following configuration steps are required to use RAA to access VSAM and sequential data:

1. Install virtual tables. See [“Installing virtual tables and virtual target maps for RAA access”](#) on page 172.
2. Define RAA database in the server configuration member. [“Modifying the configuration member for RAA access”](#) on page 173.
3. Activate virtual table rules. See [“Configuring virtual table rules for RAA”](#) on page 176.
4. Define credentials for target database(s). See [“Configuring authentication for RAA”](#) on page 176.

Installing virtual tables and virtual target maps for RAA access

Install virtual tables and virtual target maps for IBM Rational Asset Analyzer (RAA) access.

About this task

The Data Virtualization Manager studio reads the RAA database using virtual tables and views installed as part of server set up. The following maps are distributed in XMIT format in the SAVZSAMP member AVZRAMPD.

ZRAATSPR

Virtual target system TSRAA_PROJECT1 for external subsystem named RAA1.

ZRAAT001-ZRAAT010

Virtual tables mapping tables in the RAA database. All tables use the same name as the corresponding RAA database table with a prefix of RAA_ (for example, “DMH”.DMH_DATA_RECORD” in Db2 has a virtual table name of RAA_DATA_RECORD).

ZRAAV001-ZRAAV003

Virtual views on the RAA_ virtual tables used by the Data Virtualization Manager studio to read RAA data. These views are all prefixed with RAAV_ (for example, RAAV_DATASETS). All data access from the studio is performed using virtual views.

These maps are not installed by default. Use the following procedure to install these maps.

Procedure

1. Locate the AVZRAMPS member in the *hlq*.SAVZCNTL data set.
2. Modify the JCL according to the instructions provided in the AVZRAMPS member.
3. Submit the job.

The virtual tables and virtual target maps are installed.

Modifying the configuration member for RAA access

Enable and configure the parameters for IBM Rational Asset Analyzer (RAA) in the Data Virtualization Manager configuration member.

About this task

The Data Virtualization Manager configuration member contains a sample DATABASE definition that defines the first RAA database.

When enabling the database definition for the first RAA instance, the LOCATION and IPADDR parameters must be set to the database information for the Db2 on z/OS subsystem hosting the RAA database. The subsystem NAME(RAA1) should not be changed because a target subsystem map is configured to use this name for the virtual tables accessing the RAA database.

For multiple RAA databases, see [“Adding an RAA database” on page 175](#).

The server member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

In the AVZSIN00 member, locate the comment “IBM Rational Asset Analyzer location”. The following example shows the section in the configuration member to locate:

```
/*-----*/
/* DRDA definition for IBM Rational Asset Analyzer location. RAA */
/* database definitions must have a NAME() starting with RAA */
/*-----*/
"DEFINE DATABASE TYPE(ZOSDRDA) "
      "NAME(RAA1) "
      "LOCATION(DRDAZOS) "
      "DDFSTATUS(ENABLE) "
      "PORT(443) "
      "IPADDR(127.0.0.1) "
      "CCSID(37) "
      "APPLNAME(DSN1LU) "
      "IDLETIME(100) "

end
```

The following table lists the parameters for configuring support for RAA:

Parameter	Description	Valid values
TYPE	Database type. Because RAA stores information in Db2 for z/OS, this value must be ZOSDRDA.	ZOSDRDA

Parameter	Description	Valid values
NAME	<p>The database name as known to the server.</p> <p>The first definition must be RAA1 because the target system map names this as the subsystem to access for RAA.</p> <p>For additional RAA databases, subsystems can have any name since you must also create a virtual target system to point to it; however, it recommended that the name start with RAA.</p> <p><i>(Required)</i></p>	A valid value consists of 1 - 4 characters, starting with RAA. For example, RAA1.
LOCATION	<p>Name of the database.</p> <p>The LOCATION parameter must be set to the database information for the Db2 on z/OS subsystem hosting the RAA database.</p> <p><i>(Required)</i></p>	A valid value is a string 1 - 16 characters.
DDFSTATUS	<p>The DDF activation status, which can be altered online by using the ISPF 4-Db2 dialog panels.</p> <p><i>(Required)</i></p>	<p>ENABLE Make this DDF definition active within Data Virtualization Manager server.</p> <p>DISABLE DDF endpoint is not used.</p>
PORT	<p>The TCP/IP port at which the server is listening. <i>(Required)</i></p>	A valid 1-5 numeric string.
IPADDR	<p>Specify the dot-notation IPV4 address of the DDF endpoint.</p> <p>For the first RAA instance, the IPADDR parameter must be set to the database information for the Db2 on z/OS subsystem hosting the RAA database.</p> <p><i>(Optional)</i></p>	If this parameter is not specified, the value 127.0.0.1 (local host) is the default. For group director definitions, use the DVIPA IP address of the group director.
CCSID	<p>Specify the EBCDIC single-byte application CCSID (Coded Character Set Identifier) configured for this RDBMS subsystem on the RDBMS installation panel DSNTIPF, option 7. <i>(Optional)</i></p>	Refer to the RDBMS vendor documentation for a list of valid CCSIDs.

Parameter	Description	Valid values
APPLNAME	Application name. The APPLNAME used by the target endpoint for passticket generations. (<i>Optional</i>)	A valid value is 1 - 8 characters. If APPLNAME is not specified in the definition statement, no default value is provided and passticket access is disabled. Note: APPLNAME is not required when connecting from the ODBC/JDBC driver.
IDLETIME	If Db2 ZPARM parameter IDTHTOIN is set to a non-zero value set IDLETIME to a value slightly less (10 secs.) than IDTHTOIN. This will also allow product DRDA threads to become inactive. (<i>Db2 for z/OS only</i>)	0-9999 seconds.

Adding an RAA database

Perform required configuration steps to add an RAA database.

About this task

For multiple RAA databases, you must perform configuration steps to define each additional RAA database. The following requirements apply when maintaining multiple RAA databases:

- For the first instance of an RAA database:
 - The database name in the must be RAA1.
 - The target system for the name RAA1 is automatically installed with the RAA maps, as described in [“Installing virtual tables and virtual target maps for RAA access” on page 172.](#)
- For subsequent RAA databases:
 - It is recommended that the database name start with RAA.
 - The target system must start with TSRAA.

Perform the following procedure for each additional RAA database.

Procedure

1. Repeat the database definition in the configuration member and make the following changes:
 - a) Change the NAME value to a unique name (for example, RAA2).
 - b) Change the LOCATION value to reference the Db2 subsystem hosting the RAA database.
 For information about the database definition parameters, see [“Modifying the configuration member for RAA access” on page 173.](#)
2. If the schema (table owner) used by RAA is not ‘DMH’, update the system global variable GLOBAL2.RAA.*database-name*.SCHEMA to the correct schema name for the RAA database tables.
3. Define a new virtual target system using the studio. The name of the virtual target system must start with TSRAA. This can be done in the Data Virtualization Manager studio by selecting the **Create Virtual Target System** in the **Server** tab under the **SQL > Target Systems > DBMS** node of the tree. The connection value in each definition must match the NAME value defined in the DATABASE definition in the configuration member.
4. If required, create authentication information using the AVZDRATH batch utility.

Configuring virtual table rules for RAA

Configure Data Virtualization Manager Server Event Facility (SEF) rules to support multiple instances of the IBM Rational Asset Analyzer (RAA) schema using common virtual table and view definitions.

About this task

To support multiple instances of the RAA schema using common virtual table and view definitions, VTB rules AVZRAATB and AVZRAAVW provide support to process tables starting with RAA_ and views starting with RAAV_.

AVZRAATB

This table rule looks at the base view of a query for double underscores “__” and uses the data after the underscores to update the target subsystem for the query. This rule will also change the schema (or table owner) name of RAA tables from DMH to another value if the global system variable GLOBAL2.RAA.*database-name*.SCHEMA is set with an alternate schema name.

AVZRAAVW

This view rule looks for the double underscores and removes them from the view name to process.

With the rules activated, the Data Virtualization Manager studio can suffix the view names with __SSID for all calls and process multiple instances of the RAA schema using a single set of maps.

These rules must be activated regardless of the number of RAA databases to be enabled.

Use the following procedure to set up these rules.

Procedure

Use the following steps to enable rules AVZRAATB and AVZRAAVW in the VTB ruleset:

- a) In the Data Virtualization Manager server - Primary Option Menu, specify option E, **Rules Mgmt.**
- b) Specify option 2, **SEF Rule Management.**
- c) Enter VTB for **Display Only the Ruleset Named.**
- d) Enable the rule by specifying E and pressing Enter.
- e) Set the rules to Auto-enable by specifying A and pressing Enter.

Setting a rule to Auto-enable activates the rule automatically when the server is re-started.

Configuring authentication for RAA

Configure authentication for communicating with the IBM Rational Asset Analyzer (RAA) database.

About this task

Since RAA is hosted on a z/OS Db2 database, the z/OS credentials that are used to connect to Data Virtualization Manager should also be usable for the z/OS system where Db2 resides. By default, the Data Virtualization Manager server will attempt to use the same user ID that was presented for logon to z/OS for access to the RAA database. To use these credentials, the user ID must have SELECT access on the RAA tables in Db2.

If you choose to specify alternate credentials when communicating between the Data Virtualization Manager server and the RAA database, you must define what credentials to use. A facility is provided in the server to optionally change the logon credentials for a user when accessing the RAA database. To accomplish this, the following tools are provided:

AVZDRATH

A utility that sets encrypted passwords in GLOBALU variables. You can also use this utility to list existing credential information.

AVZEDB2G

An ATH rule that switches credentials when connecting to an RAA database using DRDA. This rule uses AES encrypted passwords stored as GLOBALU system variables.

You can use any of the following options for authentication:

- Use z/OS IDs for authentication
- Add a global default user definition using sample job AVZDRATH and enable ATH rule AVZEDB2G
- Add authentication information for specific mainframe users using sample job AVZDRATH and enable ATH rule AVZEDB2G

If z/OS user IDs and passwords used to connect to the Data Virtualization Manager server are not authorized for the Db2 database hosting the RAA tables, you must define the credentials to use. Use the following procedure.

Procedure

1. Use the sample job AVZDRATH to add a global default user definition or authentication information for specific mainframe users as follows:
 - a) Locate the AVZDRATH member in the *hlq.SAVZCNTL* data set.
 - b) Modify the JCL according to the instructions provided in the AVZDRATH member.

When adding the SYSIN statements that define the alternate credentials for logging in to your RAA database, as instructed in the JCL, make sure to specify the correct DBTYPE. For RAA databases, specify DBTYPE=ZOSDRDA.
 - c) Submit the job.
 - d) Optional: To verify the information stored in the GLOBALU variables and list existing authentication, use the REPORT=SUMMARY statement in the AVZDRATH member and submit the job.
2. Auto-enable the SEF ATH rule SAVZXATH(AVZEDB2G) to switch credentials when connecting to RAA using DRDA. Global variables are used to define alternate authentication credential mapping for the SEF ATH rule.
 - a) On the Data Virtualization Manager server - Primary Option Menu, select option **E** for Rules Mgmt.
 - b) Select option **2** for SEF Rule Management.
 - c) Enter * to display all rules, or ATH to display only authentication rules.
 - d) Set Auto-Enable for the AVZEDB2G rule member by entering A and pressing Enter.

Chapter 6. SQL solution

SQL solution provides ANSI-92 SQL access to mainframe data. You can use this solution to transform mainframe artifacts into a familiar, easy-to-use relational format. With industry standard SQL access, Data Virtualization Manager makes the mainframe data available to business intelligence or analytic applications.

This SQL solution provides access to the following mainframe data sources:

- Software AG Adabas
- IBM® DB2
- IBM® Information Management System (IMS/DB)
- Native VSAM files
- Sequential files
- VSAM via IBM® CICS® Transaction Server

The following table provides an overview of the supported SQL access for each data source type.

Data source type	SELECT	INSERT	UPDATE	DELETE	CALL statement
Adabas	Yes	Yes	Yes	Yes	n/a
Apache Derby	Yes	Yes	Yes	Yes	n/a
Big SQL	Yes	Yes	Yes	Yes	n/a
BigInsights Hadoop	Yes	Yes	Yes	Yes	n/a
CA IDMS	Yes	No	No	No	n/a
CICS COMMAREA	n/a	n/a	n/a	n/a	Yes
dashDB	Yes	Yes	Yes	Yes	n/a
Db2 Direct	Yes	No	No	No	n/a
Db2 for z/OS	Yes	Yes	Yes	Yes	n/a
Db2 for z/OS stored procedure	n/a	n/a	n/a	n/a	Yes
Db2 for Linux, Unix and Windows	Yes	Yes	Yes	Yes	n/a
Db2 for Linux, Unix and Windows stored procedure	n/a	n/a	n/a	n/a	Yes
IMS DBCTL	Yes	Yes	Yes	Yes	n/a
IMS Direct	Yes	No	No	No	n/a
IMS OTMA	n/a	n/a	n/a	n/a	Yes
Informix	Yes	Yes	Yes	Yes	n/a
InfoSphere Federation Server	Yes	Yes	Yes	Yes	n/a
Microsoft SQL Server	Yes	Yes	Yes	Yes	n/a
MQ	Yes	No	No	No	n/a
Natural	n/a	n/a	n/a	n/a	Yes

Table 16. SQL access by data source type (continued)

Data source type	SELECT	INSERT	UPDATE	DELETE	CALL statement
Oracle	Yes	Yes	Yes	Yes	n/a
QMF	Yes	No	No	No	n/a
Sequential data set	Yes	Yes	No	No	n/a
SMF	Yes	No	No	No	n/a
VSAM data set Note: For more information about access to different VSAM file types, see "IBM Data Virtualization Manager for z/OS Interface for VSAM and Sequential files" in the <i>Administration Guide</i> .	Yes	Yes	Yes	Yes	n/a
z/OS SYSLOG	Yes	No	No	No	n/a
zFS	Yes	No	No	No	n/a

Configuring the solution includes one or more of the following tasks:

- Configure the started task JCL by modifying the AVZS (subsystem default ID) member that is in the *hlq.SAVZCNTL* library.
- Configure the server member that is included in *hlq.AVZS.SAVZEXEC(AVZSIN00)*. The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications. See "Creating system data sets" in the *Installation and Customization Guide*.
- Make definition changes in the data provider interface.

For information about configuring the SQL solution, see [Chapter 5, "Configuring access to data sources,"](#) on page 101.

Chapter 7. NoSQL solution

NoSQL solution provides access to any data source supported by IBM Data Virtualization Manager for z/OS through a very robust MongoDB query language. This solution enables developers to write applications that use a JSON-oriented query language that is created by MongoDB to interact with the virtualized data.

JSON (JavaScript Object Notation) is a lightweight data-interchange format that uses readable text to transmit data. JSON is an open standard and provides an alternative to XML. For more information, see <https://www.ietf.org/rfc/rfc4627.txt>.

After you establish a connection from a MongoDB client, all supported MongoDB methods can be used to query and update the data that is defined to the server. This exposes existing data as JSON documents and communicates with Mongo applications. Since this solution is about access to existing data, not all methods in MongoDB query language are applicable.

To configure and verify access to data by using the MongoDB query language:

- Modify the Data Virtualization Manager configuration member
- Configure data sources

Modifying the Data Virtualization Manager configuration member

To start using the MongoDB query language with the Data Virtualization Manager server, enable the **MONGODB**, **MONGOPORT**, and **MONGOZSQLDBNAME** parameters and connect to the server with your choice of MongoDB client.

Before you begin

The server must be installed.

About this task

Configure a data source that is supported by the Data Virtualization Manager server, create your source libraries, and map the virtual collections.

The Data Virtualization Manager configuration member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and copied to *hlq.AVZS.SAVZEXEC(AVZSIN00)* by the job in the AVZGNMP1 member for you to make your local modifications.

Procedure

1. In the AVZSIN00 member, locate the comment “ENABLE MONGO API.”
2. To enable the MongoDB parameters, change `if DontDoThis` to `if DoThis`.

```
if DoThis then
  do
    "MODIFY PARM NAME(MONGODB)           VALUE(YES) "
    "MODIFY PARM NAME(MONGOPORT)        VALUE(1207) "
    "MODIFY PARM NAME(MONGOZSQLDBNAME)   VALUE(DVS) "
  end
```

The following table lists the required parameters:

Parameter	Description	Valid values
MONGODB	Activates support for MongoDB query language	NO Support is not active. YES Activate support.
MONGOPORT	TCP/IP port number.	Numeric value. Default is 1207.
MONGOZSQLDBNAME	Database name for SQL engine requests.	Text string. Default is "AVZ".

The following table lists optional parameters:

Parameter	Description	Valid values
MONGOSSLPORT	TCP/IP main SSL port number.	Default is 0.
MONGOTRACELEVEL	Text trace message verbosity.	Default is normal.
MONGOCURSORLIFE	Cursor wait time limit in seconds.	Default is 600 seconds.
MONGOMAXMSGLENGTH	Maximum message length, in MB.	Default is 48 MB.

MongoDB interface examples

JavaScript, C#, and BIRT interface examples show how to use MongoDB query language with the Data Virtualization Manager server.

NoSQL for MongoDB interface example

This section describes an example of a VSAM file containing information about zip codes and cities.

The following copybook describes a VSAM file that contains information about zip codes and cities.

```
*****
* RECORD LAYOUT
*
01 RECORD.
  05 ZIP PIC X(5).
  05 CITY PIC X(16).
  05 STATE PIC X(2).
  05 POPULATION PIC 9(6).
  05 LOC_X PIC 9(3)V9(6).
  05 LOC_Y PIC 9(2)V9(6).
```

This is a sample of how data looks in the VSAM file:

```
**** Top of data ****
01001AGAWAM MA01533807262273942070206
01002CUSHMAN MA03696307251565042377017
01005BARRE MA00454607210835442409698
01007BELCHERTOWN MA01057907241095342275103
01008BLANDFORD MA00124007293611442182949
01010BRIMFIELD MA00370607218845542116543
01011CHESTER MA00168807298876142279421
```

After you virtualize this VSAM structure in a virtual collection, you can retrieve this information as a JSON document by using MongoDB query language. This example uses the MongoDB shell:

```
C:\mongodb-win32-x86_64-2.4.5\bin>mongo host:27017
MongoDB shell version: 2.4.5
connecting to: host:27017/test
```

```

> show dbs
DVS      1GB

> use DVS
switched to db DVS

> db.us_zipcodes.find({"city":"AGAWAM"})

{
  "zip" : "01001",
  "city" : "AGAWAM",
  "state" : "MA",
  "population" : 15338,
  "loc_x" : 72.622739,
  "loc_y" : 42.070206
}

```

Where:

- database – similar to schema in SQL, in the example “DVS”
- collection – similar to the table, in the example “us_zipcodes”
- find() method – similar to SQL SELECT statement with predicate specified in JSON format

JavaScript and Node.js

Download the MongoDB JavaScript driver from the MongoDB website or use the Mongo shell to run JavaScript.

The following is an example of how to use Node.js with the NoSQL interface for Mongo.

```

/*jshint node:true*/

// app.js
// This file contains the server side JavaScript code for your application.
// This sample application uses express as web application framework
// (http://expressjs.com/), and jade as template engine (http://jade-lang.com/).

var express = require('express');
var mongo = require('mongodb');

var mongonative = require('mongodb').MongoClient,
    assert = require('assert');

var dbURI = "mongodb://dvs_host_name:port/db_name";

// setup middleware
var app = express();
//app.use(app.router);
//app.use(express.errorHandler());
app.use(express.static(__dirname + '/public')); //setup static public directory
app.set('view engine', 'jade');
app.set('views', __dirname + '/views'); //optional since express defaults to
  CWD/views

// render index page
app.get('/', function (req, res) {
  res.render('index');
});

mongonative.connect(dbURI, function (err, db) {
  console.log("connect with native mongodb driver");
  assert.equal(null, err);
  console.log("connected with native mongodb driver");

  app.get('/docs', function (req, res) {
    console.log('In GET /docs');
    var collection = db.collection('us_zipcodes');
    var adminDb = db.admin();
    if (err) throw err;

    /*adminDb.ping(function (err, pingResult) {
      console.log('In adminDb functions: ', pingResult);
      assert.equal(null, err);
      db.close();
    });*/
  });
});

```

```

/* adminDb.listDatabases(function (err, dbs) {
  console.log('In admindb listdbs: ', dbs);
  assert.equal(null, err);
  assert.ok(dbs.databases.length > 0);

  db.close();
}); */

/* adminDb.command({buildInfo:1}, function(err, info) {
  console.log('In buildinfo: ', info);
  assert.equal(null, err);
  db.close();
});*/

// Return the information of all collections, using the callback format
/* db.collectionsInfo(function (err, cursor) {
  // Turn the cursor into an array of results
  cursor.toArray(function (err, items) {
    assert.ok(items.length > 0);
    var json = JSON.stringify(items);
    res.end(json);
    db.close();
  });
});*/
/* db.collectionNames(function (err, items) {
  assert.ok(items.length > 0);
  var json = JSON.stringify(items);
  res.end(json);
  db.close();
});*/

//db.collections(function (err, docs) {
//doesn't work with DVS, use above instead:
//collection.find({}).toArray(function (err, docs) {
//collection.count(function (err, docs) {
//collection.find({zip:"01001"}).toArray(function (err, docs) {
//collection.distict(city, function (err, docs) {
//collection.findOne(function (err, docs) {
//collection.find({ zip: { "$gte": "96700", "$lte": "96799" } },
//  { "sort": "population" }).toArray(function (err, docs) {
collection.find({ zip: { "$in": ["96701", "96702", "96703"] } }).
toArray(function (err, docs) {

  console.log('In find city: ', docs);
  assert.equal(err, null);
  var json = JSON.stringify(docs);
  res.end(json);
  db.close();
});
});
});

// There are many useful environment variables available in process.env.
// VCAP_APPLICATION contains useful information about a deployed application.
var appInfo = JSON.parse(process.env.VCAP_APPLICATION || "{}");
// TODO: Get application information and use it in your app.

// VCAP_SERVICES contains all the credentials of services bound to
// this application. For details of its content, please refer to
// the document or sample of each service.
var services = JSON.parse(process.env.VCAP_SERVICES || "{}");
// TODO: Get service credentials and communicate with Bluemix services.

// The IP address of the Cloud Foundry DEA (Droplet Execution Agent)
// that hosts this application:
var host = (process.env.VCAP_APP_HOST || 'localhost');
// The port on the DEA for communication with the application:
var port = (process.env.VCAP_APP_PORT || 3000);
// Start server
app.listen(port, host);
console.log('App started on port ' + port);

```

C#

Download and install the C# driver from the MongoDB website.

The following is a sample class that calls MongoDB:

```

using MongoDB.Bson.IO;
using MongoDB.Bson.Serialization;
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson.Serialization.Conventions;
using MongoDB.Bson.Serialization.IdGenerators;
using MongoDB.Bson.Serialization.Options;
using MongoDB.Bson.Serialization.Serializers;
using MongoDB.Driver.Wrappers;

namespace MongoClass
{
    public class MongoClassWrapper
    {
        public string mongoreturns;
        public MongoClassWrapper(string host, string database_name,
            string collection_name, IMongoQuery query)
        {
            MongoServerSettings serversettings = new MongoServerSettings();
            serversettings.To invoke from your
                .NET application use following:GetDatabase(database_name);

            try
            {
                MongoClient testclient =
                    MongoClient.FromUri(new Uri(host));
                MongoDB database = testclient.GetDatabase(database_name);
                MongoCursor testcursor =
                    database.GetCollection(collection_name).Find(query);
                mongoreturns = testcursor.ToJson();
            }
            catch (Exception ex) { }
            finally
            {
                server.Disconnect();
            }
        }
    }
}

```

Creating reports from a MongoDB data source

You can use the Business Intelligence Reporting Tool (BIRT) to create reports from a MongoDB data source.

Before you begin

- Eclipse Kepler (minimum) <http://www.eclipse.org/kepler/>
- BIRT (Business Intelligence and Reporting Tools) reports <http://download.eclipse.org/birt/downloads/>

Procedure

1. Install BIRT into an existing Eclipse Kepler. In Eclipse, select **Help > Install New Software**, select BIRT update site: <http://download.eclipse.org/birt/update-site/4.3>

Note: If you download BIRT with KEPLER from the BIRT site, you can skip this installation step.

2. Restart Eclipse, and follow the *BIRT Report Developer Guide*.
3. Create a new project and then new report (select blank report).
4. Add data source – select MongoDB data source.
5. Create a new data set by right clicking on DataSet node. Drag and drop the data set onto report.
6. Select **Preview** tab on the bottom of the report to run a query.

Supported MongoDB query language

Data Virtualization Manager server supports the following MongoDB query language collection, cursor, operator, aggregation, and administrative functions.

Collection

MongoDB	Example
count() collection.count()	<pre>db.us_zipcodes.count()</pre> or database command: <pre>db.runCommand({count:"us_zipcodes"})</pre>
distinct() collection.distinct(field,query)	<pre>db.us_zipcodes.distinct("city",{state:"GA"})</pre> or database command: <pre>db.runCommand({ distinct: "us_zipcodes", key: "city", query: { "state": "GA" } })</pre>
find() collection.find(query,fields)	<pre>db.us_zipcodes.find({"state": "CA"}, {"zip": 1, "city": 1}).sort({"zip": -1})</pre>
findOne() collection.findOne(query,fields)	<pre>db.us_zipcodes.findOne({"state": "CA"}, {"zip": 1, "city": 1})</pre>
getIndexes() collection .getIndexes()	<pre>db.us_zipcodes.getIndexes()</pre>
insert() collection.insert	<pre>db.us_zipcodes_update.insert({ "zip": "02001", "city" : "AGAWAM", "state" : "MA", "population" : 15338, "loc_x" : 72.622738, "loc_y" : 42.070205});</pre>
remove() collection.remove	<pre>db.us_zipcodes_update.remove({ "zip": "02001" });</pre>
update() collection.update	<pre>db.us_zipcodes_update.update({ "zip": "02001" }, { \$set: { "city" : "CUSHMAN", "population" : 36709, "loc_x" : 72.51565, "loc_y" : 42.377016999999995 } }, false, false);</pre>
drop() collection.drop	Not supported
ensureIndexes() collection.ensureIndexes	Not supported The collection.EnsureIndex() API allows to create document indexes on fields. In this version, rely only the primary keys/indexes defined in data sources.

MongoDB	Example
<u>save()</u> collection.save	Not supported In MongoDB the save () document API at the collection level does not rely on preexistence of the collection. At the first insert () or save (), if the collection does not exist MongoDB server allocate a new one based on the collection name. Since NoSQL relies on existing data set that we do not support the dynamic creation of collection

Cursor

MongoDB	Example
<u>count()</u> cursor.max(IndexBounds)	db.us_zipcodes.find({"state": "CA", "city": "LOS ANGELES"}).count()
<u>limit()</u> cursor.limit(n)	db.us_zipcodes.find().limit(10);
<u>max()</u> cursor.max (IndexBounds))	db.us_zipcodes.find({"state": "CA"}, {"zip": 1, "city": 1}).sort({"zip": -1})
<u>findOne()</u> collection.findOne(query, fields)	<pre> cursor = db.us_zipcodes.find({"city": "IRVINE" }).max({"zip": "92718" }) {"zip": "16329", "city": "IRVINE", "state": "PA", "population": 479, "loc_x": 79.28630299999999, "loc_y": 41.843283 } {"zip": "92714", "city": "IRVINE", "state": "CA", "population": 60654, "loc_x": 117.79892799999999, "loc_y": 33.6876408 } {"zip": "92715", "city": "IRVINE", "state": "CA", "population": 30690, "loc_x": 117.82125099999999, "loc_y": 33.6508841 } </pre> <pre> SELECT * FROM shadow.us_zipcodes WHERE (city = 'IRVINE' AND zip ('92718')) </pre>
<u>min()</u> cursor.min(IndexBounds)	<pre> cursor = db.us_zipcodes.find({"city": "IRVINE" }).min({"zip": "92718" }) {"zip": "92718", "city": "IRVINE", "state": "CA", "population": 1, "loc_x": 117.71147599999999, "loc_y": 33.6581796 } {"zip": "92720", "city": "IRVINE", "state": "CA", "population": 23474, "loc_x": 117.76553299999999, "loc_y": 33.7074953 } </pre>
<u>skip()</u> cursor.skip(n)	db.us_zipcodes.find({"city": "IRVINE" }).skip(1)
<u>sort()</u> cursor.sort()	db.us_zipcodes.find({"state": "CA"}, {"zip": 1, "city": 1}).sort({"zip": -1})

Operators

MongoDB	Example
Comparison	
greater than: \$gt less than: \$lt greater than or equal to: \$gte less than or equal to: \$lte cursor.find({field:{\$gt:value}})	<pre>db.us_zipcodes.find({"zip":{"\$gte":"96700", "\$lte":"96799"}}).sort({"population":1}) db.us_zipcodes.find({"zip": {"\$gt": "96700", "\$lt": "96799" } }).sort({"population": 1 })</pre>
\$in cursor .find({field:\$in: [value1,value2,value3...]}})	<pre>db.us_zipcodes.find({"zip": { \$in: [96701,96702,96703]}})</pre>
\$ne Selects the documents where the value of the field is not equal to the specified value. cursor .find({field:\$ne:value}})	<pre>db.us_zipcodes.find({"state": { \$ne: "AL" } }, {"state":1}).limit(10)</pre>
\$nin Selects the documents where the field value is not in the specified array, or the field does not exist. cursor.find({field:\$nin: [value1,value2,value3...]}})	<pre>db.us_zipcodes.find({"zip": { \$nin: [96701, 96702, 96703] } }).limit(10)</pre>
Logical	
\$and {\$and:[{expression1},{expression2},...]}	<pre>db.us_zipcodes.find({ \$and: [{"population": { \$gt: 30000}}, {"state":"MS"}]}).sort({"population": 1})</pre>
\$or {\$or: [{ (expression1) }, { (expression2) }, ... , { (expressionN) }] }	<pre>db.us_zipcodes.find({"state": "MS", \$or: [{"population": { \$gt: 35000 } }, { "loc_y": { \$gt: 34 }}] })</pre>
\$not {field: { \$not: { (operator-expression) } } }	<pre>db.us_zipcodes.find({ \$and: [{"population": { \$not:({ \$gt: 30000 }) }, {"state": "MS" }]} }).sort ({"population": 1})</pre>
Element \$exist \$type	Not supported
Evaluation \$mod \$regex \$where	Not supported
Geospatial	Not supported

Aggregation

Limitations: Only particular combinations of pipeline operators are allowed. For example, "\$project" or "\$limit" cannot follow "\$group".

Example:

```

db.us_zipcodes.aggregate([
  {$match: {$loc_x: {$gte: 90.89}}},
  {$project: {
    state: "$state",
    population_thnd: {$divide: ["$population", 1000]}}
  },
  {$group: {
    _id: "$state",
    population_thnd: {$sum: "$population_thnd"}}
  },
  {$sort: {_id: 1}},
  {$skip: 1},
  {$limit: 2}
])

```

Administrative functions

Supported?	MongoDB	Example
No	compact	
No	text	
No	mapReduce	
Yes (partially)	getLastError	<pre>db.getLastError()</pre> or <pre>db.runCommand({ getLastError:1 })</pre>
No	Sharding Commands	
Yes	Authentication Commands	
No	Geospatial Commands	
Yes	whatsmyuri	<pre>db.runCommand({whatsmyuri:1})</pre>
Yes	resetError	<pre>db.runCommand({resetError:1})</pre>
Yes	isMaster()	<pre>db.isMaster()</pre>
Yes	listDatabases()	<pre>db.adminCommand("listDatabases")</pre> or <pre>db.runCommand({listDatabases:1})</pre>
Yes (limited)	<code>db.system.namespaces.find()</code>	<pre>db.system.namespaces.find()</pre>
Yes	ping	<pre>db.runCommand({ ping:1})</pre>
Yes	buildinfo	<pre>db.runCommand({ buildinfo:1})</pre>
Partially (see aggregation)	group	<pre>db.runCommand({ group: { ns: (namespace), key: (key), \$reduce: (reduce function), \$keyf: (key function), cond: query, finalize: (finalize function) } })</pre>
No	drop	
No	getCmdLineOpts	

Supported?	MongoDB	Example
No	fsync	
No	filemd5	
No	collMod	
Yes	resync	
Yes	getPrevError	
Yes	replSet*	<pre>db.runCommand({ replSetGetStatus: 1, forShell: false/true })</pre>
Yes	isdbgrid()	<code>db.runCommand({ isdbgrid:1})</code>
No	renameCollection	
No	dropIndexes	
Yes (partially)	setParameterandgetParameter	
Yes	connPoolStats	
Yes (partially)	cursorInfo	
	dataSize	
Yes (partially)	aggregate	
	profile	
Yes	listCommands	
Yes (partially)	getLog	
No	top	
Yes (partially)	hostInfo	
Yes (partially)	serverStatus	
Yes (partially)	dbStats()	<code>db.runCommand({dbStats: 1,scale: 1})</code>
Yes (partially)	collStats	<code>db.runCommand({ collStats: 1, scale: 1})</code>
No	shutdown	
No	findAndModify	

Chapter 8. z/OS Connect solution

IBM Data Virtualization Manager for z/OS provides IBM z/OS Connect Enterprise Edition direct access to data with Select, Insert, Update and Delete functions using its RESTful interface. Data Virtualization Manager uses its web service data integration created via the IBM Data Virtualization Manager studio that is invoked via z/OS Connect and its RESTful interface.

Any data that has been virtualized via Data Virtualization Manager is available to z/OS Connect. Data Virtualization Manager deploys its Service Provider into the Unix System Services (USS) environment on Data Virtualization Manager to establish communications via WebSphere Optimized Local Adapter (WOLA) to a Data Virtualization Manager server instance in USS.

Configuring the z/OS Connect solution

To configure z/OS Connect, you need to configure the Data Virtualization Manager Service Provider on zcEE, the server started task JCL, and the configuration member.

Before you begin

The following products must be installed:

- IBM z/OS Connect Enterprise Edition

A WebSphere Liberty angel STC must be running for communications between the zcEE server and the Data Virtualization Manager server via WOLA.

For zcEE installation information, Data Virtualization Manager currently supports integration with IBM z/OS Connect Version 3.0 using OpenAPI 2.

Note: Make note of the directory path for your IBM z/OS Connect Enterprise Edition installation. You will use this directory path to modify the instructions given in the examples.

- Data Virtualization Manager server
- Data Virtualization Manager studio
- The data source(s) must be configured and virtualized in Data Virtualization Manager server (see "Configuring access to data sources" in the *Installation and Customization Guide*.)

About this task

For information about configuring the z/OS Connect solution, see the following topics.

1. [“Configuring the Data Virtualization Manager Service Provider on zcEE” on page 191](#)
2. [“Creating an IBM z/OS Connect Enterprise Edition server instance” on page 193](#)
3. [“Configuring the server started task JCL” on page 194](#)
4. [“Modifying the Data Virtualization Manager configuration member” on page 194](#)
5. [“Starting the Data Virtualization Manager server and z/OS Connect server” on page 195](#)

Configuring the Data Virtualization Manager Service Provider on zcEE

You must configure the Data Virtualization Manager Service Provider on zcEE in z/OS Connect to provide communication between z/OS Connect and the Data Virtualization Manager server.

About this task

This procedure uses the Data Virtualization Manager Service Provider on zcEE installation file, which is uploaded to IBM Fix Central page.

Procedure

1. Go to IBM Fix Central page - <https://www.ibm.com/support/fixcentral/>.
2. Under **Find Product** section, in the **Product Selector** list, select IBM Data Virtualization Manager for z/OS, and then press **Enter**.
3. In the **Installed Version** list, select a required version number.
4. In the **Platform** list, select **All** and then click **Continue**.
The Identify fixes page is displayed.
5. Click **Browse for Fixes** option and then click **Continue**.
6. Select **Service Provider** and then click AVZ-service-provider-*<version>*.zip file to download.
7. Extract the contents of the zip file to a local directory and locate the file `com.rs.dv.zosconnect.provider.feature_1.0.0.xxxxxxxxxxxx.esa`.
8. Rename the file to `com.rs.dv.zosconnect.provider.feature_1.0.0.esa`.
9. Copy the file to a UNIX System Services (USS) directory using an FTP utility.
10. Connect to UNIX System Services (USS) and install the Data Virtualization Manager Service Provider on zcEE, as follows:

- a) The **installUtility** is found in the z/OS Connect installation directory /*<zosconnect_install>*/wlp/bin, where *<zosconnect_install>* is the path directory for your z/OS Connect installation.

For example, change to the following directory:

```
cd /usr/lpp/IBM/zosconnect/v2r0/wlp/bin
```

- b) Set JAVA_HOME= environment variable to the path to your 64-bit IBM Java SDK.

For example:

```
export JAVA_HOME=/my_directory_path/java/IBM/J8.0_64
```

- c) Set WLP_USER_DIR= environment variable to the location where you want your server instances and user features to be stored.

For example:

```
export WLP_USER_DIR=/my_directory_path/var/zosconnect
```

- d) Issue the install command as follows:

```
./installUtility install  
/my_directory_path/com.rs.dv.zosconnect.provider.feature_1.0.0.esa
```

The output from the install command looks like this:

```
Step 1 of 2: Installing dvsProvider ...  
Step 2 of 2: Cleaning up temporary files ...  
  
All assets were successfully installed.  
  
Start product validation...  
Product validation completed successfully.
```

Note: To uninstall the Service Provider on zcEE (for example, before installing a new version), use the **installUtility** like this:

```
./installUtility uninstall com.rs.dv.zosconnect.provider.feature
```

The Service Provider on zcEE has been successfully installed into z/OS Connect.

Creating an IBM z/OS Connect Enterprise Edition server instance

To prepare to use the z/OS Connect solution, create an IBM z/OS Connect Enterprise Edition server instance.

About this task

Perform this task to create a z/OS Connect server instance in UNIX System Services (USS).

As part of this task, the z/OS Connect server instance must have read/write authority to the directories and files where the server was created. There are various ways to handle USS file authority. One method, which is provided in the following procedure, is to change the authority of the z/OS Connect server instance to all directories and files so that the GROUP has read/write authority. (The z/OS Connect server instance will run under the same GROUP defined to these directories/files.) It is recommended that you define your USS file authority according to your company requirements or standards.

Procedure

1. Perform security setup as needed for your server environment.
2. Connect to UNIX System Services (USS), and change the directory to the `/bin` directory. For example:
`cd /usr/lpp/IBM/zosconnect/v2r0/bin.`
3. Create the z/OS Connect server instance by issuing the following command:

```
zosconnect create <server_name> --template=zosconnect:default
```

This creates the z/OS Connect server instance in `/var/zosconnect/servers`. This is a folder in the directory with your `<server_name>`.

Note: See the [IBM documentation](#) for more details on how to create a z/OS Connect server.

4. Change the authority of the z/OS Connect server instance to have read/write access to all directories and files where the server was created, which you can do by issuing the following command:

```
chmod -R g+w <top-level-server-dir>
```

For example:

```
chmod -R g+w /var/zosconnect/servers/AVZ1test
```

where `AVZ1test` is the `server_name`.

Note: It is recommended that you define your USS file authority according to your company requirements or standards.

The z/OS Connect server instance is ready for configuration.

5. Modify the `server.xml` file, as follows:
 - a) Locate the `server.xml` file in the following directory: `/var/zosconnect/servers/<server_name>/server.xml`
 - b) Define the ports that the z/OS Connect server will listen on. These ports need to be unique.

For example:

```
httpPort="12080" httpsPort="12081"
```

- c) Define `wolaGroup="your_wola_name"`. The value must match your `WNAME` value in your `AVZSIN00` file. This is the name the z/OS Connect server will run under in SDSF.
- d) Define `registerName="your_registry_name"`. The value must match your `RNAME` value in your `AVZSIN00` file. There are two locations where the `registerName=` needs to be defined in the `server.xml` file.

The z/OS Connect server instance is now ready.

Configuring the server started task JCL

Create a WebSphere Optimized Local Adaptor (WOLA) PDSE data set and copy UNIX System Services (USS) modules into it. The WOLA data set contains the modules used to communicate between a Data Virtualization Manager server and a z/OS Connect server.

Procedure

1. Allocate a PDSE data set with 30 tracks. Using the Data Virtualization Manager server naming conventions is recommended. For example: *HLQ.WOLA.API.V2.NA*

```
Average record unit
Primary quantity . . 30
Secondary quantity . 2
Directory blocks . . 15
Record format . . . U
Record length . . . 0
Block size . . . . . 32760
Data set name type . LIBRARY
```

2. Change directory to /<zosconnect_install>/wlp/clients/zos. For example:

```
cd /usr/lpp/IBM/zosconnect/v2r0/wlp/clients/zos
```

3. Issue this command to copy modules from USS into PDSE:

```
cp -Xv ./* "'<data.set>'"
```

Where <data.set> would be the data set you allocated. For example: *HLQ.WOLA.API.V2.NA*

Note: You could have several z/OS Connect servers and/or have several Data Virtualization Manager servers and they could all use the one WOLA PDSE data set.

4. Add the WOLA PDSE data set you created to the AVZRPLB ddname in the server started task JCL. For example:

```
//AVZRPLB DD DISP=SHR,DSN=&HLQ..SAVZRPC
// DD DISP=SHR,DSN=&HLQ.WOLA.API.V2.NA
```

Modifying the Data Virtualization Manager configuration member

Configure the z/OS Connect parameters in the AVZSIN00 configuration member. This enables communication between the Data Virtualization Manager server and the z/OS Connect server.

Procedure

Configure the z/OS Connect parameters.

You need to add a **DEFINE ZCPATH** command which defines the path to the z/OS Connect server. For example:

```
if DoThis then
do
  "DEFINE ZCPATH",
  " NAME(your_zconnect_server_name)",
  " RNAME(your_registry_name)",
  " WNAME(your_wola_name)"
```

- a) Make sure the RNAME matches the registerName in your server.xml file. This is in two locations in the server.xml file and will need to be changed in both.
- b) Make sure the WNAME matches the wolaGroup in your server.xml file.

Parameter	Description	Valid values
NAME(<i>defname</i>)	This required parameter is used to provide a specific definition name.	A definition name may be up to 20 characters long, the first character must be alpha or national, and the rest must be alphanumeric or national, and the ‘_’ and ‘.’ are also allowed. The value is always treated as uppercase.
RNAME(<i>register-name</i>)	This required parameter is used to supply the register name for the connection to z/OS Connect. The name supplied here must match the RegisterName value on the localAdaptersConnectService description in the z/OS Connect server.xml file.	A register name may be up to 12 characters long, and the first character must be alpha or national, and the rest must be alphanumeric or national. The ‘_’ and ‘.’ are also allowed. The value is <i>not</i> treated as uppercase. Note: WOLA requires that each register name within a region is unique. The Data Virtualization Manager server interface to z/OS Connect ensures that the register name is unique within all connection definitions.
WNAME(<i>WOLA-name</i>)	This required parameter is used to supply the first part of the WOLA name for the connection to z/OS Connect.	The value may be up to 8 characters long, the first character must be alpha or national, and the rest must be alphanumeric or national. The value is treated as uppercase. The value supplied here must match the ‘wolaGroup’ parameter of the ‘zosLocalAdapters’ entry in the z/OS Connect server.xml file.

Note: Additional parameters for customizing the connection are described in “Data Virtualization Manager configuration member parameters” on page 200. Contact Technical Support for instructions on how to use these parameters.

Starting the Data Virtualization Manager server and z/OS Connect server

To use z/OS Connect, you need to start the Data Virtualization Manager server and z/OS Connect server. The start up process establishes a connection between the two servers.

Procedure

1. After making the modifications for z/OS Connect, re-start the Data Virtualization Manager server using the following console command:

```
S VDBS
```

Note: If you want to stop the server, issue the following console command: P VDBS

2. To start a z/OS Connect server:

```
S BAQSTRT,JOBNAME=your_jobname,PARMS='<server name> --clean'
```

Note: To stop a z/OS Connect server, use a command like the following, /p <server name>.

After starting the servers, the z/OS Connect server log shows the connection to the Data Virtualization Manager server and indicates that the "Data Integration" Web Service Operations have loaded.

You will see the following messages in the Data Virtualization Manager server log that indicate it is connected to the z/OS Connect server:

```
ZCPHASY subtask is active  
ZCPRHUPR subtask is active  
ZCPRHLWR subtask is active
```

Results

You are now ready to use the z/OS Connect solution.

z/OS Connect

This section contains detailed z/OS Connect configuration information, including configuring the Data Virtualization Manager server by using the **DEFINE ZCPATH** command, defining Data Virtualization Manager configuration member parameters, and configuring the Service Provider on zcEE.

Data Virtualization Manager server configuration

The **DEFINE ZCPATH** command can be used to define a connection to a specific z/OS Connect region (server).

DEFINE ZCPATH

Normally, you would place the required **DEFINE ZCPATH** commands in the AVZSIN00 member for a specific Data Virtualization Manager region. Note that the Data Virtualization Manager will (by default) retry a failed connection (caused by, for example, the z/OS Connect server being not active).

Make sure the RNAME matches the `registerName` in your `server.xml` file. Make sure the WAME matches the `wolaGroup` in your `server.xml` file.

Security

When a user is being checked to see if they are allowed to execute the relevant command, the resource name of ZCPATHS is used. The ZCPATHS is checked by the internal rules and no external security is involved.

```
Define ZCPATH  
  NAME(defname)  
  RNAME(register-name)  
  WNAME(WOLA-name)  
  [ WNAMEX(WOLA-namex) ]  
  [ STATUS({ Active | Inactive } ) ]  
  [ RETRY( { NO | interval } ) ]  
  [ MINCR(n) ]  
  [ MAXCR(n) ]  
  [ MAXQR(n) ]  
  [ MAXCACT({ ERROR | QUEUE} ) ]  
  [ MAXPRQ(n) ]  
  [ MAXQT(n) ]  
  [ USERID(userid) ]  
  [ MAXPTIME(n) ]  
  [ MAXRECS(n) ]  
  [ MAXRLEN(n) ]  
  [ USEACTUID( { NO | YES } ) ] }
```

The required parameters (**NAME**, **RNAME**, **WNAME**) for **DEFINE ZCPATH** are described in [“Modifying the Data Virtualization Manager configuration member”](#) on page 194.

The optional parameters for the **DEFINE ZCPATH** command are as follows:

Parameter	Description	Valid values
NAME(<i>defname</i>)	This required parameter is used to provide a specific definition name.	A definition name may be up to 20 characters long, the first character must be alpha or national, and the rest must be alphanumeric or national, and the ‘_’ and ‘.’ are also allowed. The value is always treated as uppercase.
RNAME(<i>register-name</i>)	This required parameter is used to supply the register name for the connection to z/OS Connect. The name supplied here must match the RegisterName value on the localAdaptersConnectService description in the z/OS Connect server.xml file.	A register name may be up to 12 characters long, and the first character must be alpha or national, and the rest must be alphanumeric or national. The ‘_’ and ‘.’ are also allowed. The value is <i>not</i> treated as uppercase. Note: WOLA requires that each register name within a region is unique. The Data Virtualization Manager server interface to z/OS Connect ensures that the register name is unique within all connection definitions.
WNAME(<i>WOLA-name</i>)	This required parameter is used to supply the first part of the WOLA name for the connection to z/OS Connect.	The value may be up to 8 characters long, the first character must be alpha or national, and the rest must be alphanumeric or national. The value is treated as uppercase. The value supplied here must match the ‘wolaGroup’ parameter of the ‘zosLocalAdapters’ entry in the z/OS Connect ‘server.xml’ file.
STATUS({ ACTIVE INACTIVE }) (Optional)	This optional parameter is used to supply the status of the connection.	If omitted, the default is ‘ACTIVE’. STATUS=INACTIVE can be used to create a definition, but not attempt to connect to the z/OS Connect region.
RETRY({ NO <u>interval</u> }) (Optional)	This Optional parameter is used to supply an overriding retry interval (in seconds) or prevent retry (by specifying NO).	Integer (in seconds) If this parameter is not specified, then the value of the ZCONNECTPRETRY initialization parameter is used.
MINCR(<i>n</i>) (Optional)	This optional parameter allows you to indicate the minimum number of worker tasks that will be pre-started or left active to service requests.	The valid range is 1 to 10. If this parameter is not specified, then the value of the ZCONNECTPMINCR initialization parameter is used.

Parameter	Description	Valid values
MAXCR(<i>n</i>) (<i>Optional</i>)	This optional parameter allows you to indicate the maximum number of worker tasks that can exist at any one time (== concurrent requests).	The valid range is 5 to 20. The value must be equal to or greater than the MINCR value. If this parameter is not specified, then the value of the ZCONNECTPMAXCR initialization parameter is used. See the notes associated with the ZCONNECTPMAXCR initialization parameter.
MAXQR(<i>n</i>) (<i>Optional</i>)	This optional parameter allows you to indicate the maximum number queued requests (awaiting worker tasks) that can exist at any one time.	The valid range is 0 to 50. If this parameter is not specified, the value of the ZCONNECTPMAXQR initialization parameter is used. See the notes associated with the ZCONNECTPMAXQR initialization parameter.
MAXCACT({ ERROR QUEUE }) (<i>Optional</i>)	This optional parameter allows you to indicate the action to take if a new request arrives while the server is already processing the maximum number of concurrent requests as set by the MAXCR parameter.	ERROR means that an error is returned to the requestor, and QUEUE means that the request will be queued (see the MAXQT parameter). If this parameter is not specified, then the value of the ZCONNECTMAXQACT initialization parameter is used.
MAXPRQ(<i>n</i>) (<i>Optional</i>)	This optional parameter allows you to set how many requests a given processing task may handle before it is recycled. This can be used to prevent any storage leaks during processing of requests.	The range is 0 to 1000. If the value is 0, no recycling is done. If this parameter is not specified, then the value of the ZCONNECTPMAXPRQ initialization parameter is used.
MAXQT(<i>n</i>) (<i>Optional</i>)	This optional parameter allows you to set a maximum queue time when MAXCACT is QUEUE.	Integer (in seconds) If the value (in seconds) is exceeded, then the request is failed (as if MAXCACT=ERROR was specified). The range is 0 to 600. If the value is 0, a request is queued indefinitely. If this parameter is not specified, then the value of the ZCONNECTPMAXQT initialization parameter is used.

Parameter	Description	Valid values
USERID(<i>userid</i>) (Optional)	This optional parameter allows you to nominate a default userid for the request. If unable to determine the userid associated with the request, this value is used.	<p>The value may be up to 8 characters long, and is in 'PDS member name' format – the first character must be alpha or national, and the rest must be alphanumeric or national. The value is treated as uppercase. Note that a value of '*' is also allowed (see below).</p> <p>If this parameter is not specified, the value of the ZCONNECTUSERID initialization parameter is used.</p> <p>A value of * means that the region userid will be used.</p> <p>Note: This userid is the one applied if the value for SNAME is matched with the incoming service name.</p>
MAXPTIME(<i>n</i>) (Optional)	This optional parameter allows you to set a maximum processing time (in seconds) for a request.	<p>If the time limit is exceeded, an error is returned. The range is 0 to 120.</p> <p>A value of 0 means that no time limit is imposed. If this parameter is not specified, the value of the ZCONNECTPMAXPTIME initialization parameter is used.</p>
MAXRECS(<i>n</i>) (Optional)	This optional parameter allows you to set a maximum number of returned records for a request. If the value is exceeded, an error is returned.	<p>The range is 1 to 10,000.</p> <p>If this parameter is not specified, then the value of the ZCONNECTPMAXRECS initialization parameter is used.</p>
MAXRLEN(<i>userid</i>) (Optional)	This optional parameter allows you to set a maximum length of the reply for a request. If the value is exceeded, an error is returned.	<p>The range is 100 to 50,000,000.</p> <p>If this parameter is not specified, then the value of the ZCONNECTPMAXRLEN initialization parameter is used.</p>

Parameter	Description	Valid values
USEACTUID({ NO YES }) (Optional)	This optional parameter allows you to indicate whether the actual userid (if available) is to be used.	NO means that the command-nominated (or defaulted) userid is to be used. YES (The default if this parameter is not specified on the command) means that the actual userid (if available) is to be used. If the actual userid is not available, processing continues as if this operand is NO. Note: This USEACTUID value is used if the value for SNAME is matched with the incoming service name.

Data Virtualization Manager configuration member parameters

Except for the ZCONNECT parameter, the value of these z/OS Connect parameters can be changed at any time.

However, while the parameter values can be altered at any time, the latest values of the parameters at the time a **DEFINE ZCPATH** command is processed will be used.

The parameters would normally be specified in the AVZSIN00 configuration member for a Data Virtualization Manager region.

All parameters other than the tracing parameters associated with the z/OS Connect facility are grouped into the PRODZCONNECT group.

Note: Contact Technical Support when using these parameters.

Parameter	Description	Valid values
ZCONNECT	This parameter controls whether the z/OS Connect facility will be available in this Data Virtualization Manager server. If set to NO, then nothing can be done including defining specific connections (They will be ignored with an error message). Note that the value of this parameter cannot be changed after a Data Virtualization Manager server region starts.	NO z/OS Connect support is not active. YES Activate z/OS Connect support. (default value)
ZCONNECTPWNAMEX	The default second and third parts of the name used to connect via WOLA to the z/OS Connect facility. Each part is 1-8 characters and is separated by a '.' From the next part. A specific connection definition may override this value.	NAME2.NAME3

Parameter	Description	Valid values
ZCONNECTPMINCR	The default minimum number of processing threads that are pre-attached to process incoming requests. A specific connection definition may override this value.	1-30 (default value 2)
ZCONNECTPMAXCR	The default maximum number of processing threads that are allowed. If a request comes in while the server is already processing this number of requests, an error may be returned to the requestor. A specific connection definition may override this value. See the notes below *1 and *3.	5-30 (default value 10)
ZCONNECTPMAXQR	The default maximum number of queued requests allowed if a new request arrives when the path is already processing the maximum number of concurrent requests. A specific connection definition may override this value. See the notes below *2 and *3.	0-50 (default value 10)
ZCONNECTPMAXQACT	The default action to take if the maximum number of queued requests is reached and a new request is received. Values are ERROR (Return an error to the requestor) and QUEUE (queue the request until a processing thread becomes available). A specific path definition may override this value.	ERROR
ZCONNECTPMAXQT	The default time that a queued request (as described in the previous parameter) is allowed to wait for a processing thread. If this time is exceeded, it is if ZCONNECTPMAXQACT=ERROR was in effect.	0-600 (default value 0) Note: If the value is 0, the request can wait indefinitely. A specific connection definition may override this value.
ZCONNECTPMAXPRQ	The default maximum number of requests that a given 'processing thread' will handle before it is recycled. This allows any storage leaks to be controlled.	0-1000 (default value 10) Note: If the value is 0, no recycling is performed. A specific connection definition may override this value.
ZCONNECTPMAXPTM	The default maximum time allowed to process a request. If a request cannot be completed in this time, an error is returned to the requestor. The value is in seconds. A specific connection definition may override this value.	0-600 (default value 10) Note: A value of 0 means no limit.

Parameter	Description	Valid values
ZCONNECTPMAXRECS	The default maximum number of records that can be returned on a (query) request. If the number of records is exceeded, an error is returned to the requestor. A specific connection definition may override this value.	1-10,000 (default value 1000)
ZCONNECTPMAXRLEN	The default limit on the length (in bytes) of the response that can be delivered on a request. If the size (in bytes) of the response would exceed this value, an error is returned to the requestor. A specific connection definition may override this value.	100-50,000,000 (default value 100,000)
ZCONNECTPRETRYINT	The default retry interval, in seconds. If unable to connect to a specific z/OS Connect instance, this parameter sets the number of seconds before the path is retried. A specific connection definition may override this value.	0-600 (default value 30) Note: If the value is 0, no retry is performed. If less than 10, 10 is used.
ZCONNECTPUID	If we cannot obtain a suitable userid from the WOLA or the z/OS Connect facility, this parameter can be used to nominate a userid under which the requests will run (for authorization purposes). “*” means that the region userid is used. Note: This parameter will not be needed if there is a userid.	NOUIDSET
ZCONNECTTRLEV	Sets a trace level. Customers can use this parameter (under guidance). Level meanings: 0: No tracing 1: Trace path start/stop 2: 1 + trace request begin/end 3: 2 + trace request information 4: 3 + trace i/o buffers Note: This parameter this parameter is not in the ‘PRODZCONNECT’ parameter group, but is in the tracing parameter group.	0-9

Configuring Service Provider on zcEE

After you create a z/OS Connect server, you need to edit your `server.xml` to configure support for Service Provider on zcEE.

About this task

See the [IBM documentation](#) about configuring a z/OS Connect server.

Procedure

After you create a z/OS Connect server, make the following edits to your `server.xml` to include the Service Provider on zcEE.

Note: In the various XML configuration file fragments below, colors and arrows are used to show specific connections that must be done correctly.

- Under `<featureManager>`:

```
<feature>usr:dvsProvider</feature>
<feature>zosLocalAdapters-1.0</feature>
```

Where `usr:dvsProvider` is the Service Provider on zcEE.

`zosLocalAdapters-1.0` is the WOLA provider used to communicate between the Service Provider on zcEE and the Data Virtualization Manager server.

- Be sure to include the HTTP entry where you indicate the socket ports the server will listen on:

```
<httpEndpoint id="defaultHttpEndpoint" host="*" httpPort="12345"
httpsPort="12346" />
```

- Include whatever security related entries are appropriate for your environment. For example:

```
<webAppSecurity allowFailOverToBasicAuth="true" />
<safRegistry id="saf" />
<safAuthorization id="saf2" />
<zoscconnect_zosConnectManager globalAdminGroup="ZCONUSER"
globalOperationsGroup="ZCONUSER" globalInvokeGroup="ZCONUSER" />
```

Note: These security entries must have some corresponding RACF entries on z/OS. For example, each user would need to be added to the ZCONUSER group. See the IBM z/OS Connect documentation.

At the top level add these entries for the Service Provider on zcEE:

```

<zosconnect zosConnectService
  id="zosConnectDvsService"
  serviceName="dvsService"
  serviceRef="dvsService"
  serviceDescription=" Rocket DV Service Provider"
  invokeURI="/dvs"/>
<usr dvsService
  id="dvsService"
  connectionFactoryRef="wolaCF"
  registerName="DVSR1"
  serviceName="DVSS1"
  invokeURI="/dvs"/>

```

And these entries for the WOLA service provider that DV server connects to:

```

<zosLocalAdapters
  wolaGroup="NAME1"
  wolaName2="NAME2"
  wolaName3="NAME3" />
<connectionFactory id="wolaCF" indiName="eis/ola">
  <properties.ola />
</connectionFactory>
<zosconnect zosConnectService id="sdef1" serviceName="dvs1"
serviceAsyncRequestTimeout="600s" serviceRef="svc1"/>
<zosconnect localAdaptersConnectService id="svc1" registerName="DVSR1"
serviceName="DVSS1" connectionFactoryRef="wolaCF" connectionWaitTimeout="7200" />

```

This NAME1 corresponds to the WNAME entry in the DEFINE CPATH in the server IN00 config (see previous section).

Note: The values of the wolaName2 and wolaName3 parameters can be any value, but must match the values used by the Data Virtualization Manager region. The values shown (wolaName2 = 'NAME2' and wolaName3 = 'NAME3') are the default values used by a Data Virtualization Manager region.

What to do next

See Creating RESTful services section.

Chapter 9. Distributed transactions

Two-phase commit support is a feature of IBM Data Virtualization Manager for z/OS Enterprise Transactions. IBM Data Virtualization Manager for z/OS Enterprise Transactions support various two-phase commit protocols, with support for IBM's z/OS Recovery Resource Management Services (RRMS). This support includes the use of the Resource Recovery Services attachment facility (RRSAF).

Distributed transactions use a two-phase commit protocol to synchronize related pieces of work that take place in different processes or in different data sources.

The protocol guarantees that the work is either successfully completed by all the processes or not performed at all. The goal is to ensure that each participant in a transaction takes the same action (both are committed, or both are rolled back).

After all of the work of the transaction is complete, the application attempts to commit the work by invoking the two-phase commit protocol.

A common example for the use of two-phase commit is a banking application in which a customer wants to transfer a certain amount of money from a savings account to a checking account. Either both updates must be made or neither of them should be made. The protected resource, or in this case the amount that is being transferred, must hold its integrity in case of hardware or software failures, communication failures, human errors, or a catastrophe.

Recoverable Resource Management Services (RRMS) and the two-phase commit

The IBM Data Virtualization Manager for z/OS product on z/OS implements two-phase commit through integration with IBM's z/OS Recoverable Resource Management Services (RRMS). RRMS enables transactional access to a diverse set of resources on z/OS. RRMS also enables transactional resources to access resources outside of z/OS through distributed transaction coordinators.

With the IBM Data Virtualization Manager for z/OS support of RRMS and distributed transactions, transactions originating from the Data Drivers can provide updates to:

- DB2 on the mainframe side.
- Other z/OS data sources and transactional resources, including IMS and CICS.
- Other data sources off the mainframe.

IBM Data Virtualization Manager for z/OS on the z/OS side functions as a local transaction coordinator, communicating with the client side transaction manager (typically Microsoft MTS) to allow distributed transactions using the two-phase commit protocol.

Enterprise transactions for Db2

The IBM Data Virtualization Manager for z/OS product on z/OS implements a two-phase commit through integration. This section outlines the components and requirements necessary for the integration.

Make sure that z/OS Resource Recovery Services (RRS) is configured and running. If RRS is not running, all RRSAF requests are rejected by Db2.

- **Environment Requirements**

Determine whether you are using two-phase commit with only Db2, or if you will also be using two-phase commit with IMS or CICS. Each subsystem has certain configuration requirements. Since two-phase commit is an overhead, you want to only enable what is necessary for the subsystem you are using. Everything else should be disabled.

- **The RRSAF Component**

The Resource Recovery Services attachment facility (RRSAF) is a component of RRMS that allows a z/OS address space to connect to Db2. It works with RRMS, allowing updates to a Db2 system to become part of a transaction that is managed by RRMS. RRSAF also enables two-phase commit to be used by CICS/TS and IMS/TM to coordinate updates to local data sources on the z/OS operating system and to coordinate updates between two distributed data sources, such as Oracle and Sybase.

- **The CAF Component**

Before RRSAF, call attachment facility (CAF) was the primary method that is provided by Db2 for a non-IBM address space to connect to Db2. CAF is still supported by Db2 and IBM Data Virtualization Manager for z/OS; however, it does not provide support for distributed transactions. CAF does not provide a good facility for re-using connections, and it does not provide adequate security for most customers. RRSAF provides that “request-level” facility.

Use the following guidelines to help you decide whether to use RRSAF or CAF:

- If you are using CAF and have a virtual storage shortage below the 16 MB line, try RRSAF. It uses at least 2 KB less per user than CAF, which helps reduce below-the-line virtual storage use.
- If you want to do distributed two-phase commit transactions by using IBM Data Virtualization Manager for z/OS with Db2 as one of the data sources, you must choose RRSAF as the IBM Data Virtualization Manager for z/OS attachment method.
- If only a small percentage of Db2 updates need two-phase commit transaction support, create a separate instance of the Data Virtualization Manager server for those updates, and use RRSAF with that copy.
- RRSAF can also be used to improve performance by eliminating the need to have the Data Virtualization Manager server enqueue on DSNALI OPEN-THREAD operations.
- RRSAF offers improved performance when used in conjunction with the Data Virtualization Manager server REUSETHEADS parameter. Since RRSAF allows for the use of a sign-on API, Db2 threads no longer have to be closed and reopened to correctly set the authid when the connection is obtained from the server’s pooled connections.

Configuring support for distributed Db2 transactions

This sections describes the configuration support for distributed Db2 transactions. The z/OS RRS must be properly configured to utilize a two-phase commit exclusively with Db2.

Before you begin

Make sure that RRS is enabled on the mainframe.

Make sure the following required ODBC driver connect options are set:

- XAEN
- XAOP

Procedure

1. Set up security access. The Data Virtualization Manager server address space runs as an RRS Resource Manager. Edit and submit one of the following sample jobs in *hlq*. SAVZCNTL:
 - AVZRAXA for RACF security
 - AVZA2XA for CA ACF2 security
 - AVZTSXA for CA Top Secret security
2. If you did not establish a profile for controlling access from the RRS attachment, you must edit and submit one of the following sample jobs that are located in *HLQ*. SAVZCNTL:
 - AVZRADB2 for RACF security
 - AVZA2DB2 for CA ACF2 security
 - AVZTSDB2 for CA Top Secret security

3. Add the following parameters to the AVZSIN00 configuration member:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(RRS) VALUE(YES)"
  "MODIFY PARM NAME(TRACEFULLRRSDATA) VALUE(NO)"
  "MODIFY PARM NAME(TRACERRSEVENTS) VALUE(YES)"
  "MODIFY PARM NAME(TRACERRSAF) VALUE(YES)"
  "MODIFY PARM NAME(RRS2PCALL) VALUE(YES)"
```

Parameter	Description	Valid values
RRS	Specifies whether to initialize RRS support.	YES NO Default value is NO.
RRS2PCALL	Specifies whether to use RRS COMMIT UR and RRS BACKOUT UR instead of SQL COMMIT and ROLLBACK.	YES NO Default value is NO.
TRACEFULLRRSDATA	Controls whether to trace the entire RRSAREA for RRS events.	YES The complete RRSAREA for RRS events is traced. NO (default) The complete RRSAREA for RRS events is not traced.
TRACERRSEVENTS	Specifies whether to trace RRS events.	YES Default value is YES. NO
TRACERRSAF	Specifies whether to trace each call to DSNRLI for an RRS AF request.	YES Default value is YES. NO

Configuring support for distributed Db2 transactions with the Microsoft Transaction Server

This sections describes the configuration support for distributed Db2 transactions using the Microsoft transaction server. The z/OS RRS must be properly configured to utilize a two-phase commit exclusively with Db2.

Before you begin

By default, if you use Microsoft Data Access Components (MDAC), OLE DB Session Pooling is enabled. To work with the ODBC Driver under MTS, you must turn off OLE DB Session Pooling for MSDASQL.

Make sure the required ODBC driver connect options are set.

If you change the Registry, it affects all other applications that are using the MSDASQL provider. To avoid this, you can also set this value in your application by adding the value "OLE DB Services=-4" in your connection string to turn off session pooling and autoenlistment. This setting turns off these properties for the OLE DB provider, and allows the pooling and autoenlistment to occur at the ODBC Driver level.

Procedure

1. Start the Registry Editor.

```
regedit.exe
```

2. Navigate to the following key in the registry:

```
HKEY_CLASSES_ROOT\CLSID\{c8b522cb-5cf3-11ce-ade5-00aa0044773d}\
```

3. Double-click the OLEDB_SERVICES value. The system displays the **Edit DWORD Value** dialog box.

4. In the **Value Data** field, type 0xffffffffc, and click **OK**.

Enterprise transactions for CICS/TS and IMS

System Requirements for CICS/TS

- The mainframe and each CICS region must be enabled for Resource Recovery Services (RRS).
- If you are using a VSAM file in CICS, the file must be defined with a minimum BACKOUTONLY for the RECOVERY parameter in RDO. The default for this parameter is NONE. If the parameter value is changed from NONE, you must either restart the CICS region, or deallocate and then reallocate the data set to CICS with the RECOVERY parameter set to either BACKOUTONLY or ALL.
- See information about Setting the Required Parameters within CICS in the *CICS Transaction Server for z/OS CICS System Definition Guide*.
- RRMS authorized services, is supplied in the SDFHLINK library. For information about this link list library, refer to the *CICS Transaction Server for z/OS Installation Guide*.
- See information about Setting Parameters for the z/OS Syncpoint Manager in the *CICS Transaction Server for z/OS CICS External Interfaces Guide*, which is available in the IBM Knowledge Center.

To support RRS, you must ensure that CICS and the external CICS interface both use the z/OS syncpoint manager, which is a z/OS component of Recoverable Resource Management Services (RRMS). In the context of RRMS, CICS is a resource manager. The client program can issue requests to other resource managers and have resources that are owned by those resource managers who are committed in the same unit-of-recovery (UR).

These options are controlled as follows:

- By the DPL_opts parameter of the DPL_request.
- By the SYNCONRETURN option, either specified or omitted, on the EXEC CICS LINK PROGRAM command.

If you specify SYNCONRETURN, a syncpoint is taken on completion of each DPL request. If SYNCONRETURN is omitted, a syncpoint is taken when the client program requests it using the interfaces that are described in “Taking a Syncpoint in the Client Program” in the *CICS Transaction Server for z/OS CICS External Interfaces Guide*.

Environment Requirements

You should determine whether you use two-phase commit with only DB2, or if you also use two-phase commit with IMS or CICS. Each subsystem has certain configuration requirements. Since two-phase commit is an overhead, you want to only enable what is necessary for the subsystem you are using. Everything else should be disabled.

APPC Environment Requirements for IMS

When SYNCLVL=SYNCPT is specified, Advanced Program-to-Program Communication (APPC) acquires a private context on behalf of IMS. IMS provides its resource manager name to APPC in its identity call. APPC provides the private context to IMS as the message header. IMS, using this context, then assumes the role of a participant in the two-phase commit process with the syncpoint manager, RRS/MVS.

In order to use SYNCLVL=SYNCPT, you must also be sure that an APPC/MVS logstream is defined as documented in *z/OS MVS Planning: APPC/MVS Management*. Otherwise an ATB222I error occurs when the LU is used.

In addition to SYNCLVL=SYNCPT, the keyword ATNLOSS=ALL must be specified in the VTAM definition file for whichever LUs the user wishes to enable for protected conversations.

For more information, refer to the *IMS/ESA Administration Guide: Transaction Manager*.

OTMA Environment Requirements for IMS

In an OTMA (Open Transaction Manager Access) environment, OTMA is not a resource manager who is registered with RRS/MVS. The process remains an interprocess protocol between a Server (IMS) and a number of clients (application programs).

Therefore, OTMA cannot obtain a private context token to pass to IMS as APPC does. The client-driver code that uses OTMA is responsible for obtaining and owning a private context and for providing the context ID. In messages that are passed between the partners, the context ID field contains the context token (if it is a protected conversation).

When IMS finds the context ID in the message, IMS assumes the role of a participant in the two-phase commit process, as it does in the APPC environment.

For more information about these OTMA topics, refer to the *IMS/ESA Open Transaction Manager Access Guide*.

Configuring support for CICS and IMS distributed transactions

This sections describes the configuration support for distributed CICS and IMS distributed transactions. The mainframe and each CICS region must be enabled for RRS. To support RRS, you must ensure that CICS and the external CICS interface both use the z/OS syncpoint manager.

Before you begin

Data Drivers

Make sure the following required Data Drivers connect options are configured:

- XAOP

Procedure

1. Complete the following steps to grant ALTER access to the Data Virtualization Manager server address space ID for the MVSADMIN.RRS.COMMANDS RACF resource class.

- a) Enter the following command to grant access to the resource:

```
permit MVSADMIN.** class(facility) ID(AVZS) access(alter)
```

Where AVZS is the name of the Data Virtualization Manager server address space.

- b) Enter the following command to refresh the facility class:

```
setropts refresh class(facility) raclist
```

2. Add the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(RRS) VALUE(YES)"
  "MODIFY PARM NAME(RRSCICS) VALUE(NO)"
  "MODIFY PARM NAME(RRSIMSTM) VALUE(NO)"
  "MODIFY PARM NAME(RECTABLEENTRIES) VALUE(10000)"
  "MODIFY PARM NAME(RESOURCEMGRNAME) VALUE(NEONRMAAAAA)"
```

"MODIFY PARM NAME (TRACEFULLRRSDATA) VALUE (NO)"
 "MODIFY PARM NAME (TRACERRSEVENTS) VALUE (YES)"

The following table lists the parameter for configuring Enterprise Transactions for CICS support:

Parameter	Description	Valid values
RRS	Specifies whether to enable RRS support.	YES NO Default value is YES.
RRSCICS (<i>CICS only</i>)	Specifies whether to enable RRS support for CICS.	YES NO Default value is NO.
RRSIMSTM (<i>IMS only</i>)	Allows the user to specify the RRS/two-phase interaction for IMS only.	YES NO Default value is NO.
RECTABLEENTRIES	Specifies the maximum number of entries in the RRS recovery table. If the maximum number of entries is exceeded, information about in-doubt transactions is lost.	400 (default) Valid values are 200 - 400.
RESOURCEMGRNAME	Specifies the unique sysplex name of the RRS resource manager, which is a server distributed syncpoint manager (SDSRM). If a default value is not specified, product initialization creates a 32-character name as follows: If the name is changed after the system is operational, in-doubt transactions from the previous run cannot be completed.	'NEONRRS.RESOURCE.MANAGER' <ul style="list-style-type: none"> Chars 1-24: NEONRRS.RESOURCE.MANAGER Chars 25-28: The IBM Data Virtualization Manager for z/OS subsystem name (AVZS) Chars 29-32: System SMF ID
TRACEFULLRRSDATA	Specifies whether to trace the entire RRS work area.	YES NO (default) Trace only the amount of data that fits in a standard message block.
TRACERRSEVENTS	Specifies whether to trace RRS events.	YES Default value is YES. NO

Chapter 10. Running multiple servers

There are times when you might want to bring up separate Data Virtualization Manager servers, either for testing purposes or for distributing your workload.

Configuring additional Data Virtualization Manager servers

You can bring up separate Data Virtualization Manager servers that can communicate with the peer using the Integrated DRDA Facility (IDF).

About this task

You must complete the following steps for each additional Data Virtualization Manager server you start. Sample JCL is provided in member AVZGNSUB located in the *hlq*.SAVZCNTL data set to assist with steps 1 - 5.

Procedure

1. Create a VSAM data set, for the additional Data Virtualization Manager server.
2. Make a copy of the SEF data sets for the new Data Virtualization Manager server. Copy the following data sets:

```
HLQ.ATH.SAVZEXEC  
HLQ.CMD.SAVZEXEC  
HLQ.EXC.SAVZEXEC  
HLQ.GLV.SAVZEXEC  
HLQ.PUB.SAVZEXEC  
HLQ.RPC.SAVZEXEC  
HLQ.SQL.SAVZEXEC  
HLQ.TOD.SAVZEXEC
```

3. Optional: Use the COPYMAP step that is located in the *hlq*.SAVZCNTL (AVZGNSUB) member.
Map data sets can be shared across subsystems. However it is advisable to have a separate map data set for each subsystem when separating application environments. If maps are shared across subsystems, you should make manual refreshes across the other subsystems when map changes are made. Map changes are also recognized at Data Virtualization Manager server installation.
4. Optional: Create and copy a new HTXLIB for Streams by using the COPYHTX step that is located in the *hlq*.SAVZCNTL (AVZGNSUB) member.
If you run custom RPC programs, RPC data sets are required. RPCLIB.PRELOAD is an optional RPC library that you can preload RPCs into a cache.
5. Create a new startup JCL procedure. The AVZS member of the *hlq*.SAVZCNTL library contains sample JCL procedures for running the Data Virtualization Manager server main address space (started task). You should place the AVZS PROC in a procedure library where the z/OS START command searches (this may be SYS1.PROCLIB). Optionally, you can change the name of the procedure to reflect the new Server you are starting. You must change the **SSID** parameter in the startup procedure to reflect the additional subsystem name.
6. Define the new started task to the security product.
7. Obtain or define a new TCP/IP port or VTAM application ID, depending on the communication protocol you are using.
8. Create a new AVZSIN00 member. Customize this Data Virtualization Manager configuration member to configure your interfaces and run time options.
9. Create a AVZSINEF member in the SYSEXEC concatenation where AVZS refers to the new subsystem name you are creating.

You can make a copy of your existing AVZSINEF. This member is used for initializing global variables at Data Virtualization Manager server startup. If you modified the default AVZSINEF, you should

review these changes to determine whether you want them in your new Data Virtualization Manager server. If this is not done, the REXX interpreter (SDBI or SDBX command processor) is not able to locate the specified REXX program in the data set allocated to the SYSEXEC DDNAME (for SDBI) or in the specified data set. You receive a warning message at startup of the Data Virtualization Manager server, however, you can ignore this message.

10. If the two Data Virtualization Manager servers are running at different versions set up a new REXX/EXEC to invoke the ISPF application.

Modify the LLIB statement in the REXX/EXEC to point to the Data Virtualization Manager server load library. This is not necessary if you are running at the same version but a different maintenance level. You may receive a warning regarding load library maintenance mismatches when entering the ISPF panels, however this message is informational only.

Using multiple Data Virtualization Manager servers as peers

Use the Integrated DRDA Facility (IDF) to communicate between peer Data Virtualization Manager servers.

IDF provides DRDA Application Server (AS) capability in Data Virtualization Manager, allowing communication between peer Data Virtualization Manager servers. Each Data Virtualization Manager server can use DRDA to access data sources resident at another peer Data Virtualization Manager server.

In conjunction with the DRDA Application Requestor (AR) component of Data Virtualization Manager, each Data Virtualization Manager server can operate as both a data provider and a data source.

Use IDF for peer-to-peer communications between Data Virtualization Manager servers if the servers are installed on z/OS LPARs that do not share DASD or are installed across remote z/OS locations. Using IDF, every Data Virtualization Manager server can be configured as a data source and its virtual tables made accessible to other Data Virtualization Manager servers.

To use IDF for peer-to-peer communication between your Data Virtualization Manager servers, perform the following steps:

1. Configure your Data Virtualization Manager servers. See [“Configuring multiple Data Virtualization Manager servers as peers”](#) on page 212.
2. Use the Data Virtualization Manager studio to create virtual table data maps using information provided by the remote target peer server. See [“Creating virtual maps for tables on a peer server”](#) on page 216.
3. Generate and execute requests referencing the virtual table. You can issue a query from the local server to fetch data from the remote target peer server and can include the peer table in a complex join with other local or remote tables.

Configuring multiple Data Virtualization Manager servers as peers

Configure your Data Virtualization Manager servers to use the Integrated DRDA Facility (IDF) for peer-to-peer communication.

About this task

Use the following procedure to configure multiple Data Virtualization Manager servers as peers using IDF.

The procedure uses the following terminology:

- *Local server*. The Data Virtualization Manager server that is the requester in the peer-to-peer relationship.
- *Remote target peer server*. The Data Virtualization Manager server that is the target in the peer-to-peer relationship.

DRDA is used to request and receive data from peer targets. As indicated in the procedure, you will use DEFINE DATABASE configuration commands to describe TYPE(PEER) IDF target servers.

Procedure

1. Configure the remote target Data Virtualization Manager server to activate the Integrated DRDA Facility (IDF) by setting the following start-up parameters:

```

MODIFY PARM NAME (IDF) VALUE (YES|NO)
MODIFY PARM NAME (IDFALREADYVERIFIED) VALUE (CLIENT|SERVER)
MODIFY PARM NAME (IDFLOCATION) VALUE (location)
MODIFY PARM NAME (IDFPORT) VALUE (port)
MODIFY PARM NAME (OEPORNUMBER) VALUE (port)
  
```

The following table describes these server parameters:

Parameter	Description	Valid values
IDF	IDF (Integrated DRDA Facility Activated) Determines whether IDF will be activated for the current start-up or remain inactive.	YES Activate IDF. NO IDF is not active. Default: NO
IDFPORT	IDF TCP/IP MAIN PORT TCP/IP port number on which the server listens for in-bound DRDA session requests.	Any available port number appropriate for your site. Default: 50000
IDFSSLPORT	IDF TCP/IP SSL PORT TCP/IP port number on which the server listens for in-bound DRDA SSL session requests.	Port number for SSL listener. When not set, SSL requests cannot be serviced. Default: None
IDFLOCATION	IDF LOCATION NAME DRDA location name. It is recommended that the same standard used to assign DRDA location names to Db2 subsystems be used for IDF.	A valid value is a string 1 - 16 characters. Default: ' <i>lparssid</i> ', where <i>lpar</i> is the SMFID of the LPAR and <i>ssid</i> is the server subsystem ID.
IDFALREADYVERIFIED	IDF ALREADY-VERIFIED SECURITY REQUIRED Specifies the minimum authentication level that can be used when a client connects to the IDF DRDA Application Server.	SERVER Requires a valid z/OS user ID and password. CLIENT Only a valid user ID must be supplied; a password is not required. Default: SERVER
OEPOR	Port number. This parameter is required for the Data Virtualization Manager studio to obtain metadata directly from the target.	

Note: It is recommended that the other IDF-related parameters be allowed to default unless you have specific requirements for their use.

2. On the local server, define each remote target peer server as a peer-type database server, as follows:

```
DEFINE DATABASE TYPE(PEER)
  NAME(name)
  DDFSTATUS(ENABLE)
  DOMAIN(your.domain.name)
  SECMEC(USRIDONL)
  LOCATION(location)
  PORT(port)
  OEPORT(port)
  OESSLPORT(port)
  CCSID(ccsid)
```

The following table describes these parameters:

Parameter	Description	Valid values
CCSID	The EBCDIC single-byte application CCSID (Coded Character Set Identifier). This value must match the SQLENGDFLTCCSID value specified for the target server. <i>(Required)</i>	Refer to the data source vendor documentation for a list of valid CCSID values.
DDFSTATUS	The DDF activation status. <i>(Required)</i>	ENABLE Make this DDF definition active. DISABLE DDF endpoint is not used.
DOMAIN	The domain name or hostname on which the Data Virtualization Manager server is running. Either DOMAIN or IPADDR is required, but not both.	No default value
IPADDR	The dot-notation IPV4 or IPV6 address of the host on which the Data Virtualization Manager server is running. Either DOMAIN or IPADDR is required, but not both.	
LOCATION	DRDA location name. This value must match the IDFLOCATION value specified for the target server.	A valid value is a string 1 - 16 characters.
NAME	Target server name.	A valid value consists of 1 - 4 characters.
PORT	This value must match the IDFPORT value specified for the target server.	1 - 65535
OEPORT	This value must match the OEPORT value specified for the target server.	1 - 65535

Parameter	Description	Valid values
OESSLPORT	Sets the port number that is used to LISTEN for and ACCEPT all inbound encrypted OE Sockets TCP/IP sessions. This port number should be reserved for use only by the main product address space. Each copy of the main product address space needs its own port number if SSL over OE Sockets is being used. There is no default value for the SSL port number if the value is not set in the initialization EXEC.	Null
SECMEC	The DRDA security mechanism in force. This value depends on the IDFALREADYVERIFIED value specified for the target server.	<p>USRIDPWD User ID and password are sent as is. No encryption is used. Use this setting if IDFALREADYVERIFIED is set to SERVER for the target server.</p> <p>USRIDONL User ID is sent as is. No encryption is used for the user ID only (client security). Use this setting if IDFALREADYVERIFIED is set to CLIENT for the target server.</p>
TYPE	Defines the DDF endpoint type. PEER DDF endpoint is an IDF target server.	When using IDF for peer-to-peer communication between servers, PEER is the valid value.

Example

The following example shows the settings in a peer-to-peer server configuration using IDF.

On the remote target peer server, the following start-up parameters are defined:

```

MODIFY PARM NAME(IDF) VALUE(YES)
MODIFY PARM NAME(IDFALREADYVERIFIED) VALUE(CLIENT)
MODIFY PARM NAME(IDFLOCATION) VALUE(ZOS1RDBF)
MODIFY PARM NAME(IDFPORT) VALUE(9999)
MODIFY PARM NAME(OEPORT) VALUE(9991)
MODIFY PARM NAME(OESSLPORTNUMBER) VALUE(0)
MODIFY PARM NAME(SQLENGDFLTCCSID) VALUE(1047)

```

On the local server, the remote target peer server is defined as follows:

```

DEFINE DATABASE TYPE(PEER)
NAME(RDBF)
DDFSTATUS(ENABLE)
DOMAIN(zos1.domain.name)
SECMEC(USRIDONL)
LOCATION(ZOS1RDBF)
PORT(9999)
OEPORT(9991)

```

OESSLPORT(0)
CCSID(1047)

The following table summarizes how the values correlate:

Server start-up parameter on remote target peer server	Database definition for remote target peer server on local server
IDFALREADYVERIFIED	SECMEC
IDFLOCATION	LOCATION
IDFPORT	PORT
OEPOR	OEPOR
SQLENGDFLTCCSID	CCSID

What to do next

Use the Data Virtualization Manager studio to create virtual table data maps using information provided by the remote target peer server. See [“Creating virtual maps for tables on a peer server”](#) on page 216.

Creating virtual maps for tables on a peer server

Create virtual data maps for tables defined on a remote target peer server.

Before you begin

The Data Virtualization Manager server and target peer server must be configured as peer servers. See [“Configuring multiple Data Virtualization Manager servers as peers”](#) on page 212.

About this task

Using the Integrated DRDA Facility (IDF), every Data Virtualization Manager server can be configured as a data source and its virtual tables made accessible to other Data Virtualization Manager servers. Use the Data Virtualization Manager studio to create virtual table data maps using information provided by the remote target peer server.

Procedure

1. On the **Server** tab, expand the **Discovery > Peer Subsystems > SSID** node, where *SSID* is the name of your remote target peer server.
2. Select the virtual tables, views, or both, that you want to use, and then right-click and choose **Create Virtual Table(s)**.
3. In the **New Virtual Tables Wizard**, on the **New Virtual Tables for Peer Subsystem access** page, complete the following fields:

Field	Description
Metadata Library	From the drop-down list, select the target library where the virtual table metadata will be stored (for example, hlq.USER.MAP). The target libraries are specified in the server's started task JCL.
Description	Enter an optional description.
Naming Pattern	Specify the format to use for the generated virtual table names. Use the following variables to create naming patterns that are derived from the metadata: <ul style="list-style-type: none">• {Subsystem}: Subsystem name• {Table}: Source table name

Field	Description
Virtual Target System	<p>Select a virtual target system from the drop-down list. A virtual target system points to the subsystem that contains the data that you want to access using the current virtual table. If there are no virtual target systems in the drop-down list, click Create Target System to create one.</p> <p>By using virtual target systems, you can easily change the name of the subsystem that is referenced in the virtual tables. For example, on a local server AVZ1, you create a virtual target system called TSIDF as the IDF target system, and specify that it will access the remote subsystem peer AVZ2. Then, you create 50 virtual tables that access data in the source TSIDF (that is, pointing to AVZ2). If it becomes necessary to change the name of the source AVZ2, you only have to change it in a single place by editing the virtual target system. These target systems can be located under the SQL > Target Systems > DBMS node in the server view tree.</p>

4. In the results table, review the list of selected entries. Modify the selections as needed.

Tip: Use the check box in the header row of the table to control the selection of all entries.

5. To disable MapReduce, click **Advanced** and select **Disable MapReduce**.

6. Click **Finish**.

Results

The studio creates the virtual tables (the metadata maps) on the local server. The virtual tables appear under the **SQL > Data > SSID > Virtual Tables** tree node, where *SSID* is the name of the local server.

What to do next

Generate and execute requests referencing the virtual table. You can issue a query from the local server to fetch data from the remote target peer server and can include the peer table in a complex join with other local or remote tables.

z Systems Data Compression (zEDC)

IBM z Systems Data Compression (zEDC) is an accelerated compression solution that provides high performance, low latency compression with minimal system overhead.

zEDC uses an industry standard compression library that provides efficient performance with large sequential files. zEDC facilitates cross-platform exchange of data.

Enabling zEDC

Data Virtualization Manager server provides support for IBM z Systems Data Compression (zEDC).

Before you begin

To determine the hardware and software requirements, refer to the current *IBM z Systems Data Compression* documentation.

Procedure

1. Set NETWORKBUFFERSIZE on both Data Virtualization Manager servers to a value between ZEDCMINDATASIZE and 1048512.
2. Set the following parameters in the AVZSIN00 configuration member:

```

/*-----*/
/* Enable ZEDC support.                               */
/*-----*/
if 1 = 1 then
do

```

```

"MODIFY PARM NAME(ZEDCCOMPRESSION) VALUE(YES) "
"MODIFY PARM NAME(ZEDCMINDATASIZE) VALUE(8192) "
end

if 1 = 1 then
do
"MODIFY PARM NAME(TRACEZEDCCOMPRESSION) VALUE(NO) "
"MODIFY PARM NAME(TRACEFULLZEDC) VALUE(NO) "
end

```

The following table lists the parameters for enabling zEDC:

Parameter	Description	Valid values
NETWORKBUFFERSIZE	Controls the size of the buffer used to receive blocks of data from the network. A failure will occur if a client application sends a buffer larger than the maximum size. This value should be raised to allow larger blocks of data to be sent to and from the client.	256K (default) or required size.
TRACEZEDCCOMPRESSION	Enables tracing of all zEDC calls to the Server Trace facility. It should only be set to YES if the user needs to trace zEDC calls for diagnostic purposes.	YES Enable zEDC tracing. NO (default) Do not enable zEDC tracing.
TRACEFULLZEDC	Traces the entire buffer, not just the first few bytes. It should only be set to YES if a minimal trace is not enough.	YES Enable zEDC tracing for the entire buffer. NO (default) Do not enable full zEDC tracing.
ZEDCCOMPRESSION	Enables or disables the use of the zEDC compression hardware device. Set to YES if you have the zEDC compression hardware and wish to use it.	YES Enable zEDC compression. NO (default) Do not enable zEDC compression.
ZEDCMINDATASIZE	Sets the minimum amount of data the server will compress with the zEDC hardware. It is recommended that testing first be done with a minimum size of 8K.	8192 (default) or required size.

- To verify that zEDC is in use, enable zEDC tracing (TRACEZEDCCOMPRESSION) and look for ZED events in the Server Trace.

Chapter 11. IDF for Db2 mainframe applications

You can use the Integrated DRDA Facility (IDF) for Db2 mainframe applications so that z/OS Db2 systems can communicate with Data Virtualization Manager server to access Z and distributed data sources using Distributed Relational Database Architecture (DRDA).

Support of this feature is based upon the use of three-part table names within Db2 stored procedures and applications, Db2 RESTful Services, QMF for TSO, and the SQL processor using file input (SPUFI) facility within the Db2 Interactive (Db2I) primary option menu of ISPF. Table names in SQL statements are composed of three parts: the location name, the schema or owner name, and the table name; the parts being delimited by periods. For example, *LOCATION.SCHEMA.TABLE* is a fully-qualified three-part table name. Often, the first, or first two parts of a table name are omitted with default values assumed for each.

When the first part of a three-part table name is omitted, the current Db2 Server is the assumed location where the table resides. Using fully qualified three-part table names, the location name part may designate the same or some other server where the table resides. When a table is remote, DRDA is used to communicate between Db2 and the remote server. The SQL is sent to the remote server for execution and the results returned to the requestor.

Note: Db2 supports definition of a table alias that maps to a three-part table name, thereby avoiding explicit, hard-coded location designations within applications and stored procedures. These aliases can later readily be re-assigned to change from one location to another or from a test to a production environment.

IDF allows a Data Virtualization Manager server to participate as an endpoint in multi-homed configurations. Whenever SQL is executed in Db2 and a three-part table name designates an IDF endpoint, the SQL is sent to the Data Virtualization server for execution. There, the table name designates a virtual table which may be backed by VSAM, ADABAS, IMS, or any of the other supported backend sources Data Virtualization Manager supports. Results of the execution are returned to Db2, which proceeds as it would for any local table access.

Note: IDF-Direct does not support Stored Procedure.

- SQL engine now supports DRDA pass through by default (value is set to NO). To explicitly set it, use the parameter `SQL DISABLE DRDA PASSTHROUGH (SQLENGNODRDAPT)`. In the server IN00 file, it can be set as follows: (Note that the default is NO. NO keeps the DRDA pass through enabled.)

```
"MODIFY PARM NAME(SQLENGNODRDAPT) VALUE(NO) "
```

- To disable the DRDA pass through, set the value to YES as follows:

```
"MODIFY PARM NAME(SQLENGNODRDAPT) VALUE(YES)"
```

Known Restrictions and Limitations

This section describes the known restrictions and limitations for IDF in DB2.

IDF for DB2 mainframe applications supports SELECT (including SELECT INTO), INSERT, UPDATE, and DELETE access to data sources using either bound-statements or dynamic SQL.

Note: The operations supported vary by data source. Not all data sources allow for all update-type operations. For example, you cannot alter data within a logstream in any way, nor you cannot delete rows from a sequential dataset.

Stored procedures and DB2 application programs may execute statements bound into a package or SQL statements can be prepared dynamically. DB2 applications can use the DCLGEN utility to extract meta-data information from the Data Virtualization Manager server. DB2 Bind is used to propagate packages and SQL statements to target Data Virtualization Manager servers. The facility does not yet support scrollable cursors, cursor-positioned updates, nor two-phase commit.

Following are the known restrictions and limitations of IDF:

• **Restrictions:**

- IDF supports INSERT, DELETE, UPDATE, SELECT, and SELECT INTO statements. For pathways using a bound package, the facility supports the use of DECLARE CURSOR FOR SELECT statements, along with OPEN, FETCH, and CLOSE for the cursor.
- The operations allowed may differ for each data source. For example, Data Virtualization Manager server virtual tables backed by logstreams cannot be updated.
- COMMIT and ROLLBACK operations may not work as expected due to the limitations of the back-end virtual table data source. For example, ROLLBACK may not work as expected when using VSAM datasets unless the RLS recovery attributes of the dataset meet the requirements needed for backout processing.
- You should ensure that the IDF SQL statements conform to ANSI-92 standards. That is, you should specify SQL92 for language level in the DB2 preprocessor. The SQL engine only supports apostrophes and period. You may achieve this by using the pre-compiler parm SQLFLAG(SQL92E).

• **Limitations:**

- SQLCODE and SQLSTATE values returned by execution of SQL statements over IDF may differ from codes returned by DB2. In many cases, the facility will return the non-specific SQLCODE -1 with more specific error text returned in the SQLERRMC fields.
- When DB2 applications run, and if DB2 requests IDF not to update any resources, IDF deliberately ignores the do-not-update request and instead allows the updates to occur and engages auto-commit processing after each operation.
- Versioning for DB2 stored procedures and for DB2 RESTful service definitions is not supported.
- Cursor positioned UPDATE and DELETE, that is “WHERE CURRENT OF CURSOR” form, are not supported.
- Multi-row INSERT, that is "FOR value ROWS" clause, is not supported.
- Two-phase commit is not supported.
- IDF for DB2 Applications currently does not support LOB columns.
- IDF for DB2 Applications currently does not support stored procedure CALL.

Configuring Data Virtualization Manager server for IDF

Data Virtualization Manager server requires some DRDA Application Server setup parameters for operation. Perform the following step to configure a Data Virtualization Manager server for IDF:

Procedure

To activate the DRDA-AS (Application Server) to handle IDF requests in a Data Virtualization Manager server, set the following startup parameters in the IN00 dataset:

Parameter Name	Description	Default value	Value to Set
IDF	Determines whether IDF/IDF will be activated for the current start-up or remain inactive.	NO IDF is not active.	Must be set to YES to activate IDF
IDFPORT	TCP/IP port number on which the server listens for in-bound DRDA session requests.	50000	Choose any available port number appropriate for your site.

Parameter Name	Description	Default value	Value to Set
IDFSSLPORT	TCP/IP port number on which the server listens for in-bound DRDA SSL session requests.	None	Choose a port number for SSL listener. When unset, SSL requests cannot be serviced.
IDFLOCATION	DRDA location name.	The LPAR's SMFID suffixed with server's subsystem ID to create an 8-byte location name	We recommend that same standard used to assign DRDA location names to DB2 subsystems be used for IDF.
IDFALREADYVERIFIED	Specifies the minimum authentication level that can be used when any DRDA client connects to the IDF/VDF DRDA Application Server.	SERVER	SERVER - require a valid z/OS User ID and password. CLIENT - only a valid User ID must be supplied; a password is not required.
PASSTICKETAPPNAME	Specifies the APPLNAME which the server uses to validate RACF passtickets.	The server's three-byte product ID suffixed with the LPAR's SMFID	If RACF passtickets are to be sent by DB2 when connecting to IDF, this value MUST match the LINKNAME value used to populate the corresponding DB2 CDB communications database tables.
It is recommended that the other IDF-related parameters be set with their default values unless you have specific requirements for their use.			

Populating DB2 communications database for IDF

Before DB2 can connect to a remote IBM Data Virtualization Manager for z/OS server, DB2 must know that a location name (from the three-part table name in SQL) is valid and it must know how to establish a TCP/IP communication session between itself and the peer location. DB2 refers to a set of catalog tables, collectively known as communication database (CDB), to resolve location names to the information needed to establish DRDA sessions.

It is recommended that you review the [Populating the communications database for use with TCP/IP](#) topic for more information on how to populate CDB.

Each remote location is defined to DB2 by inserting or updating rows in the following catalog tables:

- SYSIBM.LOCATIONS
- SYSIBM.IPNAMES
- SYSIBM.USERNAMES

Populating the SYSIBM.LOCATIONS table

Each IBM Data Virtualization Manager for z/OS server must have a unique IDFLOCATION name assigned to it. The assigned location name is the high-order qualifier used in three-part table names. At minimum, one row must be added to SYSIBM.LOCATIONS to define the location to DB2. The table LOCATION

column is a primary key and must match the unique IBM Data Virtualization Manager for z/OS server target IDFLOCATION. The row's LINKNAME column is used to de-reference connection information in the other tables. A LINKNAME value may be uniquely defined for the specific location's connection information, or it may designate a set of definitions used in common for accessing many remote servers. Perform the following to populate the SYSIBM.LOCATIONS table:

Procedure

Set the columns of the SYSIBM.LOCATIONS catalog table to:

Column	Value
LOCATION	Set to the unique IDFLOCATION name assigned to the target DV server. For IDF-Direct, this location can be either an IDF prefixed pseudo-location or a 1-to-16-byte location name (which matches the location name of the DRDA data source). The IDF prefixed pseudo-location is in the format IDF_XXXX where XXXX is the subsystem name provided for the data source under DEFINE DATABASE in IN00.
LINKNAME	An 8-byte link name value. This name is used to look up matching entries in the other SYSIBM tables.
PORT	Set to the value of IDFPORT or IDFSSLPORT.
SECURE	Set to "Y" if SSL is to be used (PORT should be set to IDFSSLPORT number)

Following is a sample SQL statement used to create a row in the locations table:

```
INSERT INTO SYSIBM.LOCATIONS (LOCATION, LINKNAME, PORT, SECURE)
VALUES('LQ52ABCD', 'LQ52ABCD', '7000', 'N')
```

Populating the SYSIBM.IPNAMES table

A row in the SYSIBM.IPNAMES table contains to set up information for the remote server's IP address and security processing used when connecting. The row in SYSIBM.IPNAMES is selected by matching to the LINKNAME from the SYSIBM.LOCATIONS row for the target server. Perform the following to populate the SYSIBM.IPNAMES table:

Procedure

Set the columns of the SYSIBM.IPNAMES catalog table to:

Column	Value
LINKNAME	An 8-byte link name value. This name is used to matched to the LINKNAME in the SYSIBM.LOCATIONS table.

Column	Value
SECURITY_OUT	IDF for DB2 applications supports the use of options 'A', 'P', and 'R'. For more information on these options, For more information on IPNAMES catalog table, refer the IPNAMES topic. Note: IDF does not support DRDA-level encryption of data <i>payload</i> objects (options D and E are not supported). SSL sessions should be used when data <i>payload</i> encryption is needed. IDF support the use of encrypted credentials. When you select option A, P, or R, DB2 will initially request to use clear-text credentials. IDF can be configured to require that user ID, or user ID and password be AES encrypted, and will negotiate this added level of protection with DB2 before the values are transmitted.
USERNAMES	Set 'O' to use SYSIBM.USERNAMES translation or blank to pass user IDs unchanged
IPADDR	Set to the IPv4 or IPv6 IP address or the character format domain name of the remote location.

Following is a sample SQL statement used to create a row in the IP names table:

```
INSERT INTO SYSIBM.IPNAMES (LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
VALUES('LQ52ABCD', 'A', 'O', 'LQ52')
```

Populating the SYSIBM.USERNAMES table

A row in the SYSIBM.USERNAMES table contains to set up information used to translate/output user IDs send when connecting to the remote server. The row in SYSIBM.USERNAMES is selected by matching to the AUTHID and LINKNAME columns. Perform the following to populate the SYSIBM.USERNAMES table:

Procedure

Set the columns of the SYSIBM.USERNAMES catalog table to:

Column	Value
LINKNAME	An eight-byte link name value. This name is used to matched to the LINKNAME in the SYSIBM.LOCATIONS table.
TYPE	Specify 'O' for outbound security translation
AUTHID	Set to a specific user ID or to blanks to match to all user IDs.
NEWAUTHID	New authorization ID to be output in place of AUTHID from lookup.
PASSWORD	Password to accompany translated authorization ID.

Following is a sample SQL statement used to create a row in the IP names table:

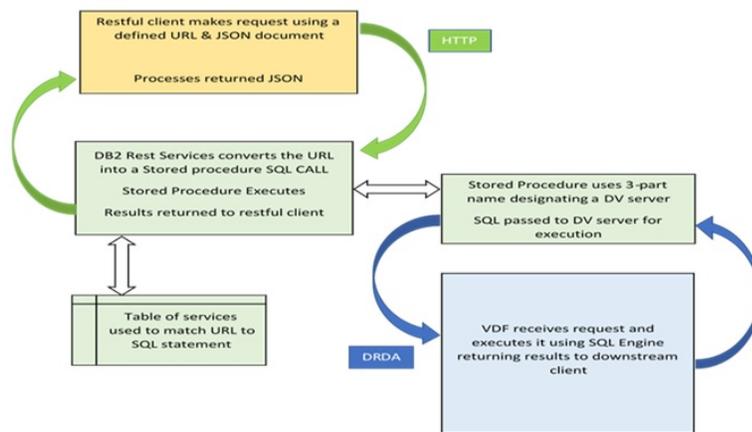
```
INSERT INTO SYSIBM.USERNAMES (LINKNAME, TYPE, AUTHID)
VALUES('LQ52ABCD', 'O', 'APJJK')
```

Using IDF for Mainframe Applications to implement DB2 RESTful services

IDF can be used for Mainframe Applications to implement DB2 RESTful services. When creating DB2 RESTful services, IDF can be used to incorporate data virtualization virtualized tables and views.

Using Data Virtualization Manager server, you can create UDTFs which are inherently read-only. Using IDF for mainframe applications, you can create a DB2 Stored Procedure that uses three-part table names to both read and update non-DB2 data sources. This stored procedure can then be incorporated into a DB2 REST Service

The following diagram illustrates the access flow when IDF three-part table name is used in a DB2 stored procedure to implement a DB2 RESTful service.



DB2 RESTful services provisioning

IDF for DB2 mainframe applications is used to create RESTful web services. As a REST service provider, Db2 enables your web, mobile, and cloud applications to interact with Db2 data through a set of scalable RESTful APIs.

One use case of IDF for DB2 mainframe applications is in the creation of RESTful web services using three-part table names within DB2 stored procedures. If this is an intended use case, refer [DB2 REST services](#) for information about provisioning DB2 REST services.

Note: IDF for mainframe applications does not support versioning for DB2 REST services, nor versioning for DB2 stored procedure packages.

Creating a DB2 RESTful service using IDF

When creating a DB2 REST service, you cannot use three-part table names directly in the SQL text bound with the DB2 REST service definitions. Three-part table name references are not supported for use in the SQL text for these services. However, this restriction applies only to the *bound-within-the-service* SQL statement text. A DB2 REST service may invoke a local stored procedure using an SQL CALL statement, and that local stored procedure *may* contain embedded three-part table name references.

About this task

The steps to define a DB2 REST service using IDF for mainframe applications are:

1. Create a DB2 stored procedure using three-part table name references to Data Virtualization Manager server-based tables.
2. Bind or copy the stored procedure package into the Data Virtualization Manager server.
3. Define the RESTful service as an SQL CALL invoking the procedure.

Example

The following sample JCL illustrates one means of creating a DB2 stored procedure using batch JCL. You can also use the SQL processor using file input (SPUFI) facility within the DB2 Interactive (DB2I) primary option menu of ISPF or run the CREATE procedure from a remote studio application. The procedure parameters are used to insert a row into a Data Virtualization Manager server managed VSAM dataset.

```
//JOB CARD JOB ,SAMPLE,CLASS=A,MSGCLASS=X,REGION=4M,
// NOTIFY=&SYSUID
//* Create stored procedure that inserts row into IDF table at LQ52ABCD server
//BIND1 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB9D.SDSNEXIT
// DD DISP=SHR,DSN=DSN.VB10.SDSNLOAD
// DD DISP=SHR,DSN=DSN.VB10.RUNLIB.LOAD
//DBRMLIB DD DISP=SHR,DSN=DSN.VB10.DBRMLIB.DATA
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB9D)
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2)
END
/*
//SYSIN DD *
--#SET TERMINATOR &
DROP PROCEDURE SYSPROC.PUTORG&
COMMIT&
CREATE PROCEDURE SYSPROC.PUTORG(IN PDEPT INT,
                                IN PMGR INT,
                                IN PNAME CHAR(20),
                                IN PDIV CHAR(10),
                                IN PLOC CHAR(20))

LANGUAGE SQL
BEGIN
INSERT INTO LQ52ABCD.VDFDEMO.ORGSEQ
(DEPTNUMB, MANAGER, DEPTNAME, DIVISION, LOCATION)
VALUES(PDEPT, PMGR, PNAME, PDIV, PLOC);
COMMIT;
END&
GRANT EXECUTE ON PROCEDURE SYSPROC.PUTORG TO PUBLIC&
COMMIT&
/*
```

When a stored procedure is created, DB2 internally generates a package containing the SQL statements from within the procedure. The stored procedure package must be copied from DB2 into the target IDF Data Virtualization Manager server. You can use SPUFI bind-package dialogs to perform the bind-copy operation, or utilize a batch job as illustrated in the following sample:

```
//JOB CARD JOB ,SAMPLE,CLASS=A,MSGCLASS=X,REGION=4M,
// NOTIFY=&SYSUID
//* Copy SP package to IDF server LQ52ABCD
//BIND2 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB9D.SDSNEXIT
// DD DISP=SHR,DSN=DSN.VB10.SDSNLOAD
// DD DISP=SHR,DSN=DSN.VB10.RUNLIB.LOAD
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(DB9D)
BIND
ACTION(REPLACE) -
CURRENTDATA(NO) -
COPY(SYSPROC.PUTORG) -
COPYVER(V1) -
PACKAGE(LQ52ABCD.SYSPROC) -
DEFER(PREPARE) -
KEEPDYNAMIC(YES) -
EXPLAIN(NO) -
FLAG(I) -
DEGREE(1) -
ISOLATION(CS) -
RELEASE(COMMIT) -
SQLERROR(CONTINUE) -
REOPT(ONCE) -
VALIDATE(BIND) -
DYNAMICRULES(RUN) -
```

```

ENCODING(EBCDIC)      -
APREUSE(NONE)        -
APCOMPARE(NONE)
END
/*

```

Note:

- The COPYVER version must be V1.
- IDF does not support stored procedure package versioning.

Finally, you must define the RESTful service URL and relate it to an SQL statement that is executed when the service is requested. The following sample shows this definition being performed using batch JCL. You may also use a DB2-provided RESTful service to create the definition.

```

//JOB CARD JOB ,SAMPLE,CLASS=A,MSGCLASS=X,REGION=4M,
// NOTIFY=&SYSUID
/* Define the REST service to DB2
//BIND3 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB DD DISP=SHR,DSN=DB9D.SDSNEXIT
// DD DISP=SHR,DSN=DSN.VB10.SDSNLOAD
// DD DISP=SHR,DSN=DSN.VB10.RUNLIB.LOAD
//SQLDDNAM DD *
      call sysproc.Putorg(:pdept, :pmgr, :pname, :pdiv, :ploc)
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
      DSN SYSTEM(DB9D)
      BIND SERVICE(VDFDEMO) NAME("Putorg") -
      SQLDDNAME(SQLDDNAM) -
      DESCRIPTION('Add a row to ORGSEQ table')
END
/*

```

Note that the SQL statement is a CALL for the storage procedure with the parameters defined when the stored procedure was created.

Once this definition is made, you can execute the RESTful service by posting a JavaScript Object Notation (JSON) document containing values for each of the parameters to the /service/VDFDEMO/PutorgURL.

Accessing IDF from DB2I (SPUFI)

Using the SQL processor using file input (SPUFI) facility to connect to DB2, you can issue SQL queries that uses 3-part names to access IBM Data Virtualization Manager for z/OS via Integrated DRDA Facility (IDF).

Example

The following selects 5 rows from a VSAM dataset mapped as STAFFVS by the IDF server:

```

SELECT * FROM LQ52ABCD.VDFDEMO.STAFFVS FETCH FIRST 5 ROWS ONLY;
-----+-----+-----+-----+-----+-----+
      ID  NAME_L  NAME_D      DEPT  JOB      YRS
-----+-----+-----+-----+-----+
      10     7  SANDERS      20  MGR      7
      20     6  PERNAL      20  SALES    8
      30     8  MARENGHI    38  MGR      5
      40     7  O'BRIEN    38  SALES    6
      50     5  HANES      15  MGR     10
DSNE610I NUMBER OF ROWS DISPLAYED IS 5
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
DSNE621I NUMBER OF INPUT RECORDS READ IS 32

```



```

WORKING-STORAGE SECTION.
*****
* DCLGEN TABLE(LQ52ABCD.VDFDEMO.ORGSEQ) *
* LIBRARY(CSD.AI38.W5.COBOL(ORGSEQ)) *
* ACTION(REPLACE) *
* LANGUAGE(COBOL) *
* STRUCTURE(ORGTAB) *
* QUOTE *
* ... IS THE DCLGEN COMMAND THAT MADE THE FOLLOWING STATEMENTS *
*****
EXEC SQL DECLARE LQ52ABCD.VDFDEMO.ORGSEQ TABLE
( DEPTNUMB          SMALLINT NOT NULL,
  MANAGER           SMALLINT NOT NULL,
  DEPTNAME          CHAR(20) NOT NULL,
  DIVISION          CHAR(10) NOT NULL,
  LOCATION          CHAR(20) NOT NULL
) END-EXEC.
*****
* COBOL DECLARATION FOR TABLE LQ52ABCD.VDFDEMO.ORGSEQ *
*****
01 ORGTAB.
10 DEPTNUMB          PIC S9(4) USAGE COMP.
10 MANAGER           PIC S9(4) USAGE COMP.
10 DEPTNAME          PIC X(20).
10 ODIVISION         PIC X(10).
10 LOCATION          PIC X(20).
01 H1-DATE.
05 H1-YYYY          PIC 9(4).
05 H1-MM            PIC 9(2).
05 H1-DD            PIC 9(2).
01 H1-TIME.
05 H1-HH            PIC 9(2).
05 H1-MN            PIC 9(2).
05 H1-SS            PIC 9(2).
05 H1-TT            PIC 9(2).
PROCEDURE DIVISION.
ACCEPT H1-DATE FROM DATE YYYYMMDD.
ACCEPT H1-TIME FROM TIME.
COMPUTE DEPTNUMB     = H1-TT + H1-MN + H1-HH.
COMPUTE MANAGER      = H1-YYYY + H1-MM + H1-DD.
MOVE 'GENERATED ENTRY' TO DEPTNAME.
MOVE H1-TIME          TO ODIVISION.
MOVE 'PGM=SAMPORG'   TO LOCATION.
EXEC SQL
  INSERT INTO LQ52ABCD.VDFDEMO.ORGSEQ VALUES(
    :DEPTNUMB, :MANAGER, :DEPTNAME, :ODIVISION, :LOCATION)
END-EXEC.
GOBACK.

```

Chapter 12. Security

Advanced security is available for SQL, NoSQL, Events, and Services solutions. System programmers typically configure advanced security during Data Virtualization Manager server installation.

IBM Data Virtualization Manager for z/OS provides the following security features:

- Security Optimization Management (SOM) is a unique feature within the mainframe integration suite that manages and optimizes mainframe security authentication for any process that requires authentication, such as a web service or SQL call.
- Secure Sockets Layer (SSL) for the Data Virtualization Manager server is transparently supported by the Application Transparent Transport Layer Security (AT-TLS), an IBM TCP/IP facility.
- Enterprise auditing supports the new and unique security requirements of Internet applications, while operating in the traditional enterprise computing environment. With enterprise auditing, web applications that access IBM z/OS operating system data and transactions can be used by people who do not have mainframe user IDs.
- Data Virtualization Manager server provides protection for its resources using RACF classes, CA Top Secret classes, and CA ACF2 generalized resource rules. You can run multiple instances of Data Virtualization Manager servers and either share the authorization rules or keep them separate.

Security Optimization Management (SOM)

Security Optimization Management (SOM) caches user authorization information for logon processing. SOM reduces the overhead costs that are associated with sign-on processing for all connections to Data Virtualization Manager server.

To accomplish this task, SOM saves an ACEE (accessor environment element) in a cache when the user successfully logs on to the Data Virtualization Manager server, allowing the ACEE to be reused the next time that the user logs on to the Data Virtualization Manager server. Caching ACEEs provides improved system performance by reducing logon times and by reducing overall input/output and processor consumption through the reduction of security database accesses.

Performance gains vary based on the number of logons performed. A user can invoke a client transaction (application transaction). Users running client transactions that use a unique user ID and perform logons for each transaction, see the most performance benefit. The security database stores information about the RACF, CA ACF2, or CA Top Secret configuration for users. Security database profile records are not updated and SMF records are not written when a cached ACEE is used to satisfy a logon request.

The Data Virtualization Manager server provides both basic and advanced SOM support. With advanced SOM support, the Data Virtualization Manager server can automatically expire cached ACEEs when security changes are made to the security database for a particular user ID.

Enabling basic SOM support

Basic SOM support is available for RACF, CA ACF2, and CA Top Secret.

Procedure

You can configure basic SOM support by using the `MODIFY PARM` command to set the following parameters that are located in the `AVZSIN00` configuration member:

```
if DoThis then
do
  "MODIFY PARM NAME(SEcurityOPTIMIZATION) VALUE(YES)"
  "MODIFY PARM NAME(SECOPTRETAIN) VALUE(28800)"
  "MODIFY PARM NAME(SECOPTTARGET) VALUE(5000)"
  "MODIFY PARM NAME(SECOPTTHRESHINT) VALUE(1200)"
  "MODIFY PARM NAME(SECOPTTHRESHOLD) VALUE(10)"
  "MODIFY PARM NAME(TRACESECOPTINT) VALUE(NO)"
```

"MODIFY PARM NAME(TRACESECOPTOPS) VALUE(NO)"
 "MODIFY PARM NAME(TRACESECOPTSUM) VALUE(NO)"

The following table lists the parameters for configuring basic SOM support:

Parameter	Description	Valid values
SECURITYOPTIMIZATION	Controls whether to cache the security environments for successful remote user logons.	YES (default) Caches the security environments. NO
SECOPTRETAIN	Specifies the amount of time, in seconds, that a cached security environment remains valid.	28800 (default value) Minimum Value: 0 Maximum Value: 86400 (24 hours)
SECOPTTARGET	Specifies the target number of security environments to keep in the user security cache. Note: This target number increases if there are not enough available cache entries to maintain an entry for all currently logged-on users.	5000 (default value) Minimum Value: 500 Maximum Value: 100000
SECOPTTHRESHINT	Amount of time, in seconds, that the security cache is scanned to find entries to delete. Valid values are integers that divide evenly into 3600 (one hour). For example, 60 is a valid value, but 33 is not because it does not divide evenly into 3600.	1200 (default value) Minimum Value: 60 Maximum Value: 43200 (12 hours)
SECOPTTHRESHOLD	Specifies the percentage of cache entries that are available after threshold interval process runs.	10 (default value) Minimum Value: 5 (percent) Maximum Value: 50 (percent)
TRACESECOPTINT	Controls tracing of intervals.	YES NO (default) Do not trace intervals.
TRACESECOPTOPS	Controls tracing of operations.	YES NO (default) Do not trace operations.
TRACESECOPTSUM	Controls tracing of summary and statistical information.	YES NO (default) Do not trace summary and statistical information.

Enabling advanced SOM support

Advanced SOM support detects user ID and password changes that occur in the security database while the user ID security information is cached in Data Virtualization Manager server. When this occurs, the cached security information is expired within the Data Virtualization Manager server. The existing connection, if one exists, is allowed to continue, however a new connection request requires that the user ID be re-authenticated by the security product.

About this task

Advanced SOM support consists of two RACF exit routines: the RACF common command exit, IRREVSX01, and the RACF new password exit, ICHPWX01.

The IRREVSX01 exit is automatically installed each time that the server starts. This exit routine notifies SOM whenever an **ALTUSER**, **CONNECT**, **DELUSER**, or **PASSWORD** command runs.

The ICHPWX01 exit is installed into LPALIB and requires an IPL to enable it. Existing ICHPWX01 exit routines can be included with the Data Virtualization Manager server version of ICHPWX01. Running without this exit means that SOM does not detect password changes that are made during logon by an application other than the Data Virtualization Manager server.

The following example shows how password processing is performed before the password exit routine is installed.

- User A logs on to the Data Virtualization Manager server. User A then logs on to TSO and specifies a new password during the logon. SOM does not detect this change.
- The Data Virtualization Manager server detects the new password only when user A uses it to log on to the Data Virtualization Manager server again.

Two cases would be encountered when the exit is not installed and the user has changed the password by logging on through another application, such as TSO:

- User A logs off the Data Virtualization Manager server and logs back on using the new password. SOM detects that the password changed, expires the existing cache entry, and then calls RACROUTE using the new password for that logon.
- User A logs off the Data Virtualization Manager server and logs back on using the old password. SOM allows the logon, if user A's cache entry has not expired, even though the password was previously changed when logging on to TSO.

You can uninstall the exits at any time using the SOM ISPF application. The RACF common command exit, IRREVSX01, is reinstalled the next time that you start another instance of the Data Virtualization Manager server that has the Advanced SOM Support feature enabled.

To enable Advanced SOM Support for RACF:

Procedure

1. Use the `MODIFY PARM` command to set the following parameter that is located in the Data Virtualization Manager configuration member, AVZSIN00:

```
if DoThis then
  do
    "MODIFY PARM NAME(SECOPTADVANCED) VALUE(YES)"
```

The following table lists the parameters for configuring advanced security optimization:

Parameter	Description	Valid values
SECOPTADVANCED	<p>Specifies whether SOM uses the advanced user profile change detection exits. When set to NO, SOM checks for the existence of the exits, but does not install them. When set to YES, SOM installs the dynamic security exits, if not already installed, and checks for the existence of the static security exits. This parameter must be configured before starting the server.</p> <p>If an instance of the Data Virtualization Manager server has the value set to YES, all servers on that LPAR are notified of the user profile changes. If a newer version is available and this parameter is set to YES. SOM replaces the exit.</p> <p>Advanced security features are only available for the RACF security server.</p>	<p>YES</p> <p>NO</p> <p>Default value is NO.</p>

2. Install RACF New password exit, ICHPWX01, using the job in member LINKPWDX in the h1q.SAVZCNTL data set. Follow the instruction in the job to install the exit routine.

Using Passtickets

There is a special consideration for SOM when using Passtickets. It is inefficient to cache ACEEs for users that are authenticated using Passtickets regardless of the Replay Protection parameter setting.

About this task

You can tell SOM not to cache an ACEE for logons using Passtickets by setting the **ATH.AUPWUSPT** parameter to 1 in a SEF ATH LOGON rule.

Logon and logoff processing

When a client logs on to the server, SOM searches the ACEE cache for a match using the user ID and password that is supplied by the remote client. If a matching entry is found, the ACEE associated with that cache entry is used for that transaction and the counter for that cache entry is increased by one.

If an available matching entry is not found, a RACROUTE call verifies the user ID and password. After the user ID and password are verified, an entry is added to the cache for the new ACEE, and the counter for the matching cache entry is set to one.

When a client logs off, the counter for the entry is decreased by one.

ACEE retention and deletion

An ACEE (accessor environment element) is retained if a remote client task is using it, and the retention period is not expired.

The retention period is set after RACROUTE logon processing. The retention period is set to the value of the **SECOPTRETAIN** parameter, and can be overridden by the *ATH.AUPWAERT* variable set in a SEF ATH LOGON RULE.

ACEEs are deleted during interval processing or during cache steal operations. An ACEE is eligible for deletion when the responsibility count is zero, and the retention period is expired or the entry was marked expired.

An entry can be set to expired status by:

- The **KILL CACHE USER** command.
- The **EXPUSER SEF CMD** rule, which can be entered from an MVS console.
- Setting the **EXPIRESECOPTENTRIES** parameter to YES.
- Issuing a RACF **ALTUSER, CONNECT, DELUSER, PASSWORD, or REMOVE** command.
- Changing the password during logon processing.

Configuring mixed case password for RACF password

Data Virtualization Manager allows to configure using mixed case characters in the passwords.

About this task

By default, the server, Data Virtualization Manager Studio, and the client drivers convert all the passwords entered to uppercase when sending authentication requests to z/OS. When RACF mixed case password support is enabled, the password sent for authentication to z/OS can result in an invalid password.

If the client application requires mixed case passwords, then both the client and the server will require additional configuration, without the additional configuration the login requests with mixed-case passwords will fail.

Note: Uppercase RACF passwords will work regardless of any component settings.

Before you begin

The client driver allows the **UCLC** parameter to be passed to its Client connection string.

Set **UCLC=NO** to leave the password unmodified as it is passed to the Server. The studio will provide the **UCLC** parameter in its client connection string while authenticating the request.

- Select **Preferences > Data Services** and then clear **Use UPPER case logon credentials for both JDBC and HTTP connections** check box to add UCLC=NO in the connection string.

Procedure

Use MODIFY PARM command to set the following parameter located in the AZKnIN00 configuration member to configure the Servers for RACF mixed case passwords.

```
MODIFY PARM NAME(PASSWORDCASE)VALUE(ASIS)
```

The following table lists the parameter for configuring RACF mixed case password.

Parameter name	Parameter description	Value
PASSWORDCASE	<p>The default for PASSWORDCASE is UPPER which causes all passwords to be in uppercase on the server before passing to RACF.</p> <p>If you do not want the password in the server in uppercase, then the PASSWORDCASE parameter should be set to ASIS.</p>	

Secure Sockets Layer (SSL)

Secure Socket Layers (SSL) is supported by the Application Transparent Transport Layer Security (AT-TLS), an IBM TCP/IP facility.

Data Virtualization Manager supports connections in the following ways:

- Ports that recognize an SSL connection and automatically enable an SSL session.
- Ports that are for secure connections that always send encrypted data.

Enabling SSL support

The Data Virtualization Manager server allow you to configure the SSL connection to send encrypted data.

Before you begin

Your user ID must have READ permission for the IRR.DIGTCERT.LISTRING and IRR.DIGTCERT.LIST profiles in the RACF FACILITY class. If SSLUSERID is not specified, the Data Virtualization Manager server address space default user ID is used.

Procedure

1. Use the **MODIFY PARM** command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
"MODIFY PARM NAME(SSL) VALUE(YES)"
"MODIFY PARM NAME(SSLAUTODETECT) VALUE(NO)"
"MODIFY PARM NAME(SSLCLIENTAUTH) VALUE(LOCAL)"
"MODIFY PARM NAME(SSLCLIENTNOCERT) VALUE(ALLOW)"
"MODIFY PARM NAME(SSLUSERID) VALUE(USERID)"
```

Parameter	Description	Valid values
SSL	Enables SSL connections.	YES (default) SSL connections enabled. NO

Parameter	Description	Valid values
SSLAUTODETECT <i>(Optional)</i>	<p>Specifies whether the server automatically detects SSL connections that are sent on the port that is normally used for cleartext connections.</p> <p>Note: A separately configured SSL port accepts only SSL connections.</p>	<p>YES When set to YES, the server automatically detects SSL connections.</p> <p>NO (default) When set to NO, only cleartext connections can be handled on the cleartext port.</p>
SSLCLIENTAUTH	<p>Specifies how SSL client certificates are authenticated. Valid values are NONE, LOCAL, and PASSTHRU.</p> <p>Configuration of SSL support for use in Data Virtualization Manager server requires that you designate the location of the certificate and keystore that the IBM-supplied SSL components use. The SSL support for the server can be configured to use a pair of native IBM SSL key database and key stash files.</p>	<p>LOCAL (default) The server requests a client certificate during the SSL connection setup handshake. Certificates that are sent by the client are authenticated by using the certificate store that is designated by other SSL startup parameters. They are either a GSK SSL key database, or a RACF keyring.</p> <p>NONE The server does not make SSL client certificate processing active and does not request client certificates.</p> <p>PASSTHRU The server requests a client certificate during the SSL connection setup handshake. Certificates that are sent by the client are not authenticated upon receipt but are available for inspection by the transaction.</p>

Parameter	Description	Valid values
SSLCLIENTNOCERT (<i>Optional</i>)	<p>Specifies the action to take if an SSL client fails to provide a valid x501 certificate during session establishment.</p> <p>Note: The failure by the client to provide a certificate might be because of the lack of mutually trusted signing authority. Lack of a certificate does not prevent the SSL session from being established and used.</p> <p>Note: The SSL handshake at session establishment completes before application of the FAILURE action.</p>	<p>ALLOW (default) Allows the server to continue processing, ignoring failure by the client or in ability to provide a certificate.</p> <p>FAIL The server terminates its session with the client at the earliest possible opportunity.</p>
SSLUSERID	<p>Specifies the user ID under which the SSL resource manager subtask operates. If not specified, the SSL resource manager operates by using the subsystem's address-space-level user ID. This user ID must be authorized to open and read the SSL private key and certificate files. Using a separate user ID for this task prevents other transaction subtasks, and the server itself, from accessing this highly confidential information.</p>	Null

2. To set up the ports, use the **MODIFY PARM** command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

Required Ports:

```
"MODIFY PARM NAME(OEPORTNUMBER) VALUE(XXXX) "  
"MODIFY PARM NAME(WSOEPORT) VALUE(XXXX) "
```

Optional Ports:

```
"MODIFY PARM NAME(OENLPORTNUMBER) VALUE(0) "  
"MODIFY PARM NAME(OESSLPORTNUMBER) VALUE(0) "  
"MODIFY PARM NAME(WSOEBALANCEDPORT) VALUE(0) "  
"MODIFY PARM NAME(WSOESSLPORT) VALUE(0) "
```

Parameter	Description	Valid values
OEPORTRNUMBER	Sets the port number that is used to LISTEN for, and ACCEPT all inbound TCP/IP sessions that should not be considered candidates for load balancing to a different Data Virtualization Manager server in the same load-balancing group. The port number should be reserved for exclusive use by the main product address space. This must be different from the main OEPORTRNUMBER and the OESSLPORTRNUMBER number if it is used.	0 (default)
WSOEPORTR	Specifies the port number that is used to listen for all inbound Services and IBM Data Virtualization Manager studio requests.	0 (default)
OENLPORTRNUMBER (<i>Optional</i>)	Sets the port number that is used to LISTEN for, and ACCEPT all inbound TCP/IP sessions that should not be considered candidates for load balancing to a different Data Virtualization Manager server in the same load-balancing group. The port number should be reserved for exclusive use by the main product address space. This must be different from the main OEPORTRNUMBER and the OESSLPORTRNUMBER number if it is used.	0 (default)
OESSLPORTRNUMBER (<i>Optional</i>)	Sets the port number that is used to LISTEN for, and ACCEPT all inbound encrypted OE Sockets TCP/IP sessions. This port number should be reserved for use only by the main product address space. Each copy of the main product address space needs its own port number if SSL over OE Sockets is being used. There is no default value for the SSL port number if the value is not set in the initialization EXEC.	Null
WSOEBCALANCEDPORTR (<i>Optional</i>)	Specifies the port number that is used to listen for Services requests that can be balanced to group members.	0 (default)

Parameter	Description	Valid values
WSOESSLPORT <i>(Optional)</i>	Specifies the port number that is used to listen for Services for encrypted sessions.	0 (default)

Configuring AT-TLS manually

The IBM® Configuration Assistant for z/OS® Communications Server, an optional GUI-based tool, provides a guided interface for configuring TCP/IP policy-based networking functions.

About this task

You can use the Configuration Assistant to generate the Policy Agent files. You can find more information about the Configuration Assistant for z/OS in the IBM Knowledge Center.

TLS cipher support

As TLS 1.3 and TLS 1.2 are supported differently, if the user chooses the encryption method as TLS 1.2, then only the ciphers supporting TLS 1.2 are sent to the server for handshake.

Similarly, if the user selects the encryption method as TLS 1.3, then only the ciphers supporting TLS 1.3 are sent to the server for handshake.

Note:

TLS 1.3 is incompatible with TLS 1.2 and lower TLS versions. To work with a particular cipher and TLS version, an SSL certificate must be generated with a corresponding supported algorithm.

Also, ensure to set AT-TLS rules on the server appropriately.

For the list of supported ciphers, see the Readme file.

Network access control

This topic describes how to enable network access control for IBM Data Virtualization Manager for z/OS. Enabling network access control for Data Virtualization Manager helps you ensure that only authorized users or applications access Data Virtualization Manager. This prevents unauthorized z/OS users or applications from viewing or even manipulating your mainframe data, either accidentally or on purpose.

Network access control provides an additional layer of security to any authentication and authorization security that is used in your network. With this function, the ability of users to send or receive data between z/OS and certain networks can be controlled through z/OS.

Network access control is implemented by classifying IP addresses into security zones. Security zones consist of one or more IP addresses or IP address ranges that are identified by a zone name. Each security zone has the same level of security sensitivity. Therefore, when a network resource attempts to access Data Virtualization Manager, access is granted or denied based on the access privileges that the requesting network resource has to the security zone. Data Virtualization Manager verifies the security zone access privileges of the incoming IP connection request before validating the user credentials of the z/OS user that initiated the connection request. If the z/OS user is not allowed access to the security zone to which the Data Virtualization Manager belongs, the connection is rejected even before attempting to validate the user ID and password of the z/OS user.

Enabling network access control

Enabling the network access control provides access privileges to specific IP addresses or group of IP addresses that can access Data Virtualization Manager over the network.

About this task

After the SAF resource profile is defined, you need to configure Data Virtualization Manager to enable network access control. You then can assign access privileges to specific IP addresses or group of IP addresses that can access Data Virtualization Manager over the network. A z/OS user can then access Data Virtualization Manager over a TCP/IP network if their user ID has at least READ access to the security zone to which the Data Virtualization Manager belongs.

Perform the following tasks to enable network access control on Data Virtualization Manager.

Procedure

1. To activate network access control, your system administrator must define System Authorization Facility (SAF) resource profiles in the SERVAUTH class. To grant access to users or groups, your administrator must run the following commands:

```
EZB.NETACCESS.system-name.stack-name CLASS(SERVAUTH) ID(id_value) ACCESS(access_privilege)
RDEF SERVAUTH EZB.NETACCESS.system-name.stack-name.zone-name UACC(access_privilege)
```

For example:

```
RDEF SERVAUTH EZB.NETACCESS.**.PROTECT (G) UACC(NONE)
RDEF SERVAUTH EZB.NETACCESS.**.WORLD (G) UACC(READ)
PERMIT EZB.NETACCESS.**.PROTECT CLASS(SERVAUTH) GENERIC ID(USER1) ACCESS(READ)
PERMIT EZB.NETACCESS.**.PROTECT CLASS(SERVAUTH) GENERIC ID(USER2) ACCESS(READ)
```

For more information on SAF resource profiles, see [Local user access control to TCP/IP resources using SAF](#).

2. Use the **MODIFY PARM** command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
"MODIFY PARM NAME(NETACCESSZONEENABLED) VALUE(YES) "
"MODIFY PARM NAME(NETACCESSZONETRACE) VALUE(YES) "
"MODIFY PARM NAME(NETACCESSZONEALLOW) VALUE(YES) "
```

Parameter	Description	Valid values
NETACCESSZONEENABLED	Enable/disable network access control for Data Virtualization Manager.	YES Network access control is enabled. NO Network access control is disabled.
NETACCESSZONETRACE	Enable/disable tracing of the security zones.	YES Security zone tracing is enabled. NO Security zone tracing is disabled.

Parameter	Description	Valid values
NETACCESSZONEALLOW	Permit/deny incoming IP connection request to the security zone.	<p>YES Permits the connection request.</p> <p>Note: Connection is permitted if the security zone is NULL.</p> <p>NO Denies the connection request.</p>

- Use the NETACCESS statement to configure the IP addresses that can access Data Virtualization Manager.

For example:

```
NETACCESS    INBOUND    OUTBOUND    ; CHECK BOTH WAYS
             10.101.8.0/24  AVZ ; SINGLE HOST
ENDNETACCESS
```

For more information on NETACCESS, see [NETACCESS statement](#).

- Optional: You can create a data set that contains the NETACCESS configuration settings. Each time you need to add a new NETACCESS configuration, you can modify this data set and run the OBEY command to activate the configuration:

```
V TCPIP,tcpip,OBEYFILE,DSN=my.ipprofile(obeynet)
```

Here, my.ipprofile(obeynet) is the name of the NETACCESS configuration data set that you created or modified.

Results

An IP connection request to Data Virtualization Manager is permitted only if the requesting IP resource has sufficient access privileges to the security zone to which the Data Virtualization Manager belongs.

Enterprise auditing

Enterprise auditing was created to support the new and unique security requirements of Internet applications, while operating in the traditional enterprise computing environment. With enterprise auditing, Web applications that access z/OS data and transactions can be used by people who do not have mainframe user IDs. Enterprise auditing can also be used with non-Internet applications.

The development of enterprise auditing grew from the need to replace traditional z/OS, UNIX, and NT security architecture, because the architecture could not adequately handle the larger volumes of data that is associated with Internet applications. In addition, traditional user IDs are too costly to create and administer.

Prerequisites for creating a z/OS security environment

Generic IDs can be passed to the z/OS System Authorization Facility (SAF) to create a z/OS security environment for running an RPC.

- The generic IDs must be valid host user IDs.
- The IBM Data Virtualization Manager for z/OS parameter TLSDYNAMICUSERIDS in the AVZSIN00 configuration member must be set to YES.

Note: Setting TLSDYNAMICUSERIDS to YES affects only the SAF processing of generic IDs. All of the other features and facilities can be used even if the TLSDYNAMICUSERIDS is set to NO.

Using generic and extended IDs

IBM Data Virtualization Manager for z/OS implements Enterprise Auditing with a host of related new facilities. All of the facilities are based on two IDs: generic ID and extended ID.

These two IDs are provided in addition to the traditional user IDs supported by IBM Data Virtualization Manager for z/OS. They are optional and can be used either together or separately. In addition, the generic and extended ID values can be used for application debugging, logging, tracing, and auditing purposes. In many respects, they are similar to the user parameter that can be set as part of the ODBC connection initialization; however, they have the advantage that they can be set and/or reset as many times as needed for each connection.

Note: Both the generic ID and extended ID values are only transmitted over the network when they are set for the first time or when they are changed.

The generic ID and the extended ID are supported on the host side using several different mechanisms. Each of these mechanisms is optional and can be used together.

The host mechanisms are as follows:

- SMF per-transaction recording
- Logging
- Trace browse
- Remote users

System Management Facility (SMF) Per-Transaction Recording

By setting the SMFTRANSACT parameter to YES, the SMF per-transaction recording is activated to support the generic ID and the extended ID.

Note: The extended ID area in the SMO6 record has room for only the first 50 bytes of the extended ID. A new record format is provided if the entire extended ID is needed in the future.

Logging

Setting the LOGSQLSOURCE parameter to YES activates per-SQL logging. The generic ID is stored in the **GENERIC_USERID** column, and the extended ID is stored in the **EXTENDED_USERID** column.

Note: The **EXTENDED_USERID** column only has room for the first 254 bytes of the extended ID.

The logging of SQL/transactions is performed on a per-SQL basis using a DB2 table. The default table name is AVZ.SQLSOURCE; however, this default can be changed using the LOGSOURCETABLE product parameter.

Trace Browse

If a generic ID exists, it is contained in the user ID column of Trace Browse for SQL/RPC operations. If the generic ID is set to a non-blank, non-zero value, the generic ID replaces the standard user ID in Trace Browse. This information is only provided for debugging, tracking, tracing, auditing, and so on.

Note: The standard user ID is stored in Trace Browse for non-SQL/ RPC operations (such as network input/output) even if the generic ID is set. This means that both the generic ID and the standard user ID normally appear in Trace Browse for one session.

Remote users

The remote users display includes two new columns for the generic ID and the extended ID, which contain their respective values if they are set.

Host side support

The generic ID and the extended ID are supported on the host side using several different mechanisms. These mechanisms are optional and any can be used together. Several of these mechanisms are intended for application security, auditing, logging, tracing, tracking, and so on.

The host side mechanisms are installation and application specific. The host mechanisms include:

- APIs
- System Management Facility (SMF) per-transaction recording
- Logging
- Trace Browse
- Remote users

Creating a z/OS security environment

The z/OS security environment that is created by passing the generic ID to SAF is maintained during RPC execution and influences what resources the RPC can access.

Note: The generic ID z/OS security environment has no impact on SQL execution authority. The DB2 security environment is initialized when the DB2 thread is created and is not modified later.

RPC authority checking

The generic ID security environment is used to determine the following:

- If the client is allowed to run an RPC.
- If RPC authority checking has been activated by setting the Data Virtualization Manager server parameter **CHECKRPCAUTHORITY** to YES. RPC authority checking uses RACF class/entity rules or ACF2 generalized resource rules to determine if a client is authorized to run an RPC.

Note: RPC authority checking can be used with or without generic ID SAF processing and vice versa.

Caching the z/OS security environment

For performance reasons, the z/OS security environments that are created by passing generic IDs to SAF are cached. That is, each generic ID is passed to SAF only once and the z/OS security environment is cached at the address space level. This approach allows use/reuse of the generic ID security environment with negligible overhead.

Note: There is no IBM Data Virtualization Manager for z/OS Event Facility processing of LOGONs for generic IDs even if ATH rules for LOGON are enabled. The generic ID z/OS security environments are maintained in the cache until the main product address space terminates.

Security considerations

There is a possible security exposure that is associated with using generic IDs with the Data Virtualization Manager server parameter **TLSDYNAMICUSERIDS** set to YES. In this case, a z/OS security environment is created without a password. In addition, client applications are able to use the generic ID z/OS security environment without providing a password. This means that only carefully controlled applications (running inside an Application server/Web server) should be allowed to connect to a copy of IBM Data Virtualization Manager for z/OS that has the parameter **TLSDYNAMICUSERIDS** set to YES. This restriction can be enforced several ways, including LOGON ATH rules.

Note: **TLSDYNAMICUSERIDS** defaults to NO and can only be set to YES by using the Data Virtualization Manager configuration member. **TLSDYNAMICUSERIDS** cannot be set to YES after the main product address space initialization has completed.

Enabling enterprise auditing

Enterprise auditing is enabled by making sure the Data Virtualization Manager Server Event Facility (SEF) ATH parameter ATH.AUPWENTL is set to 1 in an SEF ATH LOGON rule.

About this task

The ATH.AUPWENTL flag is used to control whether enterprise auditing can be used. If it is not set to “1,” the client is not recognized as a secure client, and all enterprise auditing requests from that client are ignored.

This can be done by using the sample SEF ATH LOGON rule, LOGONTLS. This sample rule checks the client IP address, and if it is set to a certain value, the rule sets the ATH.AUPWENTL to “1,” thus allowing enterprise auditing to be used from this connection. The IP address to be checked may be changed to reflect your secured server.

Protected resources

System programmers typically configure advanced security during Data Virtualization Manager server customization. Data Virtualization Manager server provides protection for its resources by using RACF classes, CA Top Secret classes, and CA ACF2 generalized resource rules.

The overall RACF class (or resource type for ACF2) for Data Virtualization Manager is specified with the server parameter RESOURCETYPE. Classes can be shared among multiple instances of servers and either share the authorization rules or keep them separate.

Important: If the RESOURCETYPE parameter is not explicitly specified, the setting defaults to NON, which disables all product authorization checking.

When a user invokes a [Data Virtualization Manager resource](#), the user's ID and the class of the resource are passed to the security program for authorization. The security program uses rules that you specify to determine whether to grant access to the resource.

To expedite future authorization checks of an identical request, Data Virtualization Manager server keeps the results of all security checks in protected storage.

The “look-aside” security check information is saved on a Task Control Block (TCB) basis and remains in effect until the TCB terminates. If you are initially denied access, but later have your security profile that is changed to allow access, you must exit the ISPF/SDF application to terminate its TCB. Depending on the security package, you may have to take other actions. Under ACF2, for example, you must issue the **ACFRESET** command. All security authorization events are logged in the Server Trace facility, and if access is denied, a message is produced.

The type of access you request — ADD/ALTER, READ, or UPDATE — depends on which resource you are using. The ACF2 ADD is equivalent to the RACF ALTER. See [“Access requirements”](#) on page 245 for the type of access that is required to use Data Virtualization Manager facilities.

Enabling security parameters for resource rules

To enable the security parameters, change `if DontDoThis` to `if DoThis`.

```
if DoThis then
do
  "MODIFY PARM NAME(RESOURCETYPE) VALUE(RAVZ)"
end
```

Parameter name	Parameter description	Default value
RESOURCETYPE	<p>RESOURCE TYPE FOR RESOURCE RULES</p> <p>Contains the name of the security server s class (or resource type for ACF2) that is used to perform resource access authorization checks. If not explicitly specified, this parameter defaults to NON.</p> <p>Valid values:</p> <p>NON Disables all product authorization checking.</p> <p>Important: If you leave generalized resource checking disabled, a security exposure may exist. Anyone with a valid TSO user ID can gain access to the Data Virtualization Manager ISPF control application, where they are fully authorized to perform the functions that are provided by the interface. This assumes, however, that the user has sufficient information at hand to log on to TSO/E and then gain access to the ISPF/SDF application.</p> <p>classname RACF class name or ACF2 resource type. When using RACF, the corresponding class name within RACF must start with R, for example, RAVZ.</p>	NON

List of protected resources

The following table describes the resources that are protected by the Data Virtualization Manager security mechanism.

Note: You cannot modify the resource names.

<i>Table 17. Protected resources</i>	
Resource name	Description
ACI.aci-mapname	Access to an ACI (Advanced Communication Interface) service definition.
ADA.ADABAS-file-name	Access to an Adabas file name.
ADATRACE	Authority to issue Adabas TRACE ON and TRACE OFF commands.
ADAXxxxx.FILyyyyy	Access to an Adabas file ID number.
ATHZOOM	Access to Server Trace authorization event PF4 Zoom information.
AVZ	Access to the ISPF/SDF interactive control facility.
CICSCONNECTIONS	Access to monitor and control CICS connections.
CONTROLBLOCKS	Data Virtualization Manager internal data structures.
DATABASES	Access databases that are defined to Data Virtualization Manager.
DATAMAP	Access to the Data Mapping Facility.
FILE	Access to shared files that are defined to Data Virtualization Manager.
FILETYPE	Access to the Data Virtualization Manager file-suffix/MIME-type control table.
GLOBALS	Access to global variables.

Table 17. Protected resources (continued)

Resource name	Description
IMSLTERM	Tables correlating user IDs or TCP/IP addresses to LTERM to legacy LTERM security can be supported using an APPC interface.
LINKS	Access to communication links that are defined to Data Virtualization Manager.
PARMS	Access to the ISPF/SDF parameter display.
RPC.<rpc_name>	RPC-based security.
SEF	Access to the Event Facility dialogs.
SIS	Access to the Instrumentation Server.
TOKENS	Access to the Data Virtualization Manager tokens display.
TRACEBROWSE	Access to the Server Trace facility.
TRACEDATA	Access to all trace data, including SQL and underlying binary file trace records.
USERS	Access to the attached/remote users applications.

Access requirements

The following table provides the type of access that is required to use each Data Virtualization Manager facility.

Table 18. Data Virtualization Manager access requirements

Resources	Action	Suggested user	Access required
ADATRACE	Issuing the ADABASTRACE ON and OFF commands.	DBA, Program Products, VTAM, Operations	READ
ATHZOOM	Viewing Server Trace authorization event PF4 zoom information.	DBA, Program Products, VTAM, Operations	READ
AVZ	Defining links using the ADDRESS AVZ DEFINE LINK command.	DBA, Program Products, VTAM, Operations	ADD/ALTER
CONTROLBLOCK	Using the Data Virtualization Manager command.	DBA, Program Products, VTAM, Operations	READ
CONTROLBLOCK, AVZ	Viewing product control blocks using the ISPF/SDF option AVZ.	DBA, Program Products	READ
CONTROLBLOCK, AVZ	Modifying product control blocks using a future facility.	DBA, Program Products	UPDATE
DATABASES	Viewing databases using the ADDRESS AVZ DISPLAY DATABASE command.	DBA, Program Products, VTAM, Operations	READ

Table 18. Data Virtualization Manager access requirements (continued)

Resources	Action	Suggested user	Access required
DATABASES, AVZ	Modifying databases using the ADDRESS AVZ MODIFY DATABASE command.	DBA, Program Products	UPDATE
GLOBALS	Viewing global variables.	All (DBA, Program Products, Operations, Developers, End-Users)	READ
GLOBALS	Updating global variables.	DBA, Administrator, Developers	UPDATE
IMSLTERM, AVZ	Correlating user IDs or TCP/IP addresses to LTERMs.	DBA, Administrator	READ, UPDATE
LINKS	Viewing links using the ADDRESS AVZ DISPLAY LINK command.	DBA, Program Products, VTAM, Operations	READ
LINKS, AVZ	Modifying links using either the ADDRESS AVZ MODIFY LINK command.	DBA, Program Products, VTAM, Operations	UPDATE
LINKS, AVZ	Defining databases using the ADDRESS AVZ DEFINE DATABASE command.	DBA, Program Products	ADD/ALTER
PARMS, AVZ	Modifying the product parameters the ADDRESS AVZ MODIFY PARM command.	DBA, Program Products, VTAM, Operations	UPDATE
PARMS, AVZ	Viewing all Server Trace data.	DBA, Program Products, VTAM, Operations	READ
SEF, DATAMAP	Refreshing Data Maps	DBA, Admin	READ access to SEF; UPDATE access to DATAMAP.
TRACEBROWSE, TRACEDATA, AVZ	Issuing SQL statements via AVZSPUFI.	DBA, Program Products, VTAM, Operations	READ
USERS, AVZ	Viewing remote users the ADDRESS AVZ DISPLAY REMOTE command.	DBA, Program Products, VTAM, Operations	READ
USERS, AVZ	Killing remote users using the ISPF/SDF option AVZ Admin / AVZ Group	DBA, Operations, Developers, End-Users	READ, UPDATE
USERS, AVZ	Viewing product Data Virtualization Manager parameters using the ADDRESS AVZ DISPLAY PARM command.	DBA, Program Products, VTAM, Operations	READ

Defining resources to RACF

The Data Virtualization Manager server protects its resources using the Resource Access Control Facility (RACF) class added to the RACF class descriptor table.

Procedure

1. Use the following JCL as a model for defining a new RACF class to the RACF class descriptor table for RAVZ.

```
//STEP1 EXEC ASMHCL
//C.SYSLIB DD DSN=SYS1.MODGEN,DISP=SHR
//C.SYSIN DD *
RAVZ ICHERCDE CLASS=RAVZ,
      ID=128,
      MAXLNTH=39,
      FIRST=ALPHANUM,
      OTHER=ANY,
      POSIT=25,
      OPER=NO
      ICHERCDE
/*
//L.SYSLMOD DD DSN=SYS1.LINKLIB,DISP=SHR
//L.SYSIN DD *
      INCLUDE SYSLMOD(ICHRRUDE)
      ORDER RAVZ
      ORDER *** Previous user-defined classes ***
      ORDER *** Previous user-defined classes ***
      ORDER ICHRRUDE
      NAME ICHRRUDE(R)
/*
```

Restart the Data Virtualization Manager server so that RACF recognizes the new class.

2. Perform an IPL to change the RACF class descriptor table. This procedure is necessary for RACF to recognize the new class.
3. Define all RACF resource types to class RAVZ with the following command:

```
RDEFINE RAVZ CONTROLBLOCKS UACC(NONE)
```

Repeat the RDEFINE command for each RACF resource type.

4. Provide access to the resource according to the following example:

```
PERMIT CONTROLBLOCKS CLASS(RAVZ) ID(USERID) ACCESS(READ)
```

Where USERID is the ID of the user to whom you want to grant READ permissions access.

If you do not want the FACILITY class to be used, the `h1q.SAVZCNTL(AVZRADF2)` member can be used as a sample for how to define the RACF class descriptor and router table.

You can edit and submit the job in `h1q.SAVZCNTL(AVZRARES)` to define and add permissions for the resource required by your site.

5. Activate the class to RACF with the following command:

```
SETROPTS CLASSACT(RAVZ)
```

What to do next

These members must be updated every time a new security resource name such as ATHZOOM or USERS is added.

Defining resources to CA Top Secret

The Data Virtualization Manager server provides protection for its resources using the CA Top Secret classes.

Procedure

1. Define an entry in the RDT, as shown in the following example:

```
TSS ADDTO(RDT) RESCLASS(AVZ) RESCODE(nn) -  
ATTR(LONG, PRIV, LIB, DEFPROT, GENERIC) -  
ACLST(NONE, ALL, ALTER=1C00, UPDATE, READ) DEFACC(READ)
```

Where *nn* is a hexadecimal code between 01 and 3F.

2. Add all the resources to an owner with the following commands:

```
TSS ADDTO(owner) AVZ(CONTROLBLOCKS)
```

Repeat this TSS ADDTO command for all resource types.

3. Permit the resources to profiles or users as follows:

```
TSS PERMIT(userid) AVZ(TRACEDATA) ACC(READ)
```

4. You can edit and submit the job in *hlq.SAVZCNTL(AVZTSRES)* to define and add permissions for the resource required by your site.

What to do next

These members must be updated every time a new security resource name such as ATHZOOM or USERS is added.

Defining resources to ACF2

Data Virtualization Manager server protects its resources using CA ACF2 generalized resource rules. Data Virtualization Manager server allows you to define and add permissions for the resource required in your site.

Procedure

1. Define a generalized resource class named AVZ.
2. Define resource rules for each of the resource class. Member *hlq.SAVZCNTL(AVZA2RES)* can be used as an example.
3. Use the following ACF2 command to allow users access to the resource rule:

```
ACFNRULE KEY(TRACEBROWSE) TYPE(AVZ) ADD(UID(*****userid) ALLOW
```

4. You can edit and submit the job in *hlq.SAVZCNTL(AVZA2RES)* to define and add permissions for the resource required by your site.

Optional security jobs

The optional security jobs can be located in the *hlq.SAVZCNTL* library.

The following table lists the jobs that are in the *hlq.SAVZCNTL* library. The jobs can be edited and submitted for the purpose that is specified in the Description column.

Description	RACF	CA ACF2	CA Top Secret
ACI persistent connection security	AVZRAACI	AVZA2ACI	AVZTSACI
ADABAS file name or file ID	AVZRAADA	AVZA2ADA	AVZTSADA

Table 19. Optional security jobs (continued)

Description	RACF	CA ACF2	CA Top Secret
BPEL role-based security	RACFBPEL	ACF2BPEL	
CICS transaction security	AVZRACIC	AVZA2CIC	AVZTSCIC
DB2 RRSAP security	AVZRADB2	AVZ2DB2	AVZTSDB2
IDMS transaction security	AVZRAIDM	AVZA2IDM	AVZTSIDM
IMS OTMA and transaction	AVZRAIMS	AVZA2IMS	AVZTSIMS
Permissions that are required for JVM installation	AVZRAJVM	AVZA2JVM	AVZTSJVM
MQ security for Streams	AVZRAMQ	AVZA2MQ	AVZTSMQ
IBM Data Virtualization Manager for z/OS Resource security	AVZRARES	AVZA2RES	AVZTSRES
IBM Data Virtualization Manager for z/OS RPC security	AVZRARPC	AVZA2RPC	AVZTSRPC
Defining IBM Data Virtualization Manager for z/OS started task to security product	AVZRAVDB	AVZA2VDB	AVZTSVDB
RRS XA-2PC security	AVZRAXA	AVZA2XA	AVZTSXA
Streams security	AVZRASTR	AVZA2STR	AVZTSSTR

ISPF load modules

If you use TSO Command to restrict access to TSO commands, you must define the IBM Data Virtualization Manager for z/OS ISPF load modules to your security product.

Table 20. IBM Data Virtualization Manager for z/OS load modules

Load module	Description
AVZ	TSO command to invoke S__ interactive application.
AVZ2RU	Routine to invoke IBM Data Virtualization Manager for z/OS ISPF application.
AVZI	REXX Implicit Interpreter TSO Command processor.
AVZICOMP	REXX Implicit Interpreter TSO Command processor.
AVZIDB	REXX Implicit Interpreter TSO Command processor.
AVZIMEX	REXX Implicit Interpreter TSO Command processor.
AVZOB	Alias for AVZOCP.
AVZOCP	Trace Browse routine.
AVZORU	Trace Browse routine.
AVZX	REXX Implicit Interpreter TSO Command processor (Server REXX).
AVZXCOMP	REXX Implicit Interpreter TSO Command processor.
AVZXDB	REXX Implicit Interpreter TSO Command processor.
AVZXSCAN	REXX Implicit Interpreter TSO Command processor.
AVZHOCM	Host command environment for address AVZ.

Table 20. IBM Data Virtualization Manager for z/OS load modules (continued)

Load module	Description
AVZISCBR	Display product control blocks.
AVZSSTRU	Display product statistics.
AVZISTBR	General-purpose table display routine.
AVZSVARU	ISPF product variables display.
AVZLINK	Main product module.
AVZRXHR	Browse routine for REXX S__ line variables.
AVZRXDM	A REXX function to call new DMF parser.
AVZRXID	A REXX function for issuing commands to IDCAMS.
AVZRXIM	Initialize the REXX environment.
AVZRXLLK	Bridge REXX TO LE/370 main routine.
AVZRXPC	Product-related control block function.
AVZRXSG	REXX function for examining storage in another address space.
AVZRXST	Product-related control block function.
AVZRXTE	Terminate REXX environment.
AVZRXTK	REXX function for parsing strings into token.
AVZRXVA	REXX function for manipulating variables in a calling REXX exec.

RACF PasSTickets

The RACF PasSTicket can be used instead of a user logon password.

About this task

When you use a RACF PasSTicket, the default application name that is passed is the three-character subsystem ID code (for example, AVZ for IBM Data Virtualization Manager for z/OS) appended with the system SMFID. This application name must match a PTKTDATA profile name for PasSTicket generation and authentication to work. For example, if the system SMFID is DEV1, the application name is AVZDEV1, and you must define a PTKTDATA profile for IBM Data Virtualization Manager for z/OS with the name AVZDEV1. The default application name can be changed by using the PASSTICKETAPPNAME parameter.

Also, a PTKTDATA profile name can be further qualified by RACF user ID and/or RACF connect group (for example, AVZDEV1 . AVZS or AVZDEV1 . SYS1 . AVZS). This allows different instances of an application to have unique single sign-on keys.

For more information on defining profiles in the PTKTDATA class, see the [z/OS Security Server RACF Security Administrator's Guide](#).

Defining security for RPCs

The Data Virtualization Manager server allows you to define all RPCs to your security product and grant RPC access to the specific users.

About this task

Use the following procedure if you need to restrict access to RPCs:

Procedure

1. Use the MODIFY PARM command to add the following parameters that are located in the AVZSIN00 configuration member:

```
"MODIFY PARM NAME(ACF/2SAFCALL) VALUE(NO)"
"MODIFY PARM NAME(CHECKRPCAUTHORITY) VALUE(YES)"
"MODIFY PARM NAME(RESOURCETYPE) VALUE(RAVZ)"
"MODIFY PARM NAME(TRACEAUTHEVENTS) VALUE(YES)"
```

Parameter	Description	Valid values
ACF/2SAFCALL (<i>ACF2 Only</i>)	To use RPC security with ACF2, you must run a version of ACF2 that supports SAF calls.	YES Enables Data Virtualization Manager server to use SAF calls for Resource Rules. NO (default) All users are allowed to run all RPCs. The RPC can always provide its own security.
CHECKRPCAUTHORITY	Controls whether the SEF and ACF2/RACF should be used to check whether each user has the authority to run each RPC. If set to YES, the SEF and ACF2/RACF are used to verify RPC execution authority.	YES The SEF and ACF2/RACF is used to verify RPC execution authority. NO (default) All users are allowed to run all RPCs. The RPC can always provide its own security.
RESOURCETYPE	Contains the name of the security server's class (or resource type for ACF2) that is used to perform resource access authorization checks.	
TRACEAUTHEVENTS (<i>Optional</i>)	Turn on authorization event tracing (this allows you to trace the RPC security checks).	YES NO Default value is NO.

2. If you want to define all RPCs to your security product and grant RPC access to the specific users, you must edit and submit one of the following sample jobs that are located in the *hlq.SAVZCNTL* library.
 - AVZRARPC for RACF security
 - AVZA2RPC for CA ACF2 security
 - AVZTSRPC for CA Top Secret security

Note: If you enable the CHECKRPCAUTHORITY parameter, you must define each RPC to your security product.

Information access with the TRACEDATA resource

The TRACEDATA resource controls access to information in the trace log.

About this task

The two types of information that are contained within the Data Virtualization Manager server trace log:

- SQL source statements (the real SQL source statements, as taken from database request modules or prepared strings, which may contain objects such as table names or column names).
- Binary data that underlies the trace log.

Users who have READ authority for the TRACEDATA resource and READ authority for AVZ and TRACEBROWSE can view the entire trace log. Users who do not have READ authority have only restricted access to this information.

For SQL events, if your user ID matches the user ID associated with the event, you are permitted to look at an uncensored log of the SQL event. Otherwise, you can only see a censored representation of the SQL statement. The censored version includes the SQL verb but does not include objects, such as table names or column names.

The TRACEDATA resource restricts data differently, depending on the type of event:

- SQL Events: If your user ID matches the user ID associated with the event, you are permitted to look at an uncensored log of the SQL event. Otherwise, you can only see a censored representation of the SQL statement. The censored version includes the SQL verb but does not include objects, such as table names or column names.
- Non-SQL Events: If your user ID matches the user ID associated with the event, you are permitted to see an uncensored view of the underlying binary data for event. Otherwise, you are not allowed to see the binary data at all; no data is displayed and a message is written to the terminal.

Resource security for test versions of Data Virtualization Manager server

All resource security is simulated for test versions of the Data Virtualization Manager server running in a TSO session. The z/OS security subsystem is not consulted, because a test TSO copy of the product is not authorized to perform this type of security check. All work is performed using the TSO user's existing z/OS authorizations.

In this environment, all security checks are assumed to complete successfully. If you are running test copies of the Data Virtualization Manager server under TSO, you should find this feature helpful in deploying new applications, because you can review the security checks that occur when the application is deployed in a production environment.

Configuring Adabas security

Configure security to access Adabas data at a DBID or file number level.

About this task

Securing Adabas files at a DBID or file number level requires the use of the following Data Virtualization Manager server parameters:

- RESOURCETYPE
- SQLVTRESOURCETYPE
- ADABASSEURITY

The following sample jobs for defining Adabas security-related definitions are provided in the *hlq.SAVZCNTL* library:

- AVZRAADA for RACF

Note: When using job AVZRAADA, make the following changes for file ID security:

```
RDEFINE FACILITY ADAxxxxx.FILyyyyy UACC(NONE)
PERMIT ADAxxxxx.FILyyyyy CLASS(FACILITY) ID(<USERID>)
ACCESS(aaaa)
SETROPTS REFRESH RACLIST(FACILITY)
```

- Change xxxxx to the Adabas database ID.

- Change *yyyyy* to the Adabas file ID.
- AVZA2ADA for CA ACF2
- AVZTSADA for CA Top Secret

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Installation and Customization Guide*.
2. Ensure the following settings are set in the AVZIN00 file:

```
MODIFY PARM NAME(RESOURCE TYPE) VALUE(RAVZ)
MODIFY PARM NAME(SQLVTRESOURCE TYPE) VALUE(RAVZ)
MODIFY PARM NAME(ADABAS SECURITY) VALUE(YES)
```

Parameter name	Parameter description	Value
RESOURCE TYPE	RESOURCE TYPE FOR RESOURCE RULES Specify the name of the security server's class (or resource type for ACF2) that is used to perform resource access authorization checks. When using RACF, the corresponding class name within RACF must start with R, for example, RAVZ.	For RACF: RAVZ
SQLVTRESOURCE TYPE	RESOURCE TYPE FOR SQL ACCESS TO VIRTUAL TABLES Specify the name of the security server's class (or resource type for ACF2) that is used to perform authorization checks for SQL access to metadata and virtual tables in the SQL Engine. When using RACF, the corresponding class name within RACF must start with R, for example, RAVZ.	For RACF: RAVZ
ADABAS SECURITY	ADABAS SECURITY ACTIVATED Set this parameter to indicate that a resource rule is to be constructed consisting of DBID and file. Note: Both RESOURCE TYPE and SQLVTRESOURCE TYPE must be set in order for ADABAS SECURITY to be in effect.	YES

Configuring CICS security

The server uses the External CICS Interface (EXCI) to access CICS programs. EXCI uses the CICS multiregion operation (MRO). EXCI is subject to the same security checks as a CICS system that uses MRO to connect to another CICS system.

For more information, please refer to the *IBM CICS RACF Security Guide*.

MRO login security

Configure MRO login security so that the server started task uses a specific MRO connection.

If you start the server started task, and the **ALLOCATE_PIPE** request fails with **RESPONSE(SYSTEM_ERROR)**, **REASON(IRC_LOGON_FAILURE)**, and **subreason1= 204(decimal)**, you must set up MRO logon security for the server. The server user ID must be authorized with UPDATE authority to the **DFHAPPL.net_name** RACF profile, where *net_name* is the name that is used in the **NETNAME()** parameter of the connection definition in the Data Virtualization Manager configuration file.

Since RACF does a separate check for LOGON security, you need to issue the following command:

```
RDEFINE FACILITY (DFHAPPL.netname) UACC(NONE)
PERMIT DFHAPPL.netname CLASS(FACILITY) ID(AVZS) ACCESS(UPDATE)
SETROPTS REFRESH RACLIST(FACILITY)
```

MRO bind time security

MRO bind time security ensures that the server started task is authorized to establish a connection, or bind, to a CICS region.

If you start the server started task, and the **OPEN_PIPE** request fails with `RESPONSE(SYSTEM_ERROR)`, `REASON(IRC_CONNECT_FAILURE)`, and `subreason1= 176(decimal)`, you must set up MRO bind time security for the server. The server user ID must have READ authority to profile `DFHAPPL.applid`, where *applid* is the ID for the CICS region that is specified in the **APPLID()** parameter of your connection definition in your Data Virtualization Manager configuration file.

If you start a server started task, and the **ALLOCATE_PIPE** request fails with `RESPONSE(SYSTEM_ERROR)`, `REASON(IRC_LOGON_FAILURE)`, and `subreason1= 204(decimal)`, you need to set up MRO logon security for the server. The server user ID must be authorized with UPDATE authority to `DFHAPPL.net_name` RACF profile, where *net_name* is the name that is used in the **NETNAME()** parameter of your connection definition in your Data Virtualization Manager configuration file.

Because RACF does a separate check for LOGON security, issue the following command:

```
RDEFINE FACILITY (DFHAPPL.netname) UACC(NONE)
PERMIT DFHAPPL.netname CLASS(FACILITY) ID(AVZS) ACCESS(UPDATE)
SETROPTS REFRESH RACLIST(FACILITY)
```

Link security

The target CICS system checks link security against requests from the server using the server region's user ID.

These security checks cover the following:

- Transaction attach security (when attaching the mirror transaction).
- Resource and command security within the Server application program.

If you have transaction and program security that is enabled in your CICS region, make sure that the server user ID can access the mirror transaction and the CICS Server program. Also, if the CICS Server program accesses a protected resource, make sure the server user ID can access the same protected resource.

User security

User security applies only when `ATTACHSEC(IDENTIFY)` is specified on the CICS EXCI connection definition.

The user ID, which is the same user ID used for logging on to the MVS system where the server resides, are passed in the DPL request and are used to perform the same checks as link security.

Notes:

- User security can never attain more privileges than link security allows.
- If you use ACF2 and want to associate the user ID that makes the connection from the server to the CICS transaction, add the following MRO statement in the ACF2/CICS parameter file for this CICS region:

```
MRO SYSID=****,FORMAT=CICS
```

Where ******** is the CICS connection name that is defined in CICS.

Virtual table SAF security

A single Data Virtualization Manager server environment can provide data virtualization to multiple independent tenants or application groups. The virtual table SAF (system authorization facility) security feature provides a SAF mechanism to secure virtual tables so that each tenant can only access tables authorized for members of the tenant group.

Activating this security feature will prevent using virtual table names in metadata queries (such as, **SQLENG.TABLES**, **SQLENG.COLUMNS**), as well as querying or updating application data mapped using unauthorized table names.

Server interface parameter

The SQLVTRESOURCETYPE parameter in the PRODSECURITY parameter group defines a security class name for virtual table resource checking. By default, this system parameter defaults to the value 'NON' indicating that security checking is disabled.

When activated with a class name, the SQLVTRESOURCETYPE parameter will enable SAF resource checking on metadata queries (such as, **SQLENG.TABLES**, **SQLENG.COLUMNS**) as well as virtual table queries using the resource name *resource_class.table_owner.table_name* where:

- *resource_class* is the class name define for the RESOURCETYPE parameter in the PRODSECURITY parameter group (for example, RAVZ)
- *table_owner* is the SQL TABLE OWNER NAME (SQLENGTABLEOWNER) as defined in the PRODSQL parameter group (for example: 'DVSQ')
- *table_name* is the map (or virtual table) name as defined in the map data set

For improved performance in SAF calls, RACROUTE REQUEST=FASTAUTH provides general resource checking. A separate INTRNLONLY parameter named 'DISABLE FASTAUTH SECURITY CHECKS' disables use of FASTAUTH if security problems are encountered. Disabling FASTAUTH will switch to RACROUTE REQUEST=AUTH checking on all resource rules which can degrade query performance on metadata tables.

When securing metadata tables, READ access is required to query rows in the following tables.

Tables	Populated state
SQLENG.COLUMNS	Y
SQLENG.COLUMNPRIVS	N
SQLENG.FOREIGNKEYS	N
SQLENG.PRIMARYKEYS	Y Dependent on tables with keys such as VSAM KSDS, IMS, specific DRDA sources, ADABAS, etc.
SQLENG.ROUTINES	N
SQLENG.SCHEMAS	Y There are two defaults: SQLENG and DVSQ for non-schemas.
SQLENG.SPECIALCOLS	Y ALT Index for specific columns, DRDA with multiple index, ADABAS super descriptor
SQLENG.STATISTICS	Y Dependent on RUNSTAT

Tables	Populated state
SQLENG . TABLES	Y T for virtual tables and V for virtual views
SQLENG . TABLEPRIVS	N

Securing tables using the generic profile SQLENG . * is also an option if preferred.

Securing specific virtual tables is also required when activating this feature. Securing virtual tables by specific or generic rules activates two security checks:

1. When querying metadata tables (SQLENG . *), users must minimally have READ access to the virtual tables in order for rows related to a table to be returned. In this case, there are no errors returned. Instead, the information about a specific table is omitted from the result set and the user has no indication that the table exists.
2. When querying virtual tables, the user must have READ access to each table in the SQL SELECT statement and UPDATE access to any table that is the target of an SQL INSERT, UPDATE, or DELETE statement.

Restrictions and Considerations

Virtual table authorization checking is built on general resource checking and is impacted by the following product parameter in the PRODSECURITY group:

- ALLOWUNPROT – The ALLOWUNPROT parameter allows access to unprotected resources. When set to YES, this parameter allows access to resource names that have no matching resource definition in the SAF database. ALLOWUNPROT should be set to NO to insure resource rules are correctly processed.

Note: ALLOWUNPROT=NO will automatically activate numerous resource checks unrelated to this feature.

The *table_owner.map_name* resource name is internally restricted to 44 bytes. While internal map names larger than 44 bytes are still allowed, resource checking will only pass the first 44 bytes of the *table_owner.map_name* string in the SAF call for validation. Generic resource rules will be necessary if map names exceed this limitation.

Because all maps are limited to a single table owner as defined in the SQLENGTABLEOWNER system parameter, users should consider a standard prefix for all map names they want to secure for application groups. This simple generic resource rules can be defined to protect these names. For example, if the SQLENGTABLEOWNER is configured as 'DVSQ' and an application group uses AG01 as a prefix on all table names, a generic resource 'DVSQ . AG01*' will control access to all tables starting with AG01 as a map name.

All SQL queries are automatically secured when this feature is activated. This means that resource rules must exist to allow READ access to the metadata tables SQLENG.*.

This feature is limited to SQL access to virtual tables. Users authorized to create tables can create tables which may not be accessible due to SQL access rules implemented using this feature.

Chapter 13. Performance

The Data Virtualization Manager server provides a number of features to enhance performance.

- Workload management – Using the IBM Workload Manager for z/OS, you can define performance goals and priorities. The system matches its resources to the work and determines whether goals are being met by monitoring and adapting its processing.
- Multiple servers – Running separate Data Virtualization Manager servers addresses server needs for testing or for distributing your workload.
- Load balancing – With load balancing, inbound connections are automatically directed to server instance that has the most available resources.
- CICS failover – This feature allows you to set up an alternate CICS ID for each CICS connection, so that if access to a primary CICS connection fails, a hot failover is performed to the alternate CICS region.
- System resource management – Several system resources, including block fetch, time limit alerts, and session failure detection maintain response times within pre-established services levels.
- Virtual Connection Facility – This feature increases the number of client connections possible.
- Enterprise transaction support – The IBM Data Virtualization Manager for z/OS Enterprise supports various two-phase commit protocols, including the Resource Recovery Services attachment facility (RRSAF).

System resources

IBM Data Virtualization Manager for z/OS provides several system resources that are used to streamline the management of Data Virtualization Manager server performance, helping to maintain response times within pre-established services levels as numbers of users grow from a few to tens of thousands.

These resources include:

- Block fetch
- CPU time limits
- Wait time for all clients
- Program execution duration time limit
- Session failures
- Dispatch priority

CPU time limits

Data Virtualization Manager server supports an internal CPU time limit for Data Virtualization Manager client and an external CPU time limit for all clients.

Internal CPU time limit

Data Virtualization Manager server provides an internal CPU time limit mechanism that limits the amount of CPU time a Client can use before it is disconnected from the host. The limit ensures that a remote Client connection does not continue by using CPU time even after the client becomes hung.

Note: The limit applies to every session and is reset each time a new session starts.

If a Client connection exceeds the CPU time limit, Data Virtualization Manager server cancels the connection and issues a message to the client and to the Server Trace log.

The time limit mechanism is activated only after a unit of work is received from the Client. It only monitors connections that are made to DB2.

Note: The internal CPU time limit mechanism does not detect time out conditions and does not stop runaway queries.

The initial time limit value is obtained from the ACF2 lid control block.

External CPU time limit

Data Virtualization Manager server provides an external CPU time limit mechanism that limits the amount of CPU time a client can use before it is disconnected from the host, avoiding runaway queries and other CPU loops.

Note: The limit applies to every session and is reset each time a new session starts.

This mechanism involves the following limits:

- **Warning Limit.** A warning message is written to the trace browse when a task exceeds the warning limit.
 - When the excess CPU time utilized during a connection exceeds the value set in WARNINGCPU TIME, the external time limit mechanism generates a CPU warning message.
 - When the connection's idle or inactive duration exceeds the value set in WARNINGWAIT TIME, the external wait time limit mechanism generates a wait warning message.

The format of this message is:

```
xVy4280T, %1 TIME LIMIT EXCEEDED FOR %2 FROM %3 NODE %4 PLAN %5 CNID %6 TP %7
```

where:

- %1: The timtyp parameter specifies the time limit type. The available options are CPU or WAIT.
- %2: The userid parameter specifies the user ID string.
- %3: The rntype parameter specifies the type of the remote node. The available options are TCP/IP or LU.
- %4: The host parameter specifies the remote host name.
- %5: The pln parameter specifies the plan name string.
- %6: The cnid parameter specifies the connection identifier value
- %7: The pgm parameter specifies the program name.

For example, the following is the warning message for CPU time limit mechanism:

```
xVy4280T CPU TIME LIMIT EXCEEDED FOR userid FROM TCP/IP/LU NODE x.x
```

- **Error Limit.:** An error message is written to the trace browse when a task exceeds the error limit.
 - When the excess CPU time utilized during a connection exceeds the value set in ERRORCPU TIME, the external time limit mechanism generates a CPU error message.
 - When the connection's idle or inactive duration exceeds the value set in ERRORWAIT TIME, the external wait time limit mechanism generates a wait error message.

The format of this message is:

```
xVy4281T %1 TIME LIMIT EXCEEDED FOR %2 FROM %3 NODE %4 PLAN %5 CNID %6 TP %7
```

where:

- %1: The timtyp parameter specifies the time limit type. The available options are CPU or WAIT.
- %2: The userid parameter specifies the user ID string.
- %3: The rntype parameter specifies the type of the remote node. The available options are TCP/IP or LU.
- %4: The host parameter specifies the remote host name.

- %5: The pln parameter specifies the plan name string.
- %6: The cnid parameter specifies the connection identifier value
- %7: The pgm parameter specifies the program name.

For example, the following is the error message for CPU time limit mechanism:

```
xVy4281T CPU TIME LIMIT EXCEEDED FOR userid FROM TCP/IP/LU NODE x.x
```

- **Failure Limit.:** An error message is written to the trace browse, and the connection will be terminated with an UA322 or UA522 abend.

The system generates a U0322 abend when the following timeout or limit is exceeded:

- When the CPU time consumed during a connection exceeds the limit in FAILCPUPTIME.
- When the execution time for a single SQL statement exceeds the threshold in FAILSQLCPUPTIME.
- When a connection holds a DB2 exclusive lock for longer than the duration specified in FAILEXCLUSIVETIME.
- When a connection holds a DB2 shared lock for longer than the duration specified in FAILSHARETIME.
- When a connection maintains a DB2 lock for longer than the duration specified in FAILUPDATETIME.

When the connection's idle or inactive duration exceeds the value set in FAILWAITTIME, the external wait time limit mechanism generates a U0522 abend.

Note: The client does not receive a message that describes why the connection was terminated; a TCP/IP input output error occurs when the user tries to perform the next operation.

Enabling time limits

Data Virtualization Manager server provides an external error, failure, and warning time limits for all clients.

Procedure

To enable the CPU time limits, use the MODIFY PARM command to add the following parameters to the AVZSIN00 configuration member:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(CHECKLIMITSINTERVAL) VALUE(15 SECONDS)"
  "MODIFY PARM NAME(ERRORCPUPTIME) VALUE(0 SECONDS)"
  "MODIFY PARM NAME(ERRORWAITTIME) VALUE(0 SECONDS)"
  "MODIFY PARM NAME(FAILCPUPTIME) VALUE(0 SECONDS)"
  "MODIFY PARM NAME(FAILWAITTIME) VALUE(0 SECONDS)"
  "MODIFY PARM NAME(WARNINGCPUPTIME) VALUE(0 SECONDS)"
  "MODIFY PARM NAME(WARNINGWAITTIME) VALUE(0 SECONDS)"
```

Parameter	Description	Valid values
CHECKLIMITSINTERVAL	Specifies how often, in seconds, each client task is checked for a violation of the execution limit. The interval value is specified in seconds and should be a factor of one hour. The interval value should divide evenly into 3600 (one hour).	1 – 3600 seconds 15 seconds (default)
ERRORCPUPTIME	When set to 0 seconds, the parameter is disabled.	0 seconds (default)

Parameter	Description	Valid values
ERRORWAITTIME	When set to 0 seconds, the parameter is disabled.	0 seconds (default)
FAILCPUPTIME	When set to 0 seconds, the parameter is disabled.	0 seconds (default)
FAILWAITTIME	When set to 0 seconds, the parameter is disabled.	0 seconds (default)
WARNINGCPUPTIME	When set to 0 seconds, the parameter is disabled.	0 seconds (default)
WARNINGWAITTIME	The external wait time limit specifies how long that a connection can remain disabled. When set to 0 seconds, the parameter is disabled.	0 seconds (default)

Enabling the program execution duration time limit mechanism

When an RPC program begins execution, the starting time is recorded. Periodically, the elapsed time for all tasks that are running RPC programs is calculated and compared to the RPC duration limit value. If an elapsed time exceeds the limit, the task in which the RPC program is running may be forced to terminate.

About this task

An exception event is scheduled before termination. An SEF EXC rule, which is scheduled to handle the event, might elect to extend the time limit and continue execution, or allow the task to be terminated abnormally.

Note: This limit is applied to total elapsed time while an RPC program is run. The program might be running normally, or it might be stalled. This limit does not test whether the RPC program is, or has, consumed CPU cycles during the elapsed time interval. The limit is applied to customer-written RPC programs. The limit is not applied to built-in CALL RPC programs available in the server or to native DB2 stored procedures that are governed by the PER-SQL time limit.

Procedure

To enable the time limit, use the MODIFY PARM command to add the following parameters to the AVZSIN00 configuration member:

```
if 1 = 1 then
do
"MODIFY PARM NAME(RPCDURATIONLIMIT) VALUE(0 SECONDS)"
```

Parameter	Description	Valid values
RPCDURATIONLIMIT	If set to a non-zero value, the parameter imposes an elapsed time limit for all RPC program executions. The value is expressed in seconds. When set to 0 seconds, no elapsed time limitation is enforced. The maximum allowed value is 86,400 seconds, equal to 24 hours.	0 (default)

Modifying the client auxiliary storage cut-off parameter

You can specify at what point the Data Virtualization Manager server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility.

About this task

The Data Virtualization Manager server listens for ENF 55 auxiliary storage shortage signals and throttles storage utilization when an auxiliary storage shortage is signaled.

The Accelerator Loader server will perform the following actions depending on the received ENF 55 signal:

- When signal ENF55QLF_AUX_WARNING is received:

1. Issue the following message:

```
AVZ4265W Data Server Client buffer expansion disabled due to auxiliary storage warning
```

2. Disable Data Virtualization Manager server buffer expansion for two hours and ten minutes.
3. Issue the following message:

```
AVZ4266I Data Server Client services resumed
```

- When signal ENF55QLF_AUX_SHORTAGE is received:

1. Disable Data Virtualization Manager server buffer expansion.
2. Issue the following message:

```
AVZ4265W Data Server Client buffer expansion disabled due to auxiliary storage shortage
```

- When signal ENF55QLF_AUX_CRITICAL_SHORTAGE is received:

1. Disable Data Virtualization Manager server buffer expansion.
2. Issue the following message:

```
AVZ4265W Data Server Client buffer expansion disabled due to auxiliary storage critical shortage
```

3. Disable new Data Virtualization Manager server requests.
4. Issue the following message:

```
AVZ4267W Data Server Client refusing new requests due to critical auxiliary storage shortage.
```

- When signal ENF55QLF_AUX_SHORTAGE_RELIEVED is received:

- Re-enable all Data Virtualization Manager server functions.
- Issue the following message:

```
AVZ4266I Data Server Client services resumed.
```

The point at which the Data Virtualization Manager server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility is controlled by the **DSCLIENTAUXSTGCUTOFF** parameter.

To change the value, complete the following steps.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Installation and Customization Guide*.
2. Use the **MODIFY PARM** command to change the **DSCLIEN TAUXSTGCUTOFF** parameter value:

```
"MODIFY PARM NAME(DSCLIEN TAUXSTGCUTOFF) VALUE(WARNING) "
```

Parameter name	Parameter description	Default value
DSCLIEN TAUXSTGCUTOFF	<p>DSCLIEN TAUX STORAGE NEW CONNECTION CUTOFF</p> <p>Specifies at what point the Data Virtualization Manager server will reject new connection attempts when an auxiliary storage shortage is signaled by the system Event Notification Facility.</p> <p>WARNING New Data Virtualization Manager server connections will be rejected when an auxiliary storage warning is received. This signal is issued when message IRA205I occurs.</p> <p>SHORTAGE New Data Virtualization Manager server connections will be rejected when an auxiliary storage shortage is signaled. This signal is issued when message IRA200E occurs.</p> <p>CRITICAL New Data Virtualization Manager server connections will not be rejected until an auxiliary storage critical shortage is signaled. This signal is issued when message IRA201E occurs.</p>	WARNING

Detecting sessions failures

The Data Virtualization Manager server can detect session failures while processing is active.

About this task

This type of failure occurs when a user submits a long-running SWL statement or RPC and then stops the application or restarts the server that hosts the application. When either condition occurs, Data Virtualization Manager server detects that the session failed and kills the SQL statement or RPC.

Procedure

To enable session failure detection, use the MODIFY PARM command to add the following parameters to the AVZSIN00 configuration member:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(CHECKSESSIONS) VALUE(YES)"
  "MODIFY PARM NAME(SESSIONFAILTIME) VALUE(15 SECONDS)"
```

Parameter	Description	Valid values
CHECKSESSIONS	Controls whether to periodically check the status of each session. When a session failure is detected, all work that is running on the host on behalf of the client is terminated.	YES Check the status of each session periodically. NO (default) Do not check check the status of each session.
SESSIONFAILTIME	Specifies the number of seconds that a remote application task (a task that is running on behalf of a client) can be in processing state (RPC, SQL, REXX) before the product checks whether the network session is still active.	15 (default)

Results

If the application is terminated by using Task Manager (or the UNIX equivalent), the host processing terminates in a few seconds (default is 15 seconds). If the client system is rebooted or part of the network fails, the host does not know about the failure until the KEEPALIVE (TCP/IP parameter) timeout occurs. The KEEPALIVE timer is usually set to 20 minutes, but it can be configured to suite your requirements.

Workload Manager (WLM)

Using the IBM Workload Manager for z/OS, you can define performance goals and assign a level of importance to each goal in business terms. The system matches its resources to the work and determines whether goals are being met by monitoring and adapting its processing. This allows you to make the best possible use of the server's resources, while achieving the best possible response times.

Goals are specified for the WLM services in IBM Data Virtualization Manager for z/OS in the same way they are specified for z/OS-managed work, by associating work with a service class. The assigned service class informs the operating system about the performance goal and importance level that is associated with the work, as well as the address spaces involved in processing the work request.

Support for the Workload Manager (WLM) is available for the SQL data access. For information about planning for and using workload management, refer to the documentation for the *MVS Planning: Workload Management* and *MVS Workload Management Services* documents.

WLM enclaves

To facilitate implementation of transaction management, WLM uses enclaves. An enclave is a group of one or more logically related z/OS task control blocks (TCB) and service request blocks (SRB) that manage the work in entities.

Using enclaves provides the following benefits:

- Work running in enclave SRBs can be offloaded to a zIIP processor. The Data Virtualization Manager server runs in enclave SRB mode, when possible, to allow CPU offloading.

- The resources that are used to process the transaction can be accounted to the transaction rather than to the address space in which the transaction runs. Service class performance goals are inherited by the enclave.

The Data Virtualization Manager server establishes a logical dispatchable unit (LDU) for each process and thread in its address space. This LDU consists of a TCB/SRB pair that is dispatched in SRB mode in a WLM enclave, if possible, switching to TCB mode only if required by system or database interfaces. The SRB mode execution is eligible for offloading to a zIIP based on the definitions in the WLM service policy.

During installation, the Data Virtualization Manager server establishes two long-running enclaves. One is the service class AVZ_SCNM, and the other is the service class AVZ_SCHI. Dispatchable units join these enclaves as appropriate. Unique enclaves are created as needed for the processes and threads for SQL data access.

Configuring Workload Manager (WLM)

You use WLM to define performance goals and assign a level of importance to each goal in business terms.

The system then matches its resources to the work, as well as monitors the goals and makes necessary processing adoptions accordingly.

This section explains several ways that you can configure WLM support and provides the definitions that are required to use the support.

WLM definitions

A service definition is the name that is given to the combination of service policies, workloads, service classes, resource groups, classification rules, and application environments. It is based on the performance objectives in a service level agreement (SLA). The following is a list of WLM definitions:

Workload

A named group of work, or service classes, that is reported as a unit.

Service Class

A named group of work that has similar performance goals, resource requirements, or importance. In the service class, you assign each goal and its relative importance, and associate the service class with a specific workload and resource group. IBM Data Virtualization Manager for z/OS requires the following service classes.

- AVZ_SCHI ZIIPCLASS=AVZ High priority. This service class is for IBM Data Virtualization Manager for z/OS critical work. Assign this class goal as close to SYSSTC as possible.
- AVZ_SCNM ZIIPCLASS=AVZ Normal work. This class is for IBM Data Virtualization Manager for z/OS administrative work. Assign this class the same goals as those used for DB2 master or the IMS control region.
- AVZ_SCTX ZIIPCLASS=AVZ Client work. This service class is for client requests. Assign this class the same goals as those supporting the data source. This would most likely be the CICS, IMS/TM, or DB2 WLM address space.

Classification Rules

A classification rule maps work coming into the system to a specific service class and report class.

A classification is based on the subsystem type and work qualifiers in the subsystem type. The work qualifiers define and associate service classes to the type of work.

Report Class

A named group of work that is for reporting purposes only. Use report classes to distinguish among types of work that run in the same service class.

Providing WLM definitions via Data Virtualization Manager

Data Virtualization Manager provides information about WLM definitions. These definitions describe the separation and affinity requirements for a set of dynamic transactions.

Before you begin

Before you start this procedure, it is important to understand the following requirements:

- Data Virtualization Manager must have proper access to the MVSADMIN.WLM.POLICY resource.
- Your user ID must have UPDATE access or the following error occurs:

```
*SDx0038S INSTALL OF WLM SERVICE DEFINITION FAILED, RC=X'0000000C',  
REASON=X'xVy0039s', DETECTED AT OPINWM+X'FFC3BF06'
```

- Your user ID for starting the server must have READ access or the following error occurs:

```
SDx3269I WLM administration userid xxxxxxxx logged on to system  
  
xVy0037E WLM EXTRACT SERVICE DEFINITION FAILED, RC=X'00000004', DETECTED AT  
OPINWM+X'00000B02'
```

Procedure

1. Add the following statements to your AVZSIN00 configuration member:

```
If DoThis then  
  do  
    "MODIFY PARM NAME(WLMFORCEPOLICY) VALUE(YES)"  
    "MODIFY PARM NAME(WLMTRANNAME) VALUE(APPLNAME)"  
    "MODIFY PARM NAME(WLMUSERID) VALUE(AVZS)"  
  End
```

The following table lists the parameters for WLM definitions:

Parameter	Description	Valid values
WLMFORCEPOLICY	Controls whether the Data Virtualization Manager server enforces service policy requirements.	<p>YES</p> <p>The server initialization examines the active policy for required elements and terminates if the elements do not exist and the server is not allowed to add them. The server also examines the policy anytime it is refreshed, and shuts down the server if the new policy is not in compliance with server requirements.</p> <p>NO</p> <p>(default) The Data Virtualization Manager server checks the policy for required definitions, and issues an error message if the subsystem type (default AVZ, set by WLMSUBSYSTEM) is not defined in the policy. The server is allowed to initialize, and is not shut down for any policy changes.</p>
WLMTRANNAME <i>(optional)</i>	Specifies which value is used as the transaction name when classifying the Data Virtualization Manager server transactions.	<p>APPLNAME</p> <p>(default) The application name set in the client ODBC data source is used as the transaction name.</p> <p>MODNAME</p> <p>The name of the application that uses the client ODBC driver is used as the transaction name.</p> <p>INTNAME</p> <p>The client application executable internal name is used as the transaction name.</p>
WLMUSERID <i>(optional)</i>	<p>Specifies a highly privileged user ID under which WLM administration functions are performed. This user ID must be authorized to update the MVSADMIN.WLM.POLICY resource.</p> <p>If WLMUSERID is not specified, the server subsystem ID is used for WLM policy administration.</p>	AVZS (default subsystem ID)

2. Enter WLM from the ISPF/PDF option 6 panel to log on to the IBM TSO/ISPF WLM administration tool.
3. Extract and save a copy of the current service definition. This is for backup purposes only.
4. Optional: Update the AVZSIN00 configuration member with a valid WLMUSERID.
5. Start Data Virtualization Manager server.

Upon startup, Data Virtualization Manager:

- Examines the current WLM service policy for the required elements. If the active policy contains the required elements, initialization continues. If the required elements are not found, Data Virtualization Manager messages xDy0706I, and xDy0707I are issued for each missing element.

```
xDy0706I DATA VIRTUALIZATION SERVER AVZS requires the following elements
missing from WLM Service Policy active_policy_name

xVy0707I Type: WORKLOAD, Data Virtualization Parameter: WLMWORKLOAD,
Value: AVZ_WKLD

xVy0707I Type: SUBSYSTEM, Data Virtualization Parameter: WLMSUBSYSTEM,
Value: AVZ

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMSERVICECLASS, Value: AVZ_SCNM

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMHISERVICECLASS, Value: AVZ_SCHI

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMTXSERVICECLASS, Value: AVZ_SCTX
xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMRP1REPORTCLASS, Value: AVZ_RCP1

xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMRP2REPORTCLASS Value: AVZ_RCP2

xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMRP3REPORTCLASS, Value: AVZ_RCP3

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTRANSACTION, Value: AVZ_TNNM

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMHITRANSACTION, Value: AVZ_TNHI

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTXTRANSACTION, Value: AVZ_TNTX
```

- Data Virtualization Manager then examines the WLM service definition for the required elements. If WLMFORCEPOLICY is set to NO, the following actions are skipped. If WLMFORCEPOLICY is set to YES, the following actions are enforced. The default is NO.

Action 1: If the required elements are found in the service definition, a WTOR is issued, requesting permission to activate the current service policy. If the current policy is no longer in the service definition, the user is asked to select one of the policies in the service definition for activation.

```
*nn xVy0719R Reply 'GO' to activate Policy
service_policy_name, or 'CANCEL' to terminate
Server initialization
```

If you reply with CANCEL, the Data Virtualization Manager server shuts down.

If you reply with GO, the Data Virtualization Manager server automatically activates your WLM Policy *service_policy_name*, and you should see the following message in the system log:

```
IWM001I WORKLOAD MANAGEMENT POLICY service_policy_name NOW IN EFFECT
```

- **Action 2:** If the required elements are not found in the service definition, the Server issues message xDy0706I, and then message xDy0707I for each missing element.

```
xDy0706I DATA VIRTUALIZATION SERVER AVZS requires the following elements
missing from WLM Service Definition service_definition_name.
```

```

xVy0707I Type: WORKLOAD, Data Virtualization Parameter: WLMWORKLOAD,
Value: AVZ_WKLD

xVy0707I Type: SUBSYSTEM, Data Virtualization Parameter: WLMSUBSYSTEM,
Value: AVZ

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMSERVICECLASS, Value: AVZ_SCNM

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMHISERVICECLASS, Value: AVZ_SCHI

xVy0707I Type: SERVICE CLASS, Data Virtualization Parameter:
WLMTXSERVICECLASS, Value: AVZ_SCTX

xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMMP1REPORTCLASS, Value: AVZ_RCP1

xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMMP2REPORTCLASS, Value: AVZ_RCP2

xVy0707I Type: REPORT CLASS, Data Virtualization Parameter:
WLMMP3REPORTCLASS, Value: AVZ_RCP3

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTRANSACTION, Value: AVZ_TNNM

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMHITRANSACTION, Value: AVZ_TNHI

xVy0707I Type: CLASSIFICATION RULE, Data Virtualization Parameter:
WLMTXTRANSACTION, Value: AVZ_TNTX

```

The preceding messages are followed by a WTOR requesting permission to update the service definition.

```

*nn xVy0708R Reply 'GO' to update the WLM Service Definition, or
'CANCEL' to terminate server initialization

```

If you reply with CANCEL, Data Virtualization Manager server shuts down.

If you reply with GO, the Data Virtualization Manager server automatically makes the proper WLM updates to your WLM policy definition. At the conclusion of the update process, you receive the following message.

```

xDy0709I WLM Service Definition service_definition_name has been updated
with required elements

```

Action 3: A separate WTOR message is presented to activate the policy.

```

*nn xVy0719R Reply 'GO' to activate Policy service_policy_name, or
'CANCEL' to terminate server initialization

```

If you reply with CANCEL, Data Virtualization Manager server shuts down. The user can use the TSO/ISPF WLM administration dialog to extract the service definition and review the additions that are made by the Data Virtualization Manager server.

If you reply with GO, the Data Virtualization Manager server automatically activates your WLM policy *service_policy_name*, and you see the following message:

```

IWM001I WORKLOAD MANAGEMENT POLICY service_policy_name NOW IN EFFECT

```

Note: After the WLM service policy is activated, if you change any IBM Data Virtualization Manager for z/OS required WLM element in the service definition to an invalid value and activate a service policy, all servers requiring the now invalid definition shut down.

Note: You should have a backup of your existing WLM service policy definitions.

Providing WLM definitions manually

WLM manually provides information about the WLM definition parameters. These definitions describe the separation and affinity requirements for a set of dynamic transactions.

About this task

If you want to manually define the required WLM definitions rather than have the Data Virtualization Manager server automatically install them at startup time, take the following steps:

Procedure

1. Start the WLM administration tool. The IBM TSO/ISPF WLM administration tool is used in the following examples. Other administrative tools can also be used.
 - a) Enter **WLM** from the ISPF/PDF option 6 panel to log on to the IBM TSO/ISPF WLM administration tool.
 - b) Select **Option 2 Extract Definition from WLM Couple Data Set** from the Choose Service Definition box.
2. Define the workloads.
 - a) Select **Option 2 Workloads**. Press Enter.
WLM displays the Workload Selection List panel.
 - b) Create workload AVZ_WKLD.
3. Define the service classes.
 - a) Select **Option 4 Service Classes**. Press Enter.
WLM displays the Service Class Selection List panel.
 - b) Here you create the following service classes:
 - AVZ_SCHI ZIIPCLASS=AVZ IBM Data Virtualization Manager for z/OS high priority
 - AVZ_SCNM ZIIPCLASS=AVZ IBM Data Virtualization Manager for z/OS normal work
 - AVZ_SCTX ZIIPCLASS=AVZ IBM Data Virtualization Manager for z/OS client work
4. Define subsystem type AVZ and its classification rules.
 - a) Select **Option 6 Classification Rules**. Press Enter.
WLM displays the Subsystem Type Selection List for Rules panel.
 - b) Define subsystem type AVZ and associated classification rules.
5. Define the report classes.
 - a) Select **Option 7 Report Classes**. Press Enter.
WLM displays the Report Class Selection List panel.
 - b) In this panel, create the following report classes:
 - AVZ_RCP1 Dnnnn Pnnn PERIOD 1
 - AVZ_RCP2 Dnnnn Pnnn PERIOD 2
 - AVZ_RCP3 P100 PERIOD 3

Note:

- Do not change report class names.
- The terms in the report class descriptions are used to provide CPU offload criteria for Data Virtualization Manager server work as follows:

Dnnnn: The number of service units during which the dispatchable units are in the associated period while eligible for offloading to the zIIP processor.

Pnnn: The percentage of time in the associated period that Data Virtualization Manager server tries to offload work to the zIIP processor.

6. Activate a Service Policy.

- a) Select **Option 3 Activate Service Policy** from the **Utilities** drop-down menu on the panel.
- b) Follow directions to activate a policy.

Using the WLM Administration Tool

This procedure describes the method to use the IBM Workload Manager for z/OS, you can define performance goals and assign a level of importance to each goal in business terms.

Procedure

1. Enter the following command to start the IBM TSO ISPF administration tool:

```
TSO WLM
```

2. Follow all prompts until the **Choose Service Definition** panel is displayed.
3. Type 2 to select the **Extract definition from WLM couple data set** option.
4. Press **ENTER**. The **WLM Definition** panel appears. You can select the option for the task that you want to perform.

Workload Manager definitions

During initialization, Data Virtualization Manager server connects the server address space to the WLM and ensures that WLM elements are in the current active service policy.

WLM Element Type	Server Parameter	Default Value
Workload	WLMWORKLOAD	AVZ_WKLD
Subsystem	WLMSUBSYSTEM	AVZ
Service Class	WLMSERVICECLASS	AVZ_SCNM
Service Class	WLMHISERVICECLASS	AVZ_SCHI
Service Class	WLMTXSERVICECLASS	AVZ_SCTX
Classification Rule	WLMTRANSACTION	AVZ_TNNM
Classification Rule	WLMHITRANSACTION	AVZ_TNHI
Classification Rule	WLMTXTRANSACTION	AVZ_TNTX
Report Class	WLMP1REPORTCLASS	AVZ_RCP1
Report Class	WLMP2REPORTCLASS	AVZ_RCP2
Report Class	WLMP3REPORTCLASS	AVZ_RCP3

Modifying the workload

The workload, AVZ_WKLD, is required by the Data Virtualization Manager server.

About this task

To modify the IBM Data Virtualization Manager for z/OS workload definition:

Procedure

Select the **Workloads** option from the **WLM Definition** panel (see [“Using the WLM Administration Tool”](#)). Press Enter.

The system displays the **Modify a Workload** panel.

Results

Note: You can change the **Workload Name** field by using the WLMWORKLOAD parameter, which is located in the Data Virtualization Manager configuration member, AVZSIN00. Do not change this name unless instructed to do so by IBM Software Support.

Modifying a service class definition

Modifying a service class definition task is used to lock the original service definition for editing and work with a temporary (or working) copy of the service definition. The changes you make to the temporary copy are not reflected in the original service definition until you save and close the copy. After you close the temporary copy, the lock is automatically released.

Before you begin

For details about setting up service class definitions, refer to the documentation for the *MVS Planning: Workload Management* and *MVS Workload Management Services* documents.

- The AVZ_SCHI service class is used for high importance server work, such as management tasks of short duration that should not be interrupted, establishing a new thread for a new transaction.
- AVZ_SCNM is the default service class for all work that is not explicitly classified, except for the following types of work:
 - Server process LDUs that are assigned AVZ_SCHI.
 - SQL transactions are classified according to WLM classification rules. If a Data Virtualization Manager server classification rule is added that assigns SQL transactions to AVZ_SCNM or AVZ_SCHI, the LDU representing the transaction is joined to one of the long-running enclaves that are established for the transaction task. A new enclave is created for this LDU.
- The AVZ_SCTX service class is used for SQL transactions that are not otherwise classified. A new enclave is created for each LDU assigned to AVZ_SCTX.

About this task

To modify the service class definition:

Procedure

1. Select the **Service Classes** option from the **WLM Definition** panel. Press Enter.

The system displays the **Service Class Selection List** panel.

2. Select a definition in the service class selection list. Select Enter.

The system displays the following panel that shows the default definition for the AVZ_SCNM service class.

The description contains the following information:

- The service class name can be modified by using the WLMHISERVICECLASS parameter, which is located in the AVZSIN00 configuration member.

Note: Do not change this name unless you are told to do so by Technical Support.

- The ZIIPCLASS=AVZ in the description field is used to construct the names of report classes that have CPU offload criteria that are specified in their descriptions. If the ZIIPCLASS keyword is not specified correctly, IBM Data Virtualization Manager for z/OS work that is dispatched as enclave SRBs assigned to this service class is not off loaded to the zIIP.

- The workload name is for reporting purposes only and can be changed to any valid workload name.
- The service class goal is a single period with an execution velocity goal. The percentage and importance can be changed, but set them at a level appropriate to a mission-critical server.

Viewing subsystem and classification rules

View the Data Virtualization Manager server classification rules.

About this task

The following classification rules are required:

- The subsystem type must be AVZ.
- A rule classifying transaction AVZ_TNHI to service class AVZ_SCHI.
- A rule classifying transaction AVZ_TNNM to service class AVZ_SCNM.
- A rule classifying transaction AVZ_TNTX to service class AVZ_SCTX.

Procedure

1. Select the **Classification Rules** option from the **WLM Definition** panel (see [“Using the WLM Administration Tool”](#)).
2. Select AVZ from the list of rules in the classification rules selection.
3. Press **ENTER**. The system displays the **Modify Rules for the Subsystem Type** panel that shows the default definition for the AVZ classification rules.

Results

Do not change the classification rules. They are used internally by the Data Virtualization Manager server. Classification rules for SQL, Streams, and Services can be added to these rules.

Modifying a report class definition

The report class is modified to produce various WLM data for reporting on all of the service definition terms by service class period and workload.

Before you begin

The following report classes are required by the Data Virtualization Manager server:

- AVZ_RCP1
- AVZ_RCP2
- AVZ_RCP3

About this task

To modify a report class definition:

Procedure

1. Select the **Report Classes** option from the **WLM Definition** panel. Press Enter.
The system displays the **Report Class Selection List** panel.
2. Select a report class name from the list of report classes. Press Enter.
The system displays the following panel that shows the default definition for the report class.
The panel shows the following information:
 - The report class definition is used to provide first period CPU offload information for service classes. The report class is not used in the classification rules.

- The report class name can be modified by using the WLM`P1REPORTLASS` parameter, which is located in the AVZSIN00 configuration member.

Note: Do not change this name unless you are told to do so by IBM Software Support.

The format of the report class name is:

`xxx_RCP1`

where `xxx` is a ZIIPCLASS=`xxx` specification on a service class description and `_RCP1` is fixed and must not be changed.

- The `Dnnnn` in the description field is the first period duration in service units for CPU offloading. The `n` value can be adjusted by the user.
- The `P100` in the description field is the percentage of time in the first period that WLM attempts to offload enclave SRBs in the associated service class to the zIIP.

WLM classification rules

WLM Classification rules depends on the transaction classes. These transactions are targeted to separate servants, each with goals defined by specific service classes.

WLM classification rules apply to the SQL solution.

Note: Before defining classification rules, make sure that WLM is installed and set up correctly.

SQL

The Data Virtualization Manager server establishes a unique enclave for each transaction. WLM classification rules can assign this enclave to a service class with velocity or response goals and one or more periods.

WLM populates the enclave definition with the following information:

- Client User ID. WLM uses the client user ID to find a classification rule match. The client user ID is mapped to the WLM qualifier type UI.
- DB2 Plan Name. WLM uses the DB2 plan name to find a classification rule match. The DB2 plan name is mapped to the WLM qualifier type PN.
- DB2 Subsystem Name. WLM uses the DB2 subsystem ID to find a classification rule match. The DB2 subsystem name is mapped to the WLM qualifier type SPM.
- Transaction Name. WLM uses the transaction name to find a classification rule match, depending on the following transaction name values. The transaction name is mapped to the WLM qualifier type TN.
 - APPLNAME: (Default) The application name that is specified in the client data source is used as the transaction name.
 - MODNAME: The name of the application by using the Data Driver is used as the transaction name.
 - INTNAME: The application executable internal name is used as the transaction name.

Using WLM classifications

You can allow WLM to use their existing service and report classes instead of using the hard-coded IBM Data Virtualization Manager for z/OS definitions.

Procedure

1. Set the following parameters that are located in the AVZSIN00 configuration file. Set the values of the WLMUSERID and WLMTRANNAME parameters to names already in your policy so that IBM Data Virtualization Manager for z/OS is correctly classified.

```
if 1 = 1 then
do
```

```
"MODIFY PARM NAME(WLMUSERID) VALUE(AVZS)"  
"MODIFY PARM NAME(WLMTRANNAME) VALUE(APPLNAME)"
```

2. If your Data Virtualization Manager configuration member, AVZSIN00, does not match your existing WLM definitions, add the following parameter to your AVZSIN00 member, and keep the default value NO:

```
if 1 = 1 then  
do  
"MODIFY PARM NAME(WLMFORCEPOLICY) VALUE(NO)"
```

Note: If you set WLMFORCEPOLICY to NO, and the service class and report class descriptions are not correct, the zIIP offload criteria is unavailable and the default value of 100% is used for all IBM Data Virtualization Manager for z/OS enclaves. The service and report classes to which reference is made are those set (or defaulted) in the Data Virtualization Manager configuration member, AVZSIN00, for the WLMnREPORTCLASS and WLM*SERVICECLASS parameters.

Activating the WLM service policy

IBM Data Virtualization Manager for z/OS allows you to activate the WLM service policies for the servers.

About this task



Warning: If you change a required element to an invalid value or remove a required definition and activate a service policy, all active servers that require that definition are shut down.

Procedure

1. After you edit the service definition, select **Utilities** from the **WLM Definition** panel (see [“Using the WLM Administration Tool”](#)).
2. From the **Utilities** menu, select the **Install Definition** option to save the updated service definition.
3. Use the **Activate Service Policy** option to activate a service policy.

Verifying WLM classification

You can verify the WLM classifications using the Trace Browse option in the Data Virtualization Manager server.

Procedure

1. Make sure the following started task parameter is added to the AVZSIN00 configuration member:

```
"MODIFY PARM NAME(TRACEWLMCALLS) VALUE(YES)"
```

This activates tracing for Data Virtualization Manager server calls made to the WLM APIs for transaction management.

2. Connect with your application, and run a transaction.
3. Go to the Data Virtualization Manager server **Primary Option** menu, and select the **Trace Browse** option. Press Enter.
4. The system displays a panel that shows the trace (trace lines are wrapped for the purposes of easier viewing).

The panel shows an ODBC connection that is created from Data Virtualization Manager studio to a Data Virtualization Manager server, and an update that is sent to a DB2 table by using this connection. The AVZSIN00 member contains the following command:

```
"MODIFY PARM NAME(WLMCLASSTRAN) VALUE(YES)"
```

The following classification rule was added to the default rules installed by the Data Virtualization Manager server for subsystem AVZ.

```
_____ 1 TN * ___ SDH_SCTX _____
```

The Trace Browse shows the following WLM operations that occurred:

- WLM enclave join executed. The LDU for the new connection thread is joined to the long running AVZ_SCHI enclave to initialize the thread.
- WLM classify work executed. The LDU for the new connection thread is classified to the AVZ_SCTX service class.
- WLM enclave create executed. An enclave is created using the AVZ_SCTX service class for the new connection thread.
- WLM offload CPU time executed. This shows the call to WLM with the criteria for offloading this enclaves SRB work to the zIIP. The durations and percentages for the offloading are obtained from the AVZ_RPCn report class definitions.
- WLM enclave leave executed. The LDU leaves the AVZ_SCHI enclave that it joined to initialize the thread.
- WLM enclave join executed. The LDU is joined to the AVZ_SCTX enclave that was created for it in a preceding step. This is where the actual transaction work is done. In this case, an update is made to the DB2 table USERID.STAFF.
- WLM enclave leave executed. Processing of the DB2 update is complete, and the LDU leaves the AVZ_SCTX enclave.
- WLM enclave join executed. The LDU rejoins the AVZ_SCHI enclave for thread termination.
- WLM enclave delete executed. The AVZ_SCTX enclave is deleted.

WLM Health Reporting

Data Virtualization Manager server reports to WLM on its relative "health" by issuing the IWM4HLTH macro with a health indicator between 0% and 100%. Data Virtualization Manager server starts with a health indicator of 100%.

This reporting is enabled by using the WLMHEALTHREPORT parameter, which by default is set to YES.

Periodically, Data Virtualization Manager server examines indicators and adjusts its health percentage. If failures, such as ACI timeouts and ACI abends, occur, the health percentage is adjusted down. The higher the failure rate, the larger the adjustment. If no failures occur, the health percentage is adjusted up. To set the interval for this parameter, use the WLMHEALTHINTERVAL.

Configure WLM health reporting by using the following parameters in the AVZSIN00 member.

Parameter	Description	Valid values
CONCURRENTMX	The maximum number of concurrent sessions, which may be open with the server. This limit is enforced such that new connection requests are rejected if the total number of active sessions would exceed this limit. Setting this limit to zero causes all new connections to be rejected, while allowing in-flight sessions to remain active.	2000 (default)
WLMHEALTHINTERVAL	Controls how often health statistics are reported to WLM. Interval is in seconds.	60 (default)

Parameter	Description	Valid values
WLMHEALTHREPORT <i>(optional)</i>	Controls whether the Data Virtualization Manager server reports its health percentage to WLM.	YES (default) Data Virtualization Manager server uses the current rates of ACI timeouts and ABENDS to compute a change in the health percentage reported to WLM. NO
WLMMAXHEALTH <i>(optional)</i>	Controls the current health value reported to WLM.	0 – 100 (default value is 100)

You can examine the current level of health by looking at the value for WLMHEALTH by selecting AVZ **Admin > AVZ Parm**s > **PRODWLM** from the IBM Data Virtualization Manager for z/OS **Server – Primary Option** menu.

You can change the current health value by using the WLMMAXHEALTH parameter.

This parameter allows the CONCURRENTMX parameter to work with the SHAREPORTWLM parameter. Setting CONCURRENTMX parameter to zero forces WLMMAXHEALTH to zero. Setting CONCURRENTMX from zero to nonzero forces WLMMAXHEALTH to 100.

Server load balancing

With load balancing, inbound connections are automatically directed to the server instance that has the most available resources. To determine which instance handles a request takes into account the number of connections currently being handled by each instance and the availability of virtual storage (above and below the 16-MB line).

Load balancing is transparent to the client application. An application uses a port number to connect to an instance, and the instance determines whether it or another instance should handle the session. If another instance is a better choice, the session is transferred.

The client application can be configured with the port numbers of more than one member of the group. This configuration improves reliability by providing a fallback if the copy of the product that uses the base port number is not available. Load balancing increases the number of concurrent connections that the server can handle. As a practical matter, this feature supports far more connections by using RPCs to be concurrently handled. This feature is a key point, because connections by using RPCs exhaust the virtual storage resources of a server instance much faster than DB2 connections.

Sequence of events

Load balancing enables IBM Data Virtualization Manager for z/OS to:

1. Find the address space of the first server.
2. Verify whether sufficient virtual storage exists. If enough storage is available, the server is marked as a candidate.
3. Check the number of active connections.
4. Repeat steps 1 - 3 until all the server candidates are identified.

From the available candidates, the server that has the least number of active connections is selected. If a candidate is not found, the connection is rejected with an inadequate host resources error message.

The group concept

Load balancing is based on the concept of a group. All copies of the server on one system with the same group name are automatically members of the same group. A copy of the product can be a member of

only one group at a time or it can be configured to be a stand-alone server and not as a member of a group. All address spaces in a group must reside on the same z/OS image.

The group name can be changed at any time, which means copies of the server can join groups or leave groups as needed.

Using a group director

You can also define a server as a group director. The load balancing group director does no work except route connections to other Servers. A group director passes connections to the best candidate server in the group, giving the user the option to not run application work in a server that accepts inbound connections. When using a group director, everyone connects to the group director, which then routes all of the connections to other Data Virtualization Manager servers.

If you have one Data Virtualization Manager server act as a group director for load balancing, this server would not perform application work, but would only route requests to other servers based on current loads. The server that is performing as a group director has the greatest stability. This method ensures that a connection is made even if some of the application servers are down. This method gives you the highest availability.

If you do not use a group director, the same Data Virtualization Manager server that is routing requests also does application work. If that server fails, all subsequent connection requests fail.

Enabling load balancing for a group director

This section describes the method to enable the load balancer for a group directory.

Procedure

1. Use the MODIFY PARM command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
if 1 = 1 then
do
"MODIFY PARM NAME(GROUPNAME) VALUE(NULL)"
"MODIFY PARM NAME(GROUPDIRECTOR) VALUE(YES)"
```

The following table lists the parameters for configuring load balancing:

Parameter	Description	Valid values
GROUPNAME	Controls the group that the current instance belongs to. All instance that belongs to the same group (that is, have the same GROUPNAME) automatically load balance among each other. If this value is not set, then the current instance does not belong to a group.	Null
GROUPDIRECTOR	Indicates that a member of the group takes the role of director.	YES NO Default value is NO.

2. Ensure that everyone connects to the instance that is the group director.
3. Add extra Data Virtualization Manager server subsystems, and set the maximum number of connections in each.

Enabling load balancing for CICS/TS

This section describes the method to enable the load balancer for CICS/TS.

Procedure

Use the **MODIFY PARM** command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(CICSLoadBALANCE) VALUE(YES)"
  "MODIFY PARM NAME(GROUPDIRECTOR) VALUE(YES)"
```

The following table lists the parameters for configuring load balancing for CICS/TS:

Parameter	Description	Valid values
CICSLoadBALANCE	Specifies whether to use the CICS transaction queue depth to decide about load balancing.	YES NO Default value is NO.
GROUPDIRECTOR	Indicates that a member of the group takes the role of director. The director only accepts inbound connections and pass them to a member of the group that is determined to be the most acceptable in terms of load and resource availability. The group director does not support an application execution environment. This configuration provides a more robust load balancing group.	YES NO Default value is NO.

Enabling load balancing for Services

A load balancing port is added to IBM Data Virtualization Manager for z/OS Services classes services. To prevent disruptions with existing services and the IBM Data Virtualization Manager studio, the current WSOEPORT remains non-load balanced.

Procedure

Use the **MODIFY PARM** command to set the following parameters that are located in the Data Virtualization Manager configuration member, AVZSIN00:

```
if 1 = 1 then
do
  "MODIFY PARM NAME(WSOEBALANCEDPORT) VALUE(0)"
  "MODIFY PARM NAME(ZSRVGROUPNAME) VALUE(XXXXXXXX)"
  "MODIFY PARM NAME(ZSRVGROUPDIRECTOR) VALUE(YES)"
```

The following table lists the parameters for configuring load balancing for Services:

Parameter	Description	Valid values
WSOEBALANCEDPORT	Specifies the port number on which to listen for requests.	0

Parameter	Description	Valid values
ZSRVGROUPNAME	Controls which Services group, if any, the current copy of the product should belong to. The product uses groups for load balancing across multiple copies (separate subsystems) of the product. All copies of the product that belong to the same group (that is, have the same GROUPNAME) automatically load balance between each other. If this parameter is not set, then the current copy of the product does not belong to any Services group.	NULL
ZSRVGROUPDIRECTOR	Indicates that a member of the Services group takes the role of director. The director accepts only inbound connections and passes them to a member of the group, which is determined to be the most acceptable in terms of load and resource availability. The group director does not support an application execution environment. This provides a more robust load balancing group.	YES NO Default value is NO.

To enable load balancing without changing the client, users must use the current WSOEPORT as the WSOEBALANCEDPORT and use a brand new port as WSOEPORT.

All group members must share Virtualization Facility libraries (or at least exact copies) and all Virtualization Facility caches must be refreshed after any change. No automated mechanism is in place to synchronize the Virtualization Facility caches. Also, the connection names that are defined in the AVZSIN00 configuration member for all the group members must match the names that are defined in the CICS Target Systems. Each connection has a unique netname that identifies the CICS definition. If this is done, the servers can share Virtualization Facility libraries.

CICS failover

Use CICS failover to set up an alternate CICS ID for each CICS connection, so that if access to a primary CICS connection fails, a hot failover is performed to the alternate CICS region.

CICS failover is on a per Data Virtualization Manager server basis. That is, when you define a CICS connection in your Data Virtualization Manager server, you can specify a primary CICS region and a failover CICS region.

When the primary CICS region is unavailable, the Data Virtualization Manager server automatically routes new and in-flight CICS calls to the failover CICS region. From then on, all transactions are routed to the failover CICS region even after the primary CICS region becomes available again. It is up to the user to decide when to switch back to the primary CICS again.

To switch back to the primary CICS region without disruption, you can log on to the failover CICS, and set the connection that is used by the Data Virtualization Manager server to “Out-Of-Service.” This causes the Data Virtualization Manager server to begin routing transactions back to the primary CICS. Once that

takes place, you can put the connection in the failover CICS back “In-Service,” so it can start handling failover support again

This failover support works for both XA and non-XA clients. The only time that in-flight transactions cannot be routed is when an XA data source is used, and your CICS transaction is not a one-for-one (one distributed program link (DPL) request per one unit of work (UOW)). That is, if you have a CICS UOW that consists of multiple DPL requests, and one of the DPL requests fails, this CICS transaction is backed out. This situation is not common because CICS transactions are one-for-one.

Enabling CICS failover

Data Virtualization Manager offers the option to enable a secondary CICS connection if the primary CICS connection fails.

Procedure

Add the following parameter to the DEFINE CONNECTION statement in the AVZSIN00 configuration member:

```
ALTAPPLID(xxxxxxxx)
```

Where *xxxxxxxx* is the application ID (APPLID) of the alternate CICS connection. The ALTAPPLID is used the connection to the primary CICS region fails.

Note: The ALTAPPLID must have the same named connection definitions and application definition that are in the target CICS APPLID.

Block fetch

Block fetch pre-extracts rows and sends them in blocks to the requesting node. This process improves the performance of most queries by minimizing network traffic and by using data that is already on the node to accommodate subsequent queries.

Data Virtualization Manager server only uses block fetch with read-only queries. This type of query occurs in the following situations:

- The SELECT statement has a FOR FETCH ONLY clause.
- The SELECT statement has an ORDER BY clause.
- The SELECT statement’s first FROM clause contains more than one table (or view).
- The SELECT statement has the UNION or UNION ALL operator.
- The SELECT statement has the DISTINCT keyword in the first SELECT clause.
- The SELECT statement has a column function in the first SELECT clause.
- The SELECT statement has a HAVING clause in the outside SELECT statement.
- The SELECT statement has a GROUP BY clause in the outside SELECT statement.
- The SELECT statement contains a subquery where the base object of the SELECT statement and the subquery is the same table.

By default, blocks hold 256 KB of data. This number is set by the Data Virtualization Manager server NETWORKBUFFERSIZE parameter. The number of blocks that are used is set by the Data Virtualization Manager server PREFETCH parameter. If Data Virtualization Manager server evaluates a query and determines that it is eligible for block fetch, it begins fetching rows into the prefetch buffers; however, no transmission of data takes place until the first (real) FETCH statement reaches the server.

Note: The maximum number of bytes that is sent for each transmission (for each VTAM SEND) is limited to 32 KB, although Data Virtualization Manager server’s internal prefetch buffers can be larger.

Use block fetch to improve the performance of queries that process many rows in a table.

Note: Using block fetch with a query in which no DESCRIBE (or PREPARE INTO) is performed in advance of fetching rows can degrade performance. Data Virtualization Manager server must internally perform a DESCRIBE to determine the types of data that may be returned.

In addition, depending on the type of isolation level that is used, remember the following considerations:

- If the plan is bound with the Repeatable Read (RR) option and block fetch is used, many more pages can be locked for update than without block fetch, especially if the number of rows that are normally extracted by the query is small.
- If the plan is bound with the Cursor Stability (CS) option and block fetch is used, data changes can take place between the time the data is extracted and the time that it is used by the application.

Enabling block fetch

Using block fetch improves performance of certain types of SQL queries by asynchronously pre-extracting rows (on the server node) ahead of the current row. The pre-extracted rows are then sent back to the requesting node in blocks that contain multiple rows of data.

Procedure

To enable block fetch, use the MODIFY PARM command to add the following parameter to the AVZSIN00 configuration member:

```
"MODIFY PARM NAME(PREFETCH) VALUE(3 BLOCKS)"
```

Parameter	Description	Valid values
PREFETCH	Controls how many blocks of rows should be fetched from DB2. These blocks of rows are used to build the compressed row buffers that are sent to an ODBC application from the server. This value should only be changed if the buffers that are being transmitted from the server to an ODBC client application are not full. Note: This parameter value should be changed only when recommended by technical support.	3 (default)

Configuring DB2 for z/OS Continuous Block Fetch

You can configure support for DB2 for z/OS Continuous Block Fetch (CBF) using DRDA for high performance.

About this task

This task applies only to IBM DB2 for z/OS using DRDA.

Procedure

1. Configure the AVZSIN00 member.
 - a) Set the DRDA configuration for DB2.

- b) In the DRDA Define, the default for the QRBLKSZ parameter is set to 128K. Modify QRBLKSZ if a larger block is needed. Recommendation is to keep the default. As an example, to add 512K:

```
"DEFINE DATABASE TYPE(MEMBER) "
"NAME (DB3A) "
"LOCATION (ZOS3DB3A) "
"DDFSTATUS (ENABLE) "
"DOMAIN (MYHOST) "
"PORT (3740) "
"CCSID (37) "
"QRBLKSZ (524288)
"IDLETIME (160) "
```

2. Add the DRDAMAXBLKEXT parameter. Start with value 8:

```
"MODIFY PARM NAME (DRDAMAXBLKEXT) VALUE (8) "
```

3. In the SQL query, estimate the number of rows in the RESULT SET and use it in the SQL query as follows:

- a) Assuming the SQL query is `SELECT * FROM CBFTABLE`, and there are 50000000 rows.
 b) Append the following to the end: `OPTIMIZE FOR 50000000 ROWS FOR FETCH ONLY`

For example:

```
SELECT * FROM CBFTABLE OPTIMIZE FOR 50000000 ROWS FOR FETCH ONLY
```

4. To verify the functionality, turn on the following TRACE BROWSE parameter:

- a) TRACE DRDA CODEPOINT READ/WRITE/FLOW YES
 b) TRACE DRDA CODEPOINT WRITE BUFFER YES

The trace should look like the following example. The number of corresponding “CodePoint(READ)” equates to the value of DRDAMAXBLKEXT set in the AVZSIN00:

```
18:56:43 0301869847          LEN=02A7,CPT=241B,ELEN=00
18:56:43 0301869848 DSNHLI INTERNAL OPEN-CURSOR - DSNT400I
          SQLCODE = 000, SUCC
18:56:43 0301869849          LEN=0221,CPT=241B,ELEN=00
18:56:43 0301869850 DSNHLI BLOCK FETCH (41490) - RC 0 REASON
          00000000 SQLCODE 0
18:56:44 0301869851          LEN=01A0,CPT=241B,ELEN=00
18:56:44 0301869852          LEN=007C,CPT=241B,ELEN=00
```

MapReduce

This section provides information on MapReduce features for performance enhancement.

You should also refer to the *IBM Data Virtualization Manager for z/OS User's Guide* for additional information on using MapReduce features.

Virtual Parallel Data

Virtual Parallel Data (VPD) allows you to group multiple simultaneous requests against the same data source and run them in parallel, while doing the input and output (I/O) only once. VPD also allows single or multiple requests to run with asymmetrical parallelism, separately tuning the number of I/O threads and the number of client or SQL engine threads.

To use this feature you must provide a VPD group name when submitting request(s). All requests submitted to the same Data Virtualization Manager server with the same group name within a time period will be placed into a VPD group. One or more I/O threads will be started to read the data source and write it to a wrapping buffer. Group members will share the data in the buffer(s), without having to read the data source directly.

A group is created when the first member request arrives. The group is closed either when all members (and all their parallel MRC threads) have joined, or when a timeout has expired. The I/O threads are

started as soon as the group is created, and data begins to flow to the buffer. If the buffer fills before the group is closed, the I/O thread(s) will wait. Once the group is closed and active members begin consuming data, the buffer space is reclaimed and I/O continues.

VPD supports MapReduce Client (MRC), and group members can use different levels of MRC parallelism. For example, a single VPD group might have six members, three members using 5 MRC threads, and the other three using 9 MRC threads. The group will consist of six members and 42 client threads. The number of I/O threads is determined separately. VPD supports a group of a single member, thus supporting asymmetrical parallelism for single requests when using MRC.

VPD is currently supported for the following data sources:

- Adabas files
- Physical sequential data sets on disk, tape, or virtual tape
- Log streams
- VSAM KSDS, RRDS, and ESDS files
- IAM files
- zFS/HFS files

Configuring Virtual Parallel Data

To configure Virtual Parallel Data, specify a group name and appropriate parameters.

Procedure

1. Configure the following parameters in the AVZSIN00 member:

```

/-----/
/* Enable Virtual Parallel Data for asymmetrical parallelism */
/-----/
if DoThis then
do
"MODIFY PARM NAME(VPDGROUPTIMEOUT) VALUE(60)"
"MODIFY PARM NAME(VPDBUFFERSIZE) VALUE(40)"
"MODIFY PARM NAME(VPDTRACEDB) VALUE(NO)"

```

The following table lists the VPD parameters:

Parameter	Description	Valid values
VPDBUFFERSIZE	Specifies the default buffer size, in megabytes above the bar, for a Virtual Parallel Data buffer.	Numeric value in megabytes. Default is 40.
VPDGROUPTIMEOUT	Specifies the maximum time, in seconds, from the time a group is formed until it is closed. Default: 60 seconds	Numeric value in seconds. Default is 60.
VPDTRACEDB	Controls whether Virtual Parallel Data processing will trace debugging messages.	NO Do not trace debugging messages (default). YES Trace debugging messages.

Parameter	Description	Valid values
VPDTRACEREC	<p>Causes Virtual Parallel Data to trace at the record level. <i>(Optional)</i></p> <p>Note: Setting this to YES will produce a large amount of trace output.</p>	<p>NO Do not trace record level messages (default).</p> <p>YES Trace record level messages.</p>

2. Supply the group name.
3. Optional: Specify the number of members in the group. Although optional, this parameter is recommended.

When this parameter is provided, the group is closed as soon as all members have joined. If the number is not provided, the group is not closed until the timeout expires. There is no default.
4. Optional: Specify a timeout value for the group formation.

When the first group member request arrives at the server, the timer is started. If the group remains open when the request expires, it is closed. Any members/threads arriving after the timeout will be placed in a new group. The default is 60 seconds, and can be overridden in the AVZSIN00 file.
5. Optional: Specify the number of I/O threads to use when reading the data source. If this value is not provided, the number of threads is determined as follows:
 - a) If the data source is a tape data set and the number of volumes can be determined, the same number of I/O threads will be started.
 - b) Otherwise, if a Map Reduce thread count is provided in the data map, that number is used.
 - c) Otherwise, if a value is configured for ACIMAPREDUCETASKS in the AVZSIN00 configuration member, that number is used.
 - d) Otherwise, a single I/O thread will be started.

Innovation Access Method (IAM)

Innovation Access Method (IAM) is a VSAM optimization product distributed by Innovation Data Processing. Enable MapReduce for IAM by setting the MAPREDUCEIAMKEYMOD parameter to YES.

MapReduce is implemented by analyzing the file to be retrieved and dividing it up into parts for simultaneous parallel retrieval. For VSAM, this is done by referencing information kept by VSAM about a file. This is supported for key-sequenced data sets (KSDS), entry-sequenced data sets (ESDS), and relative record data set (RRDS) VSAM files. For sequential files, this is done by analyzing information about the extents and volumes of the file. However, for IAM a different approach must be taken because there is no information about the internal structure of an IAM file.

To implement MapReduce for IAM, contact Innovation Data Processing and request module IAMRKTEX. This module will perform the analysis of the internal structure of the IAM file and allow implementation of MapReduce technology. This module will be provided free of charge on request to Innovation Data Processing.

Configuring MapReduce for IAM

Enable MapReduce for IAM by configuring the Data Virtualization Manager server.

Before you begin

The Data Virtualization Manager server must already be installed.

About this task

To enable MapReduce for IAM, you must configure the server configuration file.

Procedure

1. Locate the Data Virtualization Manager configuration member. The server initialization member is shipped in data set member *hlq.SAVZEXEC(AVZSIN00)* and may have been copied to a new data set for customization in the step "Copying target libraries" in the *Installation and Customization Guide*.
2. Locate the parameter MAPREDUCEIAMKEYMOD.
3. Use the **MODIFY PARM** command to change the **MAPREDUCEIAMKEYMOD** parameter value, as follows:

```
"MODIFY PARM NAME(MAPREDUCEIAMKEYMOD) VALUE(YES) "
```

Metadata repository

The metadata repository for MapReduce stores statistics about virtual tables that are used to enhance performance in conjunction with MapReduce and parallelism. This support applies to JDBC Gateway sources, DRDA and IMS data sources, including those accessed via the IBM Federated Server (such as Terradata and Sybase), as well as data sources accessed via direct DRDA support (DB2 LUW and Oracle) provided by the Data Virtualization Manager server. The gathered metadata persists across server restarts.

Populating the metadata repository

You can periodically run the **DRDARange** command to gather metadata repository information about the backend virtual tables. In case of an IMS data source, you can use IMS Direct to gather the metadata repository information.

About this task

You can run the metadata repository command for DRDA either using ISPF panels, a batch job, or studio.

Note: When using MapReduce support, **DRDARange** is required for a relational database management system (RDBMS).

The following restrictions and considerations apply when using this feature:

- Current support does not contain any optimizer enhancements for processing complex queries or joins other than what may be used to enhance MapReduce.
- If a table does not contain enough rows to properly calculate a DRDA Range, then the following error is also returned for this condition:

```
Table <schema>.<table_ name> not eligible for range processing
```

An additional error message can be found in the tracebrowse for this error. For example:

```
22:10:53 Row count 14 too small for range processing
22:10:53 SELECT DRDARANGE('virtual_table.DBLIDX') FOR FETCH ONLY - SQLCODE 0
22:10:53 SQL ENGINE HPO OPEN-CURSOR - SQLCODE 0
22:10:53 SQL ENGINE HPO FETCH - SQLCODE 100
```

Procedure

Run the appropriate command as follows:

- – For DRDA and JDBC Gateway sources, use the SELECT statement at the virtual table level.

```
SELECT DRDARANGE('<TABLE NAME>',MAX_SCAN,'OPTION1','OPTION2',...);
```

Where OPTION1 and OPTION2 can be **SCAN** or **DISPLAY**.

Note:

- It is recommended to use option PARTONLY for partitioned tables. Using this option will force the use of partition boundaries when determining parallelism.

- This command can be executed both in ISPF panels and studio.
- Using a batch job, which you can use to schedule the commands to refresh the statistics on a specified schedule. A sample job is provided in *hlq.SAVZCNTL(AVZRANGE)*. Instructions for required edits to the job are provided in the member.

```
//RANGE EXEC PGM=AVZMAPD,PARM='SSID=AVZS,,MXR=30000000'
//STEPLIB DD DISP=SHR,DSN=loadlibrary
//RPT DD SYSOUT=*
//FMT DD SYSOUT=*,DCB=LRECL=4096
//OUT DD SYSOUT=*
//IN DD *
SELECT DRDARANGE('<TABLE NAME>','MAX_SCAN','OPTION1','OPTION2',...);
```

Work Area or Record Buffer

Work area or record buffer is used when working with the virtual tables.

Work Area Expansion in a Post-Read Exit

To expand the record buffer area passed to a Post-Read exit, the exit should set a return value of 40

Ensure that the record buffer area passed to the Post-Read exit is equal to the size of the virtual table using the Post-Read exit. Expanding the record area beyond the length of the virtual table will result in any data moved into this section of the buffer not being returned by the virtual table.

To expand the work area passed to a Post-Read exit, the exit should set a return code value of 44

Setting either one of the above return codes causes the associated area size to be doubled and then an additional 4K being added to the area.

64-bit buffer

IBM Data Virtualization Manager for z/OS introduces a 64-bit memory management model. This improves the performance and the zIIP utilization by reducing the zIIP spillover due to contention.

To customize the allocation of the 64-bit memory buffer pool, set the following parameters in the IN00 file:

Parameters	Description
SQLENUMSEG	This is the number of 1-megabyte (MB) segments for the main SQL engine buffers. This is the size of the main buffer. The SQLENUMSEG parameter controls the number of segments received on obtaining the 64-bit storage.
SQLENGSGBF	This is the number of SQL engine main buffers to get. The SQLENGSGBF parameter controls the number of pre-allocated 64-bit buffers allocated.
SQLENGSG1M	This parameter controls the number of pre-allocated 1M 64-bit buffers allocated.
SQLENGSG2M	This parameter controls the number of pre-allocated 2M 64-bit buffers allocated.
SQLENGSG3M	This parameter controls the number of pre-allocated 3M 64-bit buffers allocated.
SQLENGSG4M	The parameter controls the number of pre-allocated 4M 64-bit buffers allocated.
SQLENGSG6M	The parameter controls the number of pre-allocated 6M 64-bit buffers allocated.

Parameters	Description
SQLENGNSG16M	The parameter controls the number of pre-allocated 16M 64-bit buffers allocated.
SQLENGNSG64M	The parameter controls the number of pre-allocated 64M 64-bit buffers allocated.
SQLENGNSG128M	The parameter controls the number of pre-allocated 128M 64-bit buffers allocated.
SQLENGNSG256M	The parameter controls the number of pre-allocated 256M 64-bit buffers allocated.
SQLENGNSG512M	The parameter controls the number of pre-allocated 512M 64-bit buffers allocated.

Note: Using the defaults requires a minimum of 16G

Syntax for modifying the parameters in IN00 (default value - less than 56G) are as follows:

```
"MODIFY PARM NAME (SQLENGNUMSEG) VALUE (24) "
"MODIFY PARM NAME (SQLENGNSGBF) VALUE (2304) "
"MODIFY PARM NAME (SQLENGNSG1M) VALUE (16) "
"MODIFY PARM NAME (SQLENGNSG2M) VALUE (512) "
"MODIFY PARM NAME (SQLENGNSG3M) VALUE (24) "
"MODIFY PARM NAME (SQLENGNSG4M) VALUE (2048) "
"MODIFY PARM NAME (SQLENGNSG6M) VALUE (24) "
"MODIFY PARM NAME (SQLENGNSG16M) VALUE (256) "
"MODIFY PARM NAME (SQLENGNSG64M) VALUE (256) "
"MODIFY PARM NAME (SQLENGNSG128M) VALUE (1024) "
"MODIFY PARM NAME (SQLENGNSG256M) VALUE (1024) "
"MODIFY PARM NAME (SQLENGNSG512M) VALUE (1024) "
```

The default value for each parameter is based on the MEMLIMIT for the started task.

If MEMLIMIT is greater than 32G then the default values are:

SQLENGNUMSEG	24	SQL SIZE OF MAIN BUFFERS
SQLENGNSGBF	96	SQL MAIN BUFFERS
SQLENGNSG1M	16	1M BUFFERS
SQLENGNSG2M	256	2M BUFFERS
SQLENGNSG3M	8	3M BUFFERS
SQLENGNSG4M	512	4M BUFFERS
SQLENGNSG6M	4	6M BUFFERS
SQLENGNSG16M	16	16M BUFFERS
SQLENGNSG64M	8	64M BUFFERS
SQLENGNSG128M	8	128M BUFFERS
SQLENGNSG256M	4	256M BUFFERS
SQLENGNSG512M	2	512M BUFFERS

If MEMLIMIT is greater than 16G then the default values are:

SQLENGNUMSEG	24	SQL SIZE OF MAIN BUFFERS
SQLENGNSGBF	32	SQL MAIN BUFFERS
SQLENGNSG1M	8	1M BUFFERS
SQLENGNSG2M	128	2M BUFFERS
SQLENGNSG3M	8	3M BUFFERS
SQLENGNSG4M	256	4M BUFFERS
SQLENGNSG6M	4	6M BUFFERS
SQLENGNSG16M	8	16M BUFFERS
SQLENGNSG64M	4	64M BUFFERS
SQLENGNSG128M	4	128M BUFFERS
SQLENGNSG256M	0	256M BUFFERS
SQLENGNSG512M	0	512M BUFFERS

If MEMLIMIT is greater than 56G and less than 96G then the default values are:

SQLENGNUMSEG	24	SQL SIZE OF MAIN BUFFERS
--------------	----	--------------------------

SQLENGNSGBF	512	SQL MAIN BUFFERS
SQLENGNSG1M	16	1M BUFFERS
SQLENGNSG2M	2048	2M BUFFERS
SQLENGNSG3M	16	3M BUFFERS
SQLENGNSG4M	2048	4M BUFFERS
SQLENGNSG6M	8	6M BUFFERS
SQLENGNSG16M	128	16M BUFFERS
SQLENGNSG64M	32	64M BUFFERS
SQLENGNSG128M	16	128M BUFFERS
SQLENGNSG256M	8	256M BUFFERS
SQLENGNSG512M	8	512M BUFFERS

If MEMLIMIT is greater than 96G then the default values are:

SQLENNUMSEG	24	SQL SIZE OF MAIN BUFFERS
SQLENGNSGBF	256	SQL MAIN BUFFERS
SQLENGNSG1M	64	1M BUFFERS
SQLENGNSG2M	3072	2M BUFFERS
SQLENGNSG3M	32	3M BUFFERS
SQLENGNSG4M	4096	4M BUFFERS
SQLENGNSG6M	8	6M BUFFERS
SQLENGNSG16M	128	16M BUFFERS
SQLENGNSG64M	64	64M BUFFERS
SQLENGNSG128M	32	128M BUFFERS
SQLENGNSG256M	16	256M BUFFERS
SQLENGNSG512M	8	512M BUFFERS

The SQL function BUFFERINFO() is used to monitor the usage of the buffer pools.

For more information, see BUFFERINFO section.

Chapter 14. Logging and tracing server information

System information that is unique to the Data Virtualization Manager server can be recorded in Server Trace log file(s), System Management Facility (SMF) data set records, and in Data Virtualization log (DB2) tables.

Use Server Trace to trace and log server events, and to help identify and troubleshoot Data Virtualization Manager server issues.

Use the System Management Facility (SMF) to record system resource usage information in SMF data sets.

Use DB2 logging to write out the total z/OS resource usage information into an intervals table for a specified time interval.

The information that you collect can be used for:

- Billing users
- Reporting reliability
- Analyzing the configuration
- Scheduling jobs
- Summarizing direct access volume activity
- Evaluating data set activity
- Profiling system resource use
- Maintaining system security

You can choose to use any combination of logging features. For example, SMF logging can be used together with Server Trace logging, or separately. When used together, only the SMF Record Subtypes 01, 02, 06, 09, 10, 18 and 19 records are available for logging into DB2 tables. In most cases, the fields that are recorded are identical. However, some fields may not be available in both SMF and Server Trace logs.

Server Trace

Use Server Trace to trace and log server events, and to help identify and troubleshoot Data Virtualization Manager server issues.

By default, for each event that occurs on the Data Virtualization Manager server, an entry is created and stored in the trace log.

You can view the trace log from the studio or from the Data Virtualization Manager server ISPF panels.

- To view the trace log from the Studio, open the Server Trace view from the studio **Window** menu, select **Show View**, and then select **Server Trace**.
- To view the trace log from the ISPF application, select option **B Server Trace > Server Trace Facility> SIS SSID:AVZS**.

Note: To view the Server Trace information for a different server, replace AVZS with the appropriate subsystem name.

As a session runs, entries are created for the following types of events:

- SQL operations
- IMS calls
- CICS calls
- Communication events
- Thread attach and detach events

- RPC events
- Message events
- Errors and abends

When the Data Virtualization Manager server runs a SQL statement, multiple entries are created in the trace log and in the client log. Each log records the series of events from a different perspective.

For example:

A client that runs a SQL statement could record the following entries:

- SEND event
- RECEIVE event
- SQL event (The results are returned.)

While the Server Trace log records the following entries:

- RECEIVE event (Matches the client SEND event.)
- SQL event (The SQL statement that is sent to DB2®.)
- SEND event (Matches the client RECEIVE event.)

If you were to view a combined log of the SQL statement execution, it would appear in the order each event occurred. For example:

- SEND event (Client side.)
- RECEIVE event (Server side.)
- SQL event (Server side.)
- SEND event (Server side.)
- RECEIVE event (Client side.)
- SQL event (Client side.)

The Trace Browse consists of a large block of virtual storage that is used to back up active trace browse data. This block of virtual storage is subdivided into a status area, a configurable number of event blocks, and a series of vector tables. The entries are initially added to a buffer that is maintained in virtual storage. In general, the buffer can accommodate the complete history of all client and server processing for several days. The entries are written to disk (a VSAM data set) every "n" seconds, as set by the parameter BROWSEINTERVAL (SERVER TRACE CHECKPOINT INTERVAL).

The Data Virtualization Manager server supports multiple trace data set Archives. Using hierarchical storage management, you can maintain an unlimited history of entries.

You can configure the Instrumentation Server to route entries from multiple instances of the Data Virtualization Manager server in a sysplex, to a single repository; giving you with a global view of all activity in a single ISPF panel.

The SQL Trace program provides information about the SQL statements that applications issue. When you select the active session, SQL Trace uses the connection ID as criteria to obtain and display SQL entries from the trace log.

When the TRACEEXTERNTRACEDATA parameter is set to YES, the Trace Data from the driver connection is sent to the server in the Servers Trace Browse area.

Displaying and navigating log entries

Use the Server Trace panel to view, navigate, and manage the log entries that display.

About this task

By default, the Server Trace panel displays all log entries. To view a subset of the log entries, you can filter on the results, use labels, and create a profile. If the Data Virtualization Manager configuration is running on a zIIP server, entries that are related to work that runs on the zIIP server are displayed in pink. If

the server is running on a zAAP server, entries that are related to work that runs on the zAAP server are displayed in turquoise.

Procedure

1. From the Primary Option panel, enter B on the Option line.

The Server Trace panel displays the most recent entries, which are at the end of the list. By default, the time, host name, and description of the event are displayed.

2. On the Server Trace panel, you can navigate through the trace messages in the following ways:

- Use the UP, DOWN, RIGHT, and LEFT scroll commands (or their PF key equivalents) to navigate this panel.
- Use the MAX or M scroll operand to scroll the maximum amount in any direction.
- If you are at the beginning or end of the trace list (and it is full), press **ENTER** to scroll the list down. Messages are removed from the beginning and added to the end.

3. Optional: Perform any of the following steps:

- To refresh the list, press Enter.
- If you reposition the display, to see the most recent entries, issue the DOWN MAX command and then press **Enter**.
- To display a different set of columns, type D on the command line, followed by the names of the columns to display.

Server Trace panel columns

Use the DISPLAY command to display specific columns on the Server Trace panel.

Column	Description
ACTION	Displays one of the following: <ul style="list-style-type: none"> • ACC (accept) • REJ (reject) • NOA (no action)
ADDRESS	The location in memory of the actual record.
ADDRJOB	The location in memory of the current record in the JOBNAME vector.
ADDRUSR	The location in memory of the current record in the USERID vector.
APMRC	The APPC/MVS return code.
ASID	The address space ID of the user who created the current record.
AVZFLAGS	The bits that are set by the routines that created the trace.
CLOCK	The timestamp of when the record was created.
CNID	The identifier assigned to each thread that is created. This column supports the following Hexa decimal characters: ABCDEFabcdef0123456789.
CODE	The lowest level return code for each event.

Table 22. Server Trace panel columns (continued)

Column	Description
COLOR	The color assigned to a Server Trace message.
COUNT	The number of rules that processed the event.
CPUTIME	The CPU time used by a particular thread. The format depends on how much CPU time the user has used: <ul style="list-style-type: none"> • Fewer than 1000 seconds: <i>nnn.nnn</i>s • Between 1000 seconds and 100 hours: <i>hh:mm:ss</i> • 100 hours or more: <i>hhh: mm</i>
CVID	The conversation ID that LU 6.2 assigns when a conversation starts. This column supports the following Hexa decimal characters: ABCDEFabcdef0123456789.
DATE	The date when the message was created, in <i>dd:mm:yy</i> format.
ELAPSED	The total time that the current event used, in decimal microseconds (millionths of a second). To derive the total, the STCK (clock store) value that is taken at the beginning of processing is subtracted from the STCK value that is taken at the end of processing.
EVENT	The type of event that created the entry.
GTRIDTKN	The global transaction.
HOSTNAME	The TCP/IP host name or LU 6.2 host name. This column supports the NATIONAL characters (#, \$, and @).
HOSTX	The TCP/IP host name extended or the LU6.2 host name/mode.
IPADDR	The IP address, which is the TCP/IP source or target that is associated with the entry.
IPV6ADDR	Internet Protocol Version 6 address.
JOBNAME	The name of the job or address space that created the entry. This column supports the NATIONAL characters (#, \$, and @).
LENGTH	The length of the text section of the message.
LUNAME	The LU 6.2 source or target that is associated with the message.
MSGNO	The message number. When data collection begins, message 1 is the first message collected; message 2 is the second message; and so on. When there is no more room in the message area, the oldest message is discarded to make room for a new message. Therefore, the first message in the list might not be message 1.

Table 22. Server Trace panel columns (continued)

Column	Description
MSGORIGN	The SIS/XCF (Instrumentation Server XCF) member name where the message originated. A message origin has the following format: <i>SYSIDALS_SSIDSISID</i> where <ul style="list-style-type: none"> • <i>SYSID</i> is the system ID. • <i>ALS_SSID</i> is the Data Virtualization Manager subsystem ID. • <i>SISID</i> is the Instrumentation Server ID. • This column supports the NATIONAL characters (#, \$, and @).
NODENAME	The name of the communications node that is associated with the message. The format of each entry depends on the communication link type.
OERC	The TCP/IP return code of the OE socket.
PATHID	IUCV path ID
PROCESS	OE Process ID, if task is dubbed
RC	The highest level return code for the message.
REASON	The second-level return code for the message.
RULESET	The name of the first RULESET.RULE that processed an event on NONE.NONE.
SECONDS	The first four bytes of the binary timestamp, which indicates when the message was created.
SESSION	The communications session that is associated with the message. The format of each entry depends on the type of communication link.
SOCKET	The socket number that is associated with the message. This column applies only to TCP/IP events.
SQLRC	The SQL return code.
SSID	The subsystem ID, for example, Db2, IMS, or CICS. This column supports the NATIONAL characters (#, \$, and @).
TCBADDR	The TCB (task control block) address field that contains the address of the TCB that created the message. This column supports the following Hexa decimal characters: ABCDEFabcdef0123456789.
TERMNAME	The name of the terminal that is associated with the event.
TIME	The time that the message was created, in <i>hh:mm:ss</i> format.
TIMEX	The time that the message was created, calculated to the microsecond, in <i>hh:mm:ss.uuuuuu</i> format.
TRACE1	The trace data that is specific to the message.

Table 22. Server Trace panel columns (continued)

Column	Description
USERID	The security product user ID that best identifies the message. This column supports the NATIONAL characters (#, \$, and @).
VCID	The unique virtual connection ID. This column supports the following Hexa decimal characters: ABCDEFabcdef0123456789.
VERSION	The version of the product that generated the message.
VTAMRC	The VTAM® return code.
XIDTOKEN	The XA token ID.

Displaying information about SQL entries

Use the SDINFO, SDTRAC, and SDDATA commands to display information about the SQL that is associated with a selected entry.

Procedure

1. In the **Command** field, type one of the following commands:

Command	Description
SDINFO	Invokes the SQL Explain program that presents explanatory text for the SQLCODE that is associated with the specified entry.
SDTRAC	Invokes the SQL Trace program that traces all SQL events for the connection ID that is associated with the specified entry.
SDDATA	Invokes the SQL Data program that presents a formatted SQL Communications Area (SQLCA) control block for the specified entry.

2. Position the cursor on the entry for which you want information, and press Enter.

Displaying information about the Data Virtualization Manager server

The TCBADDRESS column of the server log specifies whether the Data Virtualization Manager configuration is running on a zAAP or zIIP server or in SRB mode.

Procedure

On the command line, enter `d tcb` to display the TCBADDRESS column.

The leftmost hexadecimal digit in the high-order byte of the TCB address specifies the mode:

- 20 indicates that the Data Virtualization Manager server is running on a zAAP server.
- 80 indicates that the Data Virtualization Manager server is running in SRB mode.
- C0 indicates that the Data Virtualization Manager server is running on a zIIP server in SRB mode.
- D0 indicates that the Data Virtualization Manager server is running on a zIIP server in SRB mode and the zIIP server is running at a different speed than a CP (Turbo mode).

Locating entries in the server log

Use the LOCATE command to position the display at an entry that contains a specific date, time, message number, or label.

Procedure

1. From the **Primary Option** panel, enter B on the Option line.
2. On the **Server Trace** panel, use the DISPLAY command to display the appropriate column.
For example, enter D date.
3. Enter the LOCATE command, followed by the criteria.
For example, to find an entry that has the time 21:51:58, enter L 21:51:58.

To specify criteria, use the following formats:

Criteria	Format
Time	One of the following: <ul style="list-style-type: none">• <i>hh</i>• <i>hh:mm</i>• <i>hh:mm:ss</i>
Date	One of the following: <ul style="list-style-type: none">• <i>dmmm</i>, single-digit date and current month• <i>ddmmm</i>, date and current month• <i>ddmmyy</i>, date, month, and 2-digit year• <i>ddmmyyyy</i>, date, month, and 4-digit year
Message number	The specific message number
Label	The previously specified label that was added to an entry

Filtering log entries

To view a subset of the log entries, create a profile. In the profile, you specify the criteria to use to select entries to display, and you select the specific events to display. The profile that you create affects only how you view log entries. Other users can create their own profiles.

Procedure

1. From the **Primary Option** panel, enter B on the Option line.
2. On the **Server Trace** panel, type PROFILE (with no operands) on the command line.
3. On the **Trace Browse Profile** panel, enter criteria in one or more of the following fields. If you enter multiple criteria, the values are joined with the logical AND operator. If you enter multiple values for a criterion, the values are joined with the logical OR operator. You can enter up to four values for each criterion.

Criterion	Description
JOBNAME	Limits entries to those that contain the specified value in the JOBNAME column. You can use an asterisk (*) as a wildcard character.

<i>Table 23. Profile filtering criteria (continued)</i>	
Criterion	Description
USERID	Limits entries to those that contain the specified value in the USERID column. You can use an asterisk (*) as a wildcard character.
CONNECT	Limits entries to those that contain the specified value in the CONNECT column.
VCID	Limits entries to those that contain the specified value in the VCID (virtual connection ID) column.
HOST NAME	Limits entries to those that contain the specified value in the HOST NAME column. You can use an asterisk (*) as a wildcard character.
TCB	Limits entries to those that contain the specified value in the TCB column.
SSID	Limits entries to those that contain the specified value in the SSID column. You can use an asterisk (*) as a wildcard character.
XIDTOKEN	Limits entries to those that contain the specified value in the XIDTOKEN (XA token ID) column.
GTRIDTKN	Limits entries to those that contain a matching GTRIDTKN (global transaction ID).
CONVTKN	Limits entries to those that contain a matching CONVTKN (conversation token ID).
MSGORIGIN	Limits entries to those that contain a matching MSGORIGIN (message origin). You can use an asterisk (*) as a wildcard character. Use the following format to enter the values: <i>SYSIDALS_SSIDSSID</i> where <ul style="list-style-type: none"> • <i>SYSID</i> is the system ID. • <i>ALS_SSID</i> is the subsystem ID. • <i>SSID</i> is the Instrumentation Server ID.

4. Enter Y or N to include or exclude the following specific types of events from the result set:

<i>Table 24. Profile filtering events</i>	
Event	Description
ABN	Abend entries.
ADA	ADABAS entries.
APM	APPC/MVS entries.
ATH	Authorization entries.
BKR	ACI broker entries.
CMD	Command entries.
CPG	C program entries.

<i>Table 24. Profile filtering events (continued)</i>	
Event	Description
DET	Detach entries.
DIS	Disable entries.
ECI	CICS EXCI entries.
ENA	Enable entries.
EXC	Exception entries.
FIL	File entries.
GLV	Global variable entries.
IMS	IMS entries.
MFL	MicroFlow (MFL) entries.
MQS	MQ message entries.
OTC	IBM OE sockets TCP/IP entries.
OTM	IMS/OTMA entries.
PUB	IBM Data Virtualization Manager for z/OS Streams entries.
RPC	RPC entries.
RRS	RRS entries.
RSF	RRSAF entries.
SIS	Instrumentation Server entries.
SQL	SQL entries.
SOM	Security Optimization Management entries.
SQM	SQM entries.
SSL	SSL entries.
STG	Storage alteration entries.
STR	System trace entries.
TOD	Time-of-day entries.
TSO	TSO entries.
TXT	Product initialization, termination, and general execution entries.
TYP	TYP entries.
WLM	Workload Manager entries.
WWW	WWW entries.
XCF	Coupling Facility entries.
XTX	Extended text entries.
ZSR	Services entries.
6.2	LLU 6.2 entries.

5. Press **Enter** to save the profile.

Labeling and locating specific log entries

To quickly locate significant entries in the server log, replace the message number of an entry with a label.

About this task

After you add labels to entries the trace log, use the LOCATE command to find the entries.

Procedure

1. From the **Primary Option** panel, enter B on the Option line.
2. On the **Server Trace** panel, use the DISPLAY command to display the relevant columns.
For example, enter DISPLAY msgno date.
3. When you locate the entry to which you want to add a label, edit the MSGNO column and enter a label that consists of a period and up to seven alphabetic characters.
For example, enter . POINTA.
4. Enter the LOCATE command, followed by the criteria.
To specify criteria, use the following formats:

Criteria	Format
Time	One of the following: <ul style="list-style-type: none"> • <i>hh</i> • <i>hh:mm</i> • <i>hh:mm:ss</i>
Date	One of the following: <ul style="list-style-type: none"> • <i>dmmm</i>, single-digit date and current month • <i>ddmmm</i>, date and current month • <i>ddmmyy</i>, date, month, and 2-digit year • <i>ddmmyyyy</i>, date, month, and 4-digit year
Message number	The specific message number
Label	The previously specified label that was added to an entry

Finding character strings in the server log

Use the FIND and RFIND commands to find a specific character string in the server log. You can find a string in a specific column or in a range of columns.

Procedure

1. From the **Primary Option** panel, enter B on the Option line.
2. On the **Server Trace** panel, enter the FIND command to find the character string.
To search for a string in the USERID, EVENT, or SSID column, use the following syntax:

```
FIND column-name string prefix direction
```

Where

- *column-name* is USERID, EVENT, or SSID.
- *string* is the search string.

- *prefix* specifies that the search string is generic and specifies only the prefix characters. Specify this argument when you search EVENT or SSID columns.
- *direction* specifies the next match to find. Specify FIRST (default), LAST, PREV, or NEXT.

To search for the string in a range of columns, use the following syntax:

```
FIND TEXT string direction start-column end-column msgno
```

Where

- TEXT is an optional keyword that indicates that you are searching only the text of the entries.
- *string* is the search string. If the search string contains blank spaces or is identical to a FIND keyword, enclose the string in quotation marks. Enter an asterisk (*) to use the search string from the previous FIND command.
- *direction* specifies the next match to find. Specify FIRST (default), LAST, PREV, or NEXT.
- *start-column* specifies the number of the first column for the search.
- *end-column* specifies the number of the last column for the search.
- *msgno* is the maximum number of entries to search. The default is 5000.

The following FIND command searches for the string AVS1234W from the first message, beginning at column 10 and ending at column 30, for 10,000 messages:

```
F 'AVS1234W XYZ' 10 30 10000
```

3. Optional: Enter RFIND to repeat the previous FIND command.

Capturing the entries from the server trace

Use the P, PP, and SS commands to print server log entries to the ISPF list data set.

About this task

Each entry that you print contains the same columns that are displayed in the **Server Trace** panel and includes the entire contents of the text field. If the text field exceeds one line, the printed entry wraps to include three additional lines. Make sure that the ISPF list data set has enough space to hold the printed entries. The SS command requires more space than the PP command. The SS command prints 1 - 100 entries as they appear in the trace log, followed by the zoomed formatting for each entry, followed by the next 1 - 100 entries.

Procedure

On the **Server Trace** panel, to print log entries, perform one of the following steps:

- To print a single entry, enter P in the MESSAGENUM column.
- To print the summary information for a range of entries, enter PP in the MESSAGENUM column on the first and last entry in the range.
- To print the summary and detailed information, enter S in the MESSAGENUM column.
- To print the summary and detailed information for a range of entries, enter SS in the MESSAGENUM column on the first and last entry in the range.

Archiving the Server Trace

To save Server Trace information that is in virtual memory to a data set, enable Server Trace archiving.

About this task

Note: Do not enable the Trace Browse archive if you are using the Instrumentation Server, which handles archiving.

Procedure

To enable and configure Server Trace archiving, use the **MODIFY PARM** command and set the following parameters in the Data Virtualization Manager configuration member, AVZSIN00:

```
"MODIFY PARM NAME(ARCHIVEDSNPREFIX) VALUE(HLQ.ARCHIVE)"
"MODIFY PARM NAME(ARCHIVESTORCLASS) VALUE(SYSSMS)"
"MODIFY PARM NAME(BROWSEARCHIVE) VALUE(AUTO)"
"MODIFY PARM NAME(BROWSEARCHIVECOUNT) VALUE(30000)"
"MODIFY PARM NAME(BROWSEARCHIVECUSHION) VALUE(15000)"
```

The following table lists the Server Trace archiving parameters:

Parameter	Description	Valid values
ARCHIVEDSNPREFIX	Defines the high-level qualifier, which the subsystem uses to construct data sets names for archive files. The value ".Dyyyyddd.Thhmmss" is appended to the qualifier, where yyyyddd is the Julian date, and hhmmss is the time of day.	'NULL'
ARCHIVESTORCLASS	Defines the STORCLASS operand value that is used to define linear clusters for archive data sets. When not set, STORCLASS is not specified when the linear data sets are allocated.	'NULL'
BROWSEARCHIVE	<p>Controls whether the product produces archives of the wrap-around trace and how the archival procedure is inaugurated. Possible values are NONE, AUTO, and MESSAGE.</p> <p>When set to NONE, archival of the trace is not supported and only user-requested archive extracts are supported. Explicitly requested extract archives are not considered to be "backup" type archives.</p> <p>When set to AUTO, archival is triggered by automatically generating an ARCHIVE BACKUP command. When set to MESSAGE, a message is generated when reaching should be performed. The generation of the ARCHIVE BACKUP command is not performed automatically.</p>	AUTO MESSAGE NONE (default)

Parameter	Description	Valid values
BROWSEARCHIVECOUNT	<p>Specifies the number of messages to write for each automated archival operation. The archival process begins when the BROWSEARCHIVECOUNT value is reached. This value should be 30% of the BROWSEMAX parameter to allow the archival process to complete before the BROWSEMAX value is reached and trace browse records would be overwritten.</p>	30000
BROWSEARCHIVECUSHION	<p>Specifies the number of messages that are used as a threshold for automated triggering of an archive event and as a guard against archiving overwritten messages. An archive event is scheduled for each group of BROWSEARCHIVECOUNT messages. However, scheduling is deferred until BROWSEARCHIVECUSHION additional messages have been logged.</p> <p>This cushion is required because some messages are updated in place, and allow the system to get beyond the ACTIVE message range before actually copying the messages to a backup. The cushion value is also used if a backup is requested and overlay of previously un-backed-up message is in progress or imminent. The system begins the archive with the next unarchived message, when possible. But if overlay is imminent or already in-progress, this number of messages is skipped in order to ensure that these overlaid messages are not copied.</p> <p>Note: The default setting for BROWSEARCHIVECUSHION is 50% of the BROWSEARCHIVECOUNT.</p>	15000
BROWSEMAX	<p>Specifies the maximum number of messages a trace holds. The maximum value allowed is 1910786.</p>	100,000

Setting a server trace profile value for a subsystem

Set profile values for ISPF server trace for every subsystem.

In the ISPF server trace, you can set profile values in order to limit the number of records being viewed. The profile values are now specific to the subsystem (SSID) server trace that you are currently viewing and not a global variable that applies to the ISPF session.

System Management Facility logging

Using the System Management Facility (SMF), you can record system resource usage information in SMF data sets.

To enable SMF support during product customization, provide a value for the **SMF record number** product parameter. SMF logging can be used together with IBM Data Virtualization Manager for z/OS logging, or separately.

The following sections include SMF record subtype information.

Enabling SMF logging

Enable the SMF logging feature to collect and record information that is used to evaluate system usage. It can be used together with IBM Data Virtualization Manager for z/OS logging, or separately.

Procedure

Use the `MODIFY PARM` command to set the following parameter in the `hlq.SAVZEXEC(AVZSIN00)` member `IN00`:

```
"MODIFY PARM NAME(SMFNUMBER) VALUE(nnn)"
```

Where `SMFNUMBER` controls SMF recording. To enable SMF recording, set the value to the appropriate number for each SMF record subtype. If the value is set to zero, no logging takes place.

Results

After you enable SMF record subtypes, you can configure SMF parameter settings.

The following table lists and describes SMF parameters.

Parameter	Description	Valid values
ADABASDBIDSMF	Causes one SMF record to be written per DBID accessed at the end of each session. The records contain command usage statistics. SMF Subtype 17: ADABAS Command by DBID Records	YES NO This is the default value.
CHECKSTORAGEINTERV	Controls how often (in seconds) statistics for allocated storage are gathered from the Data Virtualization Manager server. A value of zero turns off this function.	900 This is the default value.

Table 25. SMF Parameters (continued)

Parameter	Description	Valid values
LOGERRORSSMF	Controls whether DB2 SQL error information should be written to SMF. When set to YES, this value generates SMF subtype 13 records. SMF Subtype 13: DB2 SQL Errors	YES NO This is the default value.
LOGINTERVALS	Controls whether session interval information is logged. Session interval information is logged by inserting rows in to a DB2 table. One row is inserted for each session at the end of each recording interval and at session termination time.	YES This is the default value. NO
LOGINTERVALSSMF	Controls whether session interval information should be written to SMF.	YES This is the default value. NO
LOGSESSIONINTVALSMF	Controls whether interval type records are written to SMF. Interval records can also be written to the session log. SMF Subtype 02: Interval Summary Records	YES This is the default value. NO
LOGSTORAGEESMF	Controls whether storage usage information should be written to SMF. Storage usage information can also be written to a DB2 table. SMF Subtype 09: Storage Interval Summary Records	YES NO This is the default value.
LOGWSTORTM	Enables logging Services information for Real-Time Monitoring. SMF Subtype 18: Services Records	YES NO This is the default value.
MONITORAPPC/MVS	Specifies whether to monitor APPC/MVS conversations. SMF Subtype 10: APPC/MVS Interval Summary Records	YES This is the default value. NO

Table 25. SMF Parameters (continued)

Parameter	Description	Valid values
MONRESPONSETIME	<p>Controls whether to monitor the client response time if application names are defined in the initialization EXEC by using the DEFINE RTMONAPP statement.</p> <p>When set to YES, monitoring of the client response time occurs if application names are defined.</p> <p>SMF Subtype 14: Client Response Time Records</p>	<p>YES NO</p> <p>This is the default value.</p>
PUBLISHINTERVALSMF	<p>Specifies whether to write SMF records for the Streams long running tasks.</p> <p>SMF Subtype 19: Streams Records</p>	<p>YES NO</p> <p>This is the default value.</p>
RECORDINGINTERVAL	<p>Controls how often interval summary and per-client SMF and/or SQL records are created. These records show what resources were used during the current recording interval. The interval value is specified in seconds and should be a factor of one hour. The value should divide evenly into 3600.</p>	<p>900 (default)</p>
SMFNUMBER	<p>Controls SMF recording. To enable SMF recording, set SMFNUMBER to the desired number. If set to zero, no logging takes place. This number must be a unique SMF record.</p>	<p>0 (default) no logging</p>
SMFRULEDISABLE	<p>Indicates whether this type of SMF record should be written.</p> <p>SMF Subtype 03: SEF Rule Disablement Records</p>	<p>YES NO</p> <p>This is the default value.</p>

Table 25. SMF Parameters (continued)

Parameter	Description	Valid values
SMFTRANSACT	<p>Controls the creation of SMF transaction records. Possible values are YES and NO.</p> <p>When set to YES, an SMF record is created for each inbound client request.</p> <p>When set to NO, no per-transaction records are created.</p> <p>SMF Subtype 06: Per Transaction SMF Records</p>	<p>YES NO</p> <p>This is the default value.</p>
WSSMF	<p>Causes SMF records to be written for each Services transaction. WSSMF is only required if ends of session records are desired. If you only want summary records, this parameter should be set to NO.</p> <p>SMF Subtype 18: Services Records</p>	<p>YES NO</p> <p>This is the default value.</p>
WSSMFSUMMARY	<p>Causes summary SMF records to be written for Services transactions. Interval recordings of Services are summarized at the highest level (a single record per interval). Records have SM18RCTY = 'I'.</p> <p>SMF Subtype 18: Services Records</p>	<p>YES NO</p> <p>This is the default value.</p>
WSSMFSUMMARYOPER	<p>Causes Operation summary SMF records to be written for Services transactions. Enables interval recording of Services summarized at the operation level. Records have SM18RCTY = 'O'.</p> <p>SMF Subtype 18: Services Records</p>	<p>YES NO</p> <p>This is the default value.</p>

Table 25. SMF Parameters (continued)

Parameter	Description	Valid values
WSSMFSUMMARYVDIR	Causes Virtual Directory summary SMF records to be written for Services transactions. Enables interval recording of Services summarized at the Virtual Directory level. Records have SM18RCTY = 'V'. SMF Subtype 18: Services Records	YES NO This is the default value.
WSSMFSUMMARYWS	Causes Virtual Directory summary SMF records to be written for Services transactions. Enables interval recording of Services summarized at the Web Service level. Records have SM18RCTY= 'W'. SMF Subtype 18: Services Records	YES NO This is the default value.

Record Subtype 01: Client System

This record is used to collect session and connection information about the client system.

About this task

The Subtype 01 record is used to record:

- End of session records, which indicate resource usage per client connection.
- Connection usage for a specific connection for the INTERVAL RECORDING PERIOD set by the RECORDINGINTERVAL parameter.

The information is written at the end of every connection.

Use the SM01RCTY field in the SMF record to set the record type to one of the following values:

- S: The final end-of-session record.
- F: The final interval record that shows the usage of CPU time for that specified interval.
- I: The interim interval record.

If you are only interested in end-of-session records for charge back situations, always check the SM01RCTY field for each 01 record to ensure that it is not an interval record; otherwise, incorrect calculations could be interpreted.

A sample SAS program is provided that can be used to print these SMF fields. The program is located in AVZSFDV1 in the hlq.SAVZCNTL member.

The following table lists the parameters used to configure the Subtype 01 record:

Table 26. Subtype 01 Record Information

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record sub type
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SMO1CLNA	CL16	Client system name
53	SMO1CLTY	CL8	Client type (protocol type)
61	SMO1CLUS	CL8	Client user id
69	SMO1CLCP	D	Client CPU time
77	SMO1SMID	CL4	Host system SMFID
81	SMO1ODVR	XL1	ODBC version code
82	SMO1ODRL	XL1	ODBC version code
83	SMO1ODMD	XL2	ODBC modification code (MM/DD)
85	SMO1ODYR	AL2	ODBC year value
87	SMO1ODMN	AL1	ODBC month value
88	SMO1ODDD	AL1	ODBC day value
89	SMO1CNID	XL4	Connection ID
93	SMO1LGTM	XL8	Client logon time (GMT TOD)
105	SMO1ELTM	XL8	Client elapsed time (TOD) in microseconds
113	SMO1WRTO	XL8	Raw total bytes written
121	SMO1TOTM	XL4	Client total response time in seconds
125	SMO1HOTM	XL4	Client host response time in seconds
129	SMO1ABCD	XL2	Client system abend code
131	SMO1USAB	XL2	Client user abend code
133	SMO1ENZQ	D	Enclave zIIP qualified CPU time
141	SMO1ADLT	XL8	Adjusted client logon time

Table 26. Subtype 01 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
145	SMO1IPAD	XL4	TCP/IP client IP address
153	SMO1ORUS	CL8	Original user ID value
161	SMO1PLAN	CL8	DB2 plan name
169	SMO1SSNA	CL4	DB2 subsystem/group name
173	SMO1DBCP	CL8	DB2 CPU time
181	SMO1NTCP	CL8	Network CPU time
189	SMO1OHCP	CL8	Other CPU time
197	SMO1RXCP	CL8	REXX CPU time
205	SMO1RPCP	CL8	RPC CPU time
213	SMO1INST	CL8	Adjusted interval start time
221	SMO1SQCN	F	SQL count
225	SMO1SSAC	CL4	Group attachment member name
229	SMO1ENCP	CL8	Enclave CPU time
238	SMO1RCTY	C	Record type <ul style="list-style-type: none"> • C'F'=Final interval record type • C'I'=Interim interval record type • C'S'=Session record type
239	SMO1APLN	H	Application name length
241	SMO1APNA	CL18	Application name from client
261	SMO1ENZI	D	Enclave zIIP CPU time
269	SMO1ENZC	D	Enclave zIIP time on CP
277	SMO1SLCP	D	SSL CPU time
291	SMO1USLN	H	User parameter length
293	SMO1USPA	CL100	User parameter from client
393	SMO1PDSS	CL4	Product subsystem name
397	SMO1CLWT	XL8	Client WAIT time in microseconds
405	SMO1CLRC	F	Client READ DATA COUNT
409	SMO1LNID	CL100	Client LAN (network) user ID
509	SMO1HONA	CL16	Host name (CMLI)
525	SMO1ADCT	F	ADABAS command count
533	SMO1SRCP	D	SRB CPU time
541	SMO1IPV6	XL16	TCP/IP CLIENT IPV6 ADDRESS
557	SMO1LCPO	H	LOCAL_PORT NUMBER

Record Subtype 02: Internal Summary

This record is used to collect session information for all users who are connected during a specific interval and the information is written at the end of each interval. All the resources that are used by all connections during that interval are recorded using this record.

About this task

The interval in which Subtype 02 records are written is determined by the RECORDINGINTERVAL parameter.

A sample SAS program is provided that can be used to print the fields in Subtype 02 records. The program is located in the SMFSDB02 member of the hlq.SAVZEXEC(AVZSIN00) data set.

Interval summary records are automatically written if the LOGINTERVALS parameter is set to YES in the hlq.SAVZEXEC(AVZSIN00) member. You must have LOGINTERVALS enabled in order to also record Interval records into SMF.

Procedure

To log interval records to the logging tables but not log interval information to SMF, in the hlq.SAVZEXEC(AVZSIN00) member, set the LOGINTERVALS parameter as follows:

```
"MODIFY PARM NAME(LOGLSESSIONINTVALSMF) VALUE(NO)"
```

Where LOGLSESSIONINTVALSMF controls whether interval type records are written to SMF. Interval records can also be written to the session log.

Results

The following table lists the parameters used to configure the Subtype 02 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none">• X'10' = MVS/ESA 4• X'08' = MVS/XA• X'04' = MVS/ESA• X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SMO2SMID	CL4	Host system (SMF ID)
41	SMO2PDSS	CL4	Product subsystem name
45	SMO2RCTY	C	Record type: C'I'=INTERVAL SUMMARY Record type

Table 27. Subtype 02 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
53	SMO2INST	CL8	Interval start time
61	SMO2SQCN	F	SQL COUNT
69	SMO2ENCP	CL8	Enclave CPU time
77	SMO2CLCP	CL8	Client task CPU time
85	SMO2DBCP	CL8	DB2 CPU time
93	SMO2NTCP	CL8	Network CPU time
101	SMO2OHCP	CL8	OTHER CPU time
109	SMO2RXCP	CL8	REXX CPU time
117	SMO2RPCP	CL8	RPC CPU time
125	SMO2ELTM	XL8	CLIENT ELAPSED time (TOD)
133	SMO2WRTO	XL8	RAW TOTAL BYTES WRITTEN
141	SMO2USCN	F	USER count FOR THIS INTERVAL
145	SMO2MXUS	F	MAX INTERVAL CONCURRENT USERS
149	SMO2RPHW	F	RPC HIGH WATER MARK
153	SMO2RPCU	F	CURRENT NUMBER EXECUTING RPCS
157	SMO2CLWT	XL8	CLIENT WAIT time
165	SMO2CLRC	F	CLIENT READ DATA count
173	SMO2ENZQ	D	Enclave zIIP QUALIFIED CPU time
181	SMO2ENZI	D	Enclave zIIP CPU time
189	SMO2ENZC	D	Enclave zIIP time ON CP
197	SMO2SLCP	D	SSL CPU time
205	SMO2SRCP	D	SRB CPU time
213	SMO2ASTZ	D	ZIIP CPU TIME
221	SMO2ASZC	D	ZIIP TIME ON CP
229	SMO2ASTF	D	TIME ON IFA
237	SMO2ASFC	D	IFA TIME ON CP

Record Subtype 03: SEF Rule Disablement

This record is created whenever an Event Facility (SEF) rule is disabled. All the resources that are used by all connections during that interval are recorded in this record.

About this task

These records are typically written when the Data Virtualization Manager server is shutdown. They are also written if a rule is manually disabled.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(SMFRULEDISABLE) VALUE(YES)"
```

Where SMFRULEDISABLE indicates whether this type of SMF record should be written.

Results

The following table lists the parameters used to configure the Subtype 03 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none">• X'10' = MVS/ESA 4• X'08' = MVS/XA• X'04' = MVS/ESA• X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM03RLTY	C	Rule type flag
38	SM03LACK	XL8	Last time this rule fired (TOD)
49	SM03PRCN	F	Process count
53	SM03FILI	F	Firing limit
57	SM03FIMX	F	Firing high water mark per interval
61	SM03RSNM	CL8	Ruleset name
69	SM03RLNM	CL8	Rule name
77	SM03ENTM	BL4	Rule enablement time (TIME BIN)
81	SM03ENDT	PL4	Rule enablement date (OCYYDDDF)
85	SM03CR	CL128	Rule criterion
213	SM03ENTT	XL4	Total enabled time in seconds

Record Subtype 04: Global Variable

This record is used to collect statistics about global variable utilization.

About this task

No steps are required to enable the Subtype 04 record. A single Subtype 04 record is written by the Data Virtualization Manager server when it is shut down and the System Event Facility (SEF) is in use.

The following table lists the parameters used to configure the Subtype 04 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM04_OP_OFFSET	F	Offset to the permanent section
41	SM04_OP_LENGTH	H	Length of the permanent section
43	SM04_OP_NUMBER	H	Number of permanent sections
45	SM04_OT_OFFSET	F	Offset to the temporary section
49	SM04_OT_LENGTH	H	Length of the temporary section
51	SM04_OT_NUMBER	H	Number of temporary sections
53	SM04_OO_OFFSET	F	Offset to the opsvalue section
57	SM04_OO_LENGTH	H	Length of the OPSVALUE section
59	SM04_OO_NUMBER	H	Number of OPSVALUE sections
61	SM04_P_NUM_GLOBS	F	Number of global variables (permanent section)
65	SM04_P_MAX_BLOCKS	F	Maximum number of blocks (permanent section)
69	SM04_P_HIGH_USED	F	High-used block count (permanent section)
73	SM04_P_IN_USE_BLK	F	Number of in-use blocks (permanent section)
77	SM04_P_FREE_BLK	F	Number of free blocks on free chain (permanent section)

Table 29. Subtype 04 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
81	SM04_P_FREE_AREAS	F	Number of free areas on free chain (permanent section)
85	SM04_P_PAGES	F	Number of pages in global workspace (permanent section)
89	SM04_P_UPDATES	F	Global variable update count (permanent section)
93	SM04_P_CHKPT_INTVL	F	SYSCCHK1 checkpoint interval in seconds
97	SM04_P_CHKPT_COUNT	F	SYSCCHK1 checkpoint count (permanent section)
101	SM04_P_CHKPT_RETRY	F	SYSCCHK1 checkpoint retry count
105	SM04_P_ERRORS	F	Global variable error message count (permanent section)
109	SM04_T_NUM_GLOBS	F	Number of global variables (temporary section)
113	SM04_T_MAX_BLOCKS	F	Maximum number of blocks (temporary section)
117	SM04_T_HIGH_USED	F	High-used block count (temporary section)
121	SM04_T_IN_USE_BLKs	F	Number of in-use blocks (temporary section)
125	SM04_T_FREE_BLKs	F	Number of free blocks on free chain (temporary section)
129	SM04_T_FREE_AREAS	F	Number of free areas on free chain (temporary section)
133	SM04_T_PAGES	F	Global variable update count (temporary section)
137	SM04_T_UPDATES	F	Global variable error message count (temporary section)
141	SM04_T_ERRORS	F	Global variable error message count (temporary section)
149	SM04_O_SYS_OPSVAL	F	Normal opsvalue function calls
153	SM04_O_GVAC_TOTAL	F	Internal OPSVALUE - unknown caller
157	SM04_O_GVAC_UNKNWN	F	Internal OPSVALUE - TOD catchup
161	SM04_O_GVAC_TODC	F	Internal OPSVALUE - TOD catchup
165	SM04_O_GVAC_EVENT	F	Internal OPSVALUE - GLVEVENT
169	SM04_O_GVAC_JOBID	F	Internal OPSVALUE - GLVJOBID

Record Subtype 05: Services (Non-SOAP requests)

This record is used to log Services for non-SOAP Web requests.

About this task

The layout for the Subtype 05 record can be found in member OPSMRC of the *hlq.SAVZSAMP* data set.

A sample SAS program is provided which can be used to print these SMF fields. The program is located in member AVZSFDV5 of the *hlq.SAVZCNTL* member.

No steps are required to enable Subtype 05 records.

The following table lists the parameters used to configure the Subtype 05 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMF ID)
15	SMFHSSID	CL4	Subsystem ID (SWS_)
19	SMFHSUTY	BL2	Record subtype (05)
21	SMFHVRCD	CL8	SWS version code
29	SMFHRS00	CL8	Reserved for future use
37	SMO5CLIP	CL16	Client IP address
53	SMO5SMID	CL4	Host system SMFID
57	SMO5PDSS	CL4	Product subsystem name
61	SMO5CLUS	CL8	Client user ID or blanks
69	SMO5AUTH	CL4	Client authorization status: NONE: Authorization not sent SENT: Authorization information sent but was not used by the server YES: Client user ID/password were valid NO: Client user ID/password were invalid
73	SMO5RS00	CL4	Reserved for future use
77	SMO5SRCP	D	CPU time that is used (TIMEUSED macro)
85	SMO5CNID	XL4	Connection ID
89	SMO5LGTM	XL8	Transaction connect time (GMT TOD)
97	SMO5ELTM	XL8	Transaction elapsed time
105	SMO5WRTO	XL8	Total bytes written (raw)
113	SMO5RS01	XL4	Reserved for future use
117	SMO5ADLT	XL8	Transaction connect time (local TOD)
125	SMO5MTCT	F	Count of URL matches processed
129	SMO5ABCD	XL4	Transaction abend code (if any)

Table 30. Subtype 05 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
133	SMO5ABRS	XL4	Transaction abend reason (if any)
137	SMO5TRRC	F	Overall return code
141	SMO5TRST	F	HTML status code
145	SMO5TRRS	F	Reason code
149	SMO5IPAD	F	IP address of client
153	SMO5DBCP	CL8	DB2 CPU time (TOD Format)
161	SMO5NTCP	CL8F	Network CPU time (TOD Format)
169	SMO5RXCP	CL8	IBM Data Virtualization Manager for z/OS/REXX CPU time (TOD Format)
177	SMO5RPCP	CL8	User program CPU time (TOD Format)
185	SMO5OHCP	CL8	Other CPU time (TOD Format)
193	SMO5SLCP	CL8	SSL processing CPU time (TOD Format)
201	SMO5ENCP	CL8	Enclave CPU time (TOD Format)
209	SMO5SRBT	CL8	SRB CPU time (TOD Format)
217	SMO5RS02	CL8	Reserved for future use
225	SMO5RDTO	XL8	Total bytes sent inbound
233	SMO5INUR	CL128	Original inbound URL value
361	SMO5RESC	F	Count of URL re-scans
365	SMO501CR	CL128	WWW rule criterion (URL match string)
493	SMO501RS	CL8	WWW rule event procedure set name
501	SMO501RL	CL8	WWW rule event procedure member name
509	SMO501EU	CL8	Runtime MVS user ID in effect (TOD Format)
517	SMO5LSCR	CL128	WWW rule criterion (URL match string)
645	SMO5LSRS	CL8	WWW rule event procedure set name
653	SMO5LSRL	CL8	WWW rule event procedure member name
661	SMO5LSEU	CL8	Runtime MVS user ID in effect (TOD Format)
669	SMO5USR1	CL256	User data area 1
925	SMO5USR2	CL256	User data area 2
1181	SMO5ENZQ	D	Enclave zIIP qualified time (TOD Format)
1189	SMO5ENZI	D	Enclave zIIP CPU time (TOD Format)
1197	SMO5ENZC	D	Enclave zIIP time on CP (TOD Format)
1205	SMO5IPV6	XL16	IPV6 ADDRESS OF CLIENT
1221	SMO5IP6C	CL40	IPV6 ADDRESS OF CLIENT

Record Subtype 06: Per Transaction SMF Records

This record is used to log each inbound client request.

About this task

Each SMF transaction record contains information about all the work that is done on behalf of the client for each transaction request. The inbound client request may have caused zero, one, or more SQL operations to be run. A high number of Subtype 06 SMF records may be written in high volume environments because one SMF record is created for each transaction.

A sample SAS program is provided which can be used to print these SMF fields. The program is located in the hlq.SAVZEXEC(AVZSIN00) file data set.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME SMFTRANSACT VALUE(YES)"
```

Where SMFTRANSACT controls the creation of SMF transaction records. When set to YES, an SMF record is created for each inbound client request.

Results

The following table lists the parameters used to configure the Subtype 06 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none">• X'10' = MVS/ESA 4• X'08' = MVS/XA• X'04' = MVS/ESA• X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SMO6CLNA	CL16	Client machine's hostname
53	SMO6CLTY	CL8	Client communication type
61	SMO6IPAD	XL4	IP address for TCP/IP clients
65	SMO6CLUS	CL8	Client user ID
73	SMO6CNID	XL4	Unique client connection ID

Table 31. Subtype 06 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
77	SMO6SQOP	XL2	SQL operation code
79	SMO6GNID	CL8	Generic user ID
87	SMO6EXSZ	H	Extended user ID size
89	SMO6EXID	CL50	Extended user ID area
89	SMO6SIID	CL16	SQLSETEI client user identification
105	SMO6WSNA	CL18	SQLSETEI client workstation name
139	SMO6GNVL	CL1	Validation of generic ID
140	SMO6SETI	CL1	Extended user ID IS SQLSETEI Y or N
141	SMO6PDSS	CL4	4-character IBM Data Virtualization Manager for z/OS subsystem name
145	SMO6PLAN	CL8	DB2 plan name
153	SMO6SSNA	CL4	DB2 subsystem name
157	SMO6ADLT	XL8	Client logon time adjusted for GMT to local time
165	SMO6ADCU	XL8	Current time (adjusted for GMT)
173	SMO6ELTM	XL8	Elapsed time of the client connection
181	SMO6SQEL	XL8	Current SQL statement elapsed time
189	SMO6SQCP	XL8	Current SQL statement CPU time
197	SMO6SQRC	F	Current SQL statement return code
201	SMO6SQRE	F	Current SQL statement reason code
205	SMO6SQSQ	F	Current SQL statement SQL CODE
209	SMO6SQAB	F	Current SQL statement Abend code
217	SMO6VCID	F	VCID of current user
221	SMO6APPL	CL32	SQLSETEI application name
221	SMO6APNA	CL18	Application name
253	SMO6ATKN	CL22	SQLSETEI accounting token
281	SMO6NASB	CL8	Natural subprogram name
289	SMO6SQAC	F	Actual SQL string length
293	SMO6SQLN	F	SQL source length
297	SMO6SQSR	CL256	SQL source string

Record Subtype 09: Storage Interval Summary

This record is used to monitor Data Virtualization Manager server storage usage above and below the 16 MB threshold.

About this task

This record is written at the end of every Data Virtualization Manager server storage recording interval. They are set by the CHECKSTORAGEINTERVAL parameter. If the CHECKSTORAGEINTERVAL parameter is set to 900 (the default), storage usage recording in the Data Virtualization Manager server is disabled.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(LOGSTORAGESMF) VALUE(YES)"
```

Where LOGSTORAGESMF controls whether storage usage information should be written to SMF. Storage usage information can also be written to a DB2 table.

Results

The following table lists the parameters used to configure the Subtype 09 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SMO9SMID	CL4	Host system SMFID
41	SMO9PDSS	CL4	Product subsystem name
45	SMO9RCTY	C	Record type
53	SMO9INST	CL8	Interval start time
77	SMO9MXUS	F	Max interval concurrent user
81	SMO9TSSP	F	Transient subpool
85	SMO9TSBE	F	Transient HI ALLOC BTL

Table 32. Subtype 09 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
89	SMO9TSAB	F	Transient HI ALLOC ATL
93	SMO9HWBA	246D	HI ALLOC BTL HI ALLOC ATL

Record Subtype 10: APPC/MVS Interval Summary

This record is used to log APPC/MVS interval summary information.

Before you begin

APPC/MVS monitoring must be enabled for SMF recording of APPC/MVS summary records.

About this task

Subtype 10 records are written at the end of every Data Virtualization Manager server recording interval (which defaults to 15 minutes).

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(MONITORAPPC/MVS) VALUE(YES)"
```

Where MONITORAPPC/MVS specifies whether to monitor APPC/MVS conversations.

Results

The following table lists the parameters used to configure the Subtype 10 record:

Table 33. Subtype 10 Record Information

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM10SMID	CL4	Host system SMFID

Table 33. Subtype 10 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
41	SM10PDSS	CL4	Product subsystem name
45	SM10RCTY	C	Record type
53	SM10INST	XL8	Interval start time
77	SM10CVTO	F	Total conversations
81	SM10ALTO	F	Total allocated conversations
85	SM10SNT0	F	Total number of sends
93	SM10SDTO	D	Total data sent
101	SM10RCTO	F	Total number of receives
109	SM10RDTO	D	Total data received
117	SM10ACTO	F	Total active conversations

Record Subtype 11: APPC/MVS Conversation Summary SMF

This record is only used internally to display IMS APPC/MVS real-time detail.

About this task

No additional steps are required to enable Subtype 11 records.

The following table lists the parameters used to configure the Subtype 11 record:

Table 34. Subtype 11 Record Information

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYDDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM11SMID	CL4	Host system SMFID
41	SM11PDSS	CL4	Product subsystem name
45	SM11RCTY	C	Record type

Table 34. Subtype 11 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
53	SM11INST	XL8	Internal start time
77	SM11CVID	XL8	Conversation ID
85	SM11INOT	F	Inbound/Outbound indicator
89	SM11PLLO	F	Partner LU location
93	SM11CVKN	F	Conversation kind
97	SM11PLUW	XL26	Logical unit of work ID
123	SM11CVCO	XL8	Conversation correlator
131	SM11USID	CL10	Conversation Userid
141	SM11SCNM	CL8	Scheduler name
149	SM11TPNM	CL8	TP name
157	SM11LTPN	CL8	Local TP name
165	SM11LUNM	CL8	LU name
173	SM11PLNM	CL17	Partner LU name
193	SM11ARTM	XL8	Allocate arrival time
201	SM11AVTM	XL8	Conversation available time
209	SM11CSTM	XL8	Conversation start time
217	SM11CETM	XL8	Conversation end time
225	SM11MDNM	CL8	Mode name
233	SM11SYLV	F	Synchronization level
237	SM11SNTO	F	Total sends
245	SM11SDTO	D	Total data sent
253	SM11RCTO	F	Total receives
261	SM11RDTO	D	Total data received
269	SM11CSTO	F	Total callable service
273	SM11LSRC	F	Last service return code
277	SM11LSRE	F	Last service reason code
281	SM11CVST	F	Conversation state
285	SM11LSBT	XL8	Last service start time
293	SM11LSET	XL8	Last service end time
301	SM11URID	XL16	Unit of recovery identifier
317	SM11CNID	F	Connection ID
321	SM11CBAD	A	Count of URL re-scans

Record Subtype 13: DB2 SQL Errors

This record is used to record DB2 SQL errors.

About this task

This record is used for logging DB2 SQL errors. The LOGERRORSSMF parameter is used in addition to the LOGERRORS parameter, which logs DB2 SQL errors to a DB2 table.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(LOGERRORSSMF) VALUE(YES)"
```

Where LOGERRORSSMF controls whether DB2 SQL error information should be written to SMF. Set the value to YES to generate SMF Subtype 13 records.

Results

The following table lists the parameters used to configure the Subtype 13 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG • SMFHESA4 • SMFHXA • SMFHESA • SMFHVS2	BL1	Header flag byte: • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYYDDDF)
9	SM13GNVL	CL1	VALIDATION OF GENERIC ID
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS VERSION CODE
37	SM13SMID	CL4	Host system SMFID
41	SM13PDSS	CL4	PRODUCT subsystem name
45	SM13RCTY	C	Record type
49	SM13SSAC	CL4	GROUP ATTACHMENT MEMBER name
69	SM13USID	CL8	CLIENT USER ID
77	SM13GNID	CL8	GENERIC USER ID
85	SM13EXID	CL(2+254)	EXTENDED USER ID

Table 35. Subtype 13 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
341	SM13HONA	CL(2+100)	CLIENT HOST name
441	SM13PRTY	CL(2+8)	PROTOCOL TYPE
453	SM13IPAD	XL4	IP ADDRESS FOR IP CLIENTS
457	SM13LUNA	CL(2+17)	LU name FOR LU 6.2 CLIENTS
477	SM13CNID	F	Session ID
481	SM13TMSP	CL8	CURRENT TIMESTAMP
489	SM13LGTM	CL8	LOGON TIMESTAMP
497	SM13APNA	CL(2+18)	APPLICATION name
517	SM13PLAN	CL8	DB2 plan name string
525	SM13SSNA	CL4	DB2 subsystem NAME STRING
529	SM13CUNM	F	Cursor number
533	SM13RC	F	Return code
537	SM13RECD	F	Reason code CODE
541	SM13SQCD	F	SQL CODE
545	SM13ABCD	F	ABEND CODE
549	SM13STNM	F	STATEMENT NUMBER
553	SM13STTY	F	STATEMENT TYPE
557	SM13IPV6	XL16	TCP/IP CLIENT IPV6 ADDRESS
805	SM13SQCA	XL136	SQLCA AREA (SQLLENEQ)
941	SM13SQSR	CL(2+254)	SQL SOURCE STRING
1197	SM13MGTX	CL(2+254)	MESSAGE TEXT

Record Subtype 14: Client Response Time

This record is used to capture client application response time exceptions.

About this task

The Subtype 14 record is written when a client application response time exception occurs. An exception occurs when the client measured response time is greater than the customer supplied response time for a particular application. Subtype 14 records are used with the Response Time Monitor feature.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(MONRESPONSETIME) VALUE(YES)"
```

Where MONRESPONSETIME causes monitoring of the client response time if application names are defined in the Data Virtualization Manager configuration member by using the DEFINE RTMONAPP statement.

Results

The following table lists the parameters used to configure the Subtype 14 record:

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS VERSION CODE
37	SM14RCTY	C	Record type
41	SM14APNM	CL32	APPLICATION NAME
73	SM14LNID	CL100	CLIENT NETWORK USER ID
173	SM14IPAD	XL4	IP ADDRESS FOR IP CLIENTS
177	SM14USID	CL8	CLIENT USER ID
184	SM14DNDA	CL100	CLIENT DOMAIN NAME
285	SM14TMMI	F	Response time in milliseconds (This is the actual client response time for the transaction that produced the exception event)
289	SM14TRTR	F	Total number of client response time records
293	SM14SRTR	F	Sum of the total response time for all of the records
297	SM14TMGR	F	Total number of client response time records that missed the response time goal
301	SM14SMGR	F	Sum of the total response time for the records that missed the response time goal
305	SM14TGRT	F	Client response time goal (this is the acceptable response time)

Record Subtype 16: Record Broker Statistics

This record is used to record broker statistics at DVS server shutdown.

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM16SMID	CL4	FMCHAR - HOST SYSTEM SMFID
41	SM16PDSS	CL4	FMCHAR - PRODUCT SUBSYSTEM NAME
45	SM16NAME	CL50	FMCHAR - STRUCTURE NAME
101	SM16DATE	D	FMCKVL - STCK VALUE OF CREATION DATE
109	SM16BKSN	CL32	FMCHAR - BROKER SERVER NAME
141	SM16BKSC	CL32	FMCHAR - BROKER SERVER CLASS
173	SM16BKSS	CL32	FMCHAR - BROKER SERVER SERVICE
205	SM16CIXF	CL8	FMCHAR - CICS PROGRAM TO XCTL TO
213	SM16CICL	H	FMHEX - CICS USER PASSED COMMARE LNG
215	SM16CICA	CL250	FMCHAR - CICS USER PASSED COMMAREA

Table 37. Subtype 16: Record Broker Statistics (continued)

Offset	Field Name	Field Subtype or Value	Description
465	SM16BKST	X	FMHEX - BROKER SERVER TYPE
466	SM16BKCP	X	FMHEX - CLIENT/ SERVER CONN PERSIST
471	SM16MAXS	H	FMBIIG - BROKER MAX SERVERS ALLOWED
473	SM16CLNA	F	FMHEX - BROKER SVR/CLI NONACTIV TIMR
477	SM16SVNA	F	FMHEX - SERVER NONACTIVITY TIMR
485	SM16SVIN	F	FMBIIG - SERVER INACTIVITY TIMEOUT CT
489	SM16MAXR	F	FMBIIG - SERVER MAX RECEIVES
493	SM16REGC	F	FMBIIG - SERVER REGISTRATION COUNT
497	SM16DREG	F	FMBIIG - SERVER DE-REGISTRATION COUNT
501	SM16RCVC	F	FMBIIG - SERVER RECIEVE COUNT
505	SM16SNDC	F	FMBIIG - SERVER SEND COUNT
509	SM16TIMC	F	FMBIIG - SERVER TIMEOUT COUNT
513	SM16ABNC	F	FMBIIG - SERVER ABEND COUNT
517	SM16WATC	F	FMBIIG - SERVER WAIT COUNT
521	SM16ACTC	F	FMBIIG - HIGH WATER SERVERS ACTIVE
525	SM16ACTT	D	FMCKVL - TIME HIGH WATER MARK TAKEN
533	SM16SUST	D	FMCKVL - TIME MAP WAS LAST SUSPENDED
541	SM16SUSR	F	FMBIIG - SUSPENSION SECONDS REMAINING
545	SM16SUSE	CL8	FMCHAR - ERROR THAT CAUSED SUSPENSION
553	SM16SUSC	F	FMBIIG - SERVER SUSPENSION COUNT

Table 37. Subtype 16: Record Broker Statistics (continued)

Offset	Field Name	Field Subtype or Value	Description
557	SM16SUEC	F	FMBIIG - SERVER TOTAL ERROR COUNT

Record Subtype 17: ADABAS Command by DBID Records

This record is used to capture the number of times a ADABAS database is accessed and the number of commands that were issued against the database before each session ended.

About this task

A Subtype 17 record is written for each Database ID (DBID) referenced and each record contains the number of times that commands were issued against the database before the session ended.

Procedure

To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(ADABASDBIDSMF) VALUE(YES)"
```

Where ADABASDBIDSMF causes one SMF record to be written per DBID accessed at the end of each session. The records contain command usage statistics.

Results

The following table lists the parameters used to configure the Subtype 17 record:

Table 38. Subtype 17 Record Information

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM17SMID	CL4	Host system SMF identification
41	SM17PDSS	CL4	Product subsystem NAME
45	SM17ID	CL8	Connection ID

Table 38. Subtype 17 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
53	SM17LID	CL8	Logon user ID
61	SM17DBID	H	ADABAS identifier (DBID)
65	SM17A1	F	A1 COUNT
69	SM17BT	F	BT COUNT
73	SM17C1	F	C1 COUNT
77	SM17C3	F	C3 COUNT
81	SM17C5	F	C5 COUNT
85	SM17E1	F	E1 COUNT
89	SM17ET	F	ET COUNT
93	SM17HI	F	HI COUNT
97	SM17L1	F	L1 COUNT
101	SM17L4	F	L4 COUNT
105	SM17L2	F	L2 COUNT
109	SM17L5	F	L5 COUNT
113	SM17L3	F	L3 COUNT
117	SM17L6	F	L6 COUNT
121	SM17L9	F	L9 COUNT
125	SM17LF	F	LF COUNT
129	SM17N1	F	N1 COUNT
133	SM17N2	F	N2 COUNT
137	SM17RC	F	RC COUNT
141	SM17RE	F	RE COUNT
145	SM17RI	F	RI COUNT
149	SM17S1	F	S1 FIND RECORDS
153	SM17S2	F	S2 FIND RECORDS
157	SM17S3	F	S3 FIND RECORDS
161	SM17S5	F	S5 FIND RECORDS
165	SM17S8	F	S8 FIND RECORDS
169	SM17S9	F	S' SORT ISN LIST
173	SM17RS02	4F	RESERVED

Record Subtype 18: Services Records

This record is used to set the level of recording you want to use for SMF data for Services.

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYDDDDF)
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (xDBy)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM18CLIP	CL16	Client IP address
53	SM18SMID	CL4	Host system SMFID
57	SM18PDSS	CL4	Product subsystem name
61	SM18CLUS	CL8	Client user ID or blanks
69	SM18AUTH	CL4	Client authorization status: <ul style="list-style-type: none"> • C'NONE' = None (authorization was not sent) • C'SENT' = Sent (authorization was sent) • C'YES' = YES (client user ID/password were valid) • C'NO' = NO (client user id/password were not valid)
73	SM18PORT	H	FMBIIG - Port number of session
75	SM18TYPE	C	Type of request: <ul style="list-style-type: none"> • C'W' = WEB SERVICE REQUEST • C'T' = TERMINAL SERVER REQUEST • C'C' = WSCICSCONN REQUEST
76	SM18RCTY	C	Record type: <ul style="list-style-type: none"> • C'S' = Session detail record type • C'I' = Interval summary record type • C'V' = Virtual directory summary • C'W' = Web service summary record • C'O' = Operation summary record

Table 39. Subtype 18 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
77	SM18SRCP	D	CPU time used TIMEUSED macro
85	SM18CNID	XL4	Connection ID
93	SM18LGTM	XL8	TRANS connect time (GMT TOD)
101	SM18ELTM	XL8	Transaction elapsed time
109	SM18WRTO	XL8	Total bytes written (RAW)
117	SM18ADLT	XL8	TRANS CONNECT TIME LOCAL TOD
125	SM18ABCD	XL4	Transaction abend code (if any)
129	SM18ABRS	XL4	Transaction abend reason (if any)
133	SM18TRRC	F	Overall return code
137	SM18TRST	F	HTML status code
141	SM18TRRS	F	Reason code
145	SM18IPAD	F	IP address of client
149	SM18DBCP	CL8	DB2 CPU time
157	SM18NTCP	CL8	Network CPU time
165	SM18RXCP	CL8	IBM Data Virtualization Manager for z/OS/REXX CPU time
173	SM18RPCP	CL8	User program CPU time
181	SM18OHCP	CL8	Other CPU time
189	SM18SLCP	CL8	SSL processing CPU time
197	SM18ENCP	CL8	Enclave CPU time
205	SM18RCCT	CL8	TRANS. Count for summary RCD
213	SN18SRBT	CL8	SRB CPU TIME
221	SM18RDTO	XL8	Total bytes sent in-bound
229	SM18INUR	CL128	Original in-bound URL value
357	SM18VDIR	CL128	Virtual directory
485	SM18WSNA	CL128	Web service
613	SM18NASP	CL128	Web service name space
741	SM18WSOP	CL50	Operation name
791	SM18WSTG	CL50	Target system name
841	SM18TRSE	C	SOAP fault length
842	SM18TRFX	CL256	SOAP fault text
1101	SM18ENZQ	D	Enclave zIIP qualified CPU time
1109	SM18ENZI	D	Enclave zIIP CPU time
1117	SM18ENZC	D	Enclave zIIP TIME on CP

Offset	Field Name	Field Subtype or Value	Description
1125	SM18INST	D	Adjusted interval start time

Record Subtype 19: Interval Usage Recording Options

This record is used to set the level of recording you want to use for SMF data for Services.

About this task

You can choose from four existing options to set the level of recording you want. The level is reflected in the Record Type Field (SM18RCTY) of the SMF record or RECORD_TYPE field in the DB2 record. You can choose different recording options for SMF and DB2, or you can choose to use one or all of the four recording options. When you choose more than one option, you get duplicate usage records summarized at different levels. Therefore, if the records are used for billing or usage information, care must be taken to not over calculate values that are based on the same usage information.

For example, resource usage for all the operations of a Web Service is reported in the Web Service level summary record, as well as in the operations summary records. If there is no Services activity at any of these levels, no record is written. The SM18RCTY field should be used to make this determination:

Procedure

1. To enable this record, use the **MODIFY PARM** command to set the parameter in the hlq.SAVZEXEC(AVZSIN00) member as follows:

```
"MODIFY PARM NAME(LOGWSTORTM) VALUE(YES)"
```

Where LOGWSTORTM enables logging Services information for Real-Time Monitoring.

2. Set the WSSMFSSUMMARY parameters as follows:

```
"MODIFY PARM NAME(WSSMFSSUMMARY) VALUE(YES)"
"MODIFY PARM NAME(WSSMFSSUMMARYOPER) VALUE(YES)"
"MODIFY PARM NAME(WSSMFSSUMMARYVDIR) VALUE(YES)"
"MODIFY PARM NAME(WSSMFSSUMMARYWS) VALUE(YES)"
```

Results

The following table lists the parameters used to configure the Subtype 18 record.

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (0CYYDDDF)

Table 40. Subtype 19 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (The following table lists the parameters used to configure)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM19SMID	CL4	Host system SMFID
41	SM19PDSS	CL4	PRODUCT subsystem NAME
45	SM19RCTY	C	Record type: <ul style="list-style-type: none"> • C'I' = INTERVAL Record type • C'F' = FINAL Record type
46	SM19PBTY	C	SOURCE OR DESTINATION TYPE: <ul style="list-style-type: none"> • C'S' = SOURCE TASK • C'D' = DESTINATION TASK
47	SM19TATY	C	SOURCE TYPE: <ul style="list-style-type: none"> • C'2' = DB2 SOURCE TASK • C'I' = IMS SOURCE TASK • C'C' = CICS SOURCE TASK • C'A' = ADABAS SOURCE TASK • C'D' = IDMS SOURCE TASK • C'V' = VSAM SOURCE TASK DESTINATION TYPE: <ul style="list-style-type: none"> • C'H' = HTTP DESTINATION • C'B' = MQ BROKER DESTINATION TASK • C'M' = MQ SERIES DESTINATION TASK
53	SM19INST	CL8	Interval start time
61	SM19TANA	CL8	Publish task name
69	SM19ENCP	CL8	Enclave CPU time
77	SM19CLCP	CL8	CPU time used
85	SM19DBCP	CL8	DB2 CPU time
93	SM19NTCP	CL8	Network CPU time
101	SM19OHCP	CL8	Other CPU time
109	SM19RXCP	CL8	IBM Data Virtualization Manager for z/OS/REXX CPU time
117	SM19RPCP	CL8	User program CPU time
125	SM19ELTM	XL8	Transaction elapsed time

Table 40. Subtype 19 Record Information (continued)

Offset	Field Name	Field Subtype or Value	Description
133	SM19WRTO	XL8	Total bytes written
141	SM19SOCA	F	Number of events captured
145	SM19SOIG	F	Number of events ignored
149	SM19SORU	F	Number of rules run
153	SM19SOFA	F	Number of rule failures
157	SM19SOEQ	F	Number of events queued
165	SM19SOBC	D	Number of bytes captured
173	SM19SOBQ	D	Number of bytes queued
181	SM19DERD	F	Number of events read
185	SM19DESH	F	Number of events shipped
189	SM19DEBS	D	Number of bytes shipped
197	SM19DEFA	F	Number of records failed
201	SM19DEOP	F	Number of connection opens
205	SM19DERF	F	Number of retrievable failures
213	SM19ENZQ	D	Enclave zIIP qualified CPU time
221	SM19ENZI	D	Enclave zIIP CPU time
229	SM19ENZC	D	Enclave zIIP time ON CP
237	SM19SLCP	D	SSL CPU time
245	SM19SRCP	D	SRB CPU time

Record Subtype 20: Transaction SMF Record

This record is used to map the WSC transaction SMF record.

Table 41. Subtype 20: Transaction SMF Record

Offset	Field Name	Field Subtype or Value	Description
1	SMFHFG	BL1	Header flag byte: <ul style="list-style-type: none"> • X'10' = MVS/ESA 4 • X'08' = MVS/XA • X'04' = MVS/ESA • X'02' = VS2
2	SMFHRCTY	BL1	Record Type
3	SMFHTIME	BL4	Record written time (TIME BIN)
7	SMFHDATE	PL4	Record written date (OCYYDDDF)

Table 41. Subtype 20: Transaction SMF Record (continued)

Offset	Field Name	Field Subtype or Value	Description
11	SMFHSYID	CL4	System identification (SMFID)
15	SMFHSSID	CL4	Subsystem ID (AVZS)
19	SMFHSUTY	BL2	Record subtype
21	SMFHVRCD	CL8	IBM Data Virtualization Manager for z/OS version code
37	SM2OCLIP	CL16	CLIENT IP ADDRESS
53	SM2OSMID	CL4	FMCHAR - HOST SYSTEM SMFID
57	SM2OPDSS	CL4	FMCHAR - PRODUCT SUBSYSTEM NAME
61	SM2OCLUS	CL8	CLIENT USERID OR BLANKS
69	SM2OTYPE	C	FMCHAR - TYPE OF REQUEST
70	SM2ORCTY	C	FMCHAR - RECORD TYPE
77	SM2OSRCP	D	FMDWTM - CPU TIME USED TIMEUSED MACRO
85	SM2OCNID	XL4	FMHEX - CONNECTION ID
89	SM2OTMPL	CL256	FMCHAR - TEMPLATE NAME
345	SM2OOPNA	CL128	FMCHAR - OPERATION NAME
345	SM2OOPNS	CL50	FMCHAR - SHORT OPERATION NAME
473	SM2OWURL	CL128	FMCHAR - URL OF WS BEING INVOKED
601	SM2OTRST	F	HTML STATUS CODE
605	SM2OTRSE	C	SOAP FAULT LENGTH
606	SM2OTRFX	CL256	SOAP FAULT TEXT
869	SM2OBCP	CL8	FMDWTM - DB2 CPU TIME
877	SM2ONTCP	CL8	FMDWTM - NETWORK CPU TIME
885	SM2ORXCP	CL8	FMDWTM - DVS/REXX CPU TIME

Table 41. Subtype 20: Transaction SMF Record (continued)

Offset	Field Name	Field Subtype or Value	Description
893	SM2ORPCP	CL8	FMDWTM - USER PROGRAM CPU TIME
901	SM20OHCP	CL8	FMDWTM - OTHER CPU TIME
909	SM20SLCP	CL8	FMDWTM - SSL PROCESSING CPU TIME
917	SM20ENCP	CL8	FMDWTM - ENCLAVE CPU TIME
925	SM20RCCT	CL8	FMBIIG - TRANS. COUNT FOR SUMMARY RCD
933	SM20SRBT	CL8	FMDWTM - SRB CPU TIME
941	SM20ENZQ	D	FMDWTM - ENCLAVE ZIIP QUALIFIED TIME
949	SM20ENZI	D	FMDWTM - ENCLAVE ZIIP CPU TIME
957	SM20ENZC	D	FMDWTM - ENCLAVE ZIIP TIME ON CP
965	SM20INST	D	FMCKVL - INTERVAL START TIME
973	SM20RDTO	XL8	FMBIIG - TOTAL BYTES READ
981	SM20WRTO	XL8	FMBIIG - TOTAL BYTES WRITTEN
989	SM20CLI6	CL40	FMCHAR - IPV6 ADDRESS OF CLIENT

DB2 logging

DB2 logging writes out the total z/OS resource usage information into a DB2 intervals table for a specified time interval.

The Data Virtualization Manager server also writes detailed information for each connection into a DB2 sessions table. When a client disconnects, a record is written to a DB2 sessions table. Use this information to provide detailed reporting of processor resource consumption in your client/server applications.

If SMF logging is also enabled, logging information is written to a set of DB2 tables, and extra Subtype 01 and Subtype 02 Records are written out to SMF. Sub Type 01 records are shared with normal end-of-sessions records. These records can be distinguished via the SM01RCTY field in the SMF type 01 record.

Enabling DB2 logging

Enables logging to DB2 tables using the Data Virtualization Manager servers.

Procedure

1. Use the **MODIFY PARM** command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(DEFAULTDB2PLAN) VALUE(SDBC1010)"  
"MODIFY PARM NAME(DEFAULTDB2SUBSYS) VALUE(XXXX)"  
"MODIFY PARM NAME(LOGDB2SUBSYS) VALUE(DSN)"  
"MODIFY PARM NAME(LOGUSERID) VALUE(SDBB)"  
"MODIFY PARM NAME(RECORDINGINTERVAL) VALUE(900)"
```

The following table lists the parameters used to configure DB2 logging:

Parameter	Description	Valid values
DEFAULTDB2PLAN	Specifies the DB2 plan name that remote clients use to access DB2 when the connection is set to PLAN=DFLT. It is also used as the logging task's target DB2 subsystem when LOGDB2PLNAME is not specified. It is the plan that is used by Streams when connected to DB2.	SDBC1010
DEFAULTDB2SUBSYS	Specifies DB2 subsystem that remote clients use for access DB2 when the connection is set to SUBSYS=DFLT. It is also used as the logging task's target DB2 subsystem when LOGDB2SUBSYS is not specified.	'NONE'
LOGDB2SUBSYS	Controls the DB2 subsystem that is used for all SQL operations. If this parameter is set, then all logging operations are routed to the specified DB2 subsystem. If this parameter is not set, then each logging operation is routed to the DB2 subsystem that the operation was associated with or the default DB2 subsystem if the operation was not associated with any DB2 subsystem.	'NONE'

Parameter	Description	Valid values
LOGUSERID	Controls the DB2 userid that is used for all SQL operations. This userid must have enough authority to update (insert) all of the tables modified by the logging task. If this field is not set, the main product address space userid is used for all update operations.	AVZS
RECORDINGINTERVAL	Controls how often interval summary and per-client SMF and/or SQL records are created. These records show what resources were used during the current recording interval. The interval value is specified in seconds and should be a factor of one hour. The value should divide evenly into 3600.	900

2. The logging tables are created by running the AVZD2LGT script, which is located in *hlq.SAVZCNTL*. The following tables are made available through logging:

- SQL
 - AVZ.APMVSSUM
 - AVZ.ERRORLOG
 - AVZ.INTERVALS
 - AVZ.SESSIONS
 - AVZ.SQLSOURCE
 - AVZ.STORAGE
- Services
 - AVZ.SERVICES
- Streams
 - AVZ.STREAMS

Record: Sessions

This record is used to log usage information for a specific connection during a specified time interval.

About this task

Sessions records get cut at client disconnect time, similar to when an SMF record is written for end of session records. Processor times are either for the entire session or for the interval, depending on the record type. The record type can be determined by the RECORD_TYPE field in the Session record. Values include:

- S: The final end-of-session record.
- F: The final interval record that shows the usage of processor time for that specified interval.
- I: The interim interval record.

Procedure

Use the **MODIFY PARM** command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGLSESSIONINTVALSMF) VALUE(YES)"
"MODIFY PARM NAME(LOGRETAINSESSIONS) VALUE(30)"
"MODIFY PARM NAME(LOGSESSIONS) VALUE(YES)"
"MODIFY PARM NAME(LOGSESSIONSTABLE) VALUE(DVS.SESSIONS)"
```

Results

The following table lists the parameters used to configure the Sessions record:

<i>Table 42. Sessions Record for DB2</i>		
Parameter	Description	Valid values
LOGLSESSIONINTVALSMF	Controls whether interval type records are written to SMF. Interval records may also be written to the session log.	YES Default value is YES. NO
LOGRETAINSESSIONS	Controls the number of days to wait before automatically deleting rows from the sessions table. That is, all rows older than the number of days are deleted. If this value is zero, rows are never automatically deleted from the sessions table.	Number of days Default is 30 days.
LOGSESSIONS	Controls whether session information should be logged. Session information is logged by inserting rows in to a DB2 table. One row is inserted for each session at session termination time.	YES Default value is YES. NO
LOGSESSIONSTABLE	Sets the name of the DB2 table that is used to log session information. If a session is active, a row is inserted into this table as part of session termination.	DVS.SESSIONS Default name

AVZ.SESSIONS

The AVZ.SESSIONS table contains one Interval record for each SQL user for each recording interval. It also contains Sessions records that have the total information for the entire connection.

Column	Description
USERID	User ID associated with the record.
CLIENT_SYSTEM	Client PC name.
PROTOCOL	Either TCP/IP or LU 6.2.
RECORD_TYPE	Either Session or Interval.
TOTAL_CPU TIME	Total CPU time.

Column	Description
SRB_CPUETIME	SRB CPU time. Time is included in TOTAL_CPUETIME.
DATABASE_CPUETIME	Total database CPU time that is used by IMS and DB2. This field does not reflect CPU usage by RPCs that access IMS and DB2.
NETWORK_CPUETIME	Total network CPU time trappable within Data Virtualization Manager server. Note: Network CPU time cannot be trapped within Data Virtualization Manager server. Therefore, this field may not reflect the total CPU time.
REXX_CPUETIME	Total CPU time that is used by running SEF REXX programs.
RPC_CPUETIME	Total CPU time that is used by RPCs. This time includes CPU time that is used in DB2 and IMS if the RPCs accessed these databases.
SSL_CPUETIME	SSL processing CPU time.
ENCLAVE_CPUETIME	Total CPU time that is associated with the WLM enclave. Note: Other CPU times in this record may also be associated with the Enclave CPU time. Also, other tasks may have contributed CPU time to the same Enclave.
ZIIP_QUALIFIED	Enclave zIIP qualified CPU time.
ZIIP_CPU	Enclave zIIP CPU time.
ZIIP_ON_CP	Enclave zIIP time on CP.
OTHER_CPUETIME	All of the unaccountable CPU time.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The 4-character IBM Data Virtualization Manager for z/OS subsystem ID.
DRIVER_VERSION	The IBM Data Virtualization Manager for z/OS Driver version.
DRIVER_DATE	The IBM Data Virtualization Manager for z/OS Driver date.
CONNECTION_ID	The unique Data Virtualization Manager server connection ID.
LOGON_TIME	Time the user logged on.
LOGOFF_TIME	Time the user logged off. For Interval records, this value is NULL.
INTERVAL_START	Interval start time. For Sessions records, this value is NULL.
CONNECT_TIME	Number of seconds the user was connected. For Interval records, this value is NULL.
BYTES_READ	Total number of bytes of data that is read from the client workstation.
BYTES_WRITTEN	Total number of bytes written to the client workstation.
COMMIT_COUNT	Total number of commits performed.
ROLLBACK_COUNT	Total number of rollbacks performed.

Column	Description
SQL_COUNT	Total number of SQL queries run.
RPC_COUNT	Total number of RPCs run.
ABEND_CODE	Abend for the session, if one occurred. For Interval records, this value is NULL.
IP_ADDRESS	For TCP/IP connections, this is the IP address of the client workstation; otherwise, this value is NULL.
LU_NAME	For LU 6.2 connections, the LU name that is used for the connection.
ORIGINAL_USERID	Original user ID, recorded in case it was changed by a SEF rule.
PLAN	DB2 plan that is used.
DATABASE	DB2 subsystem to which the user is connected.
DB_GROUP_MEMBER	DB2 group attachment member name.
APPLICATION	Application name. This value is set by the client's ODBC connection information.
USERPARM	Optional userparm from the client. This value is set by the client's ODBC connection information.
ELAPS_READ_TIME	The total number of read operations from Data Virtualization Manager server to the Client.
TOTAL_READ_COUNT	The total time (in seconds) that those reads were outstanding. If you ignore the transmission times, the time is the user response time.

Record: Interval

This record is used to log the total z/OS resources that are used by all connections from the starting interval time, until the next starting interval time.

About this task

This table contains one entry for each interval. The interval time frame is determined by the Data Virtualization Manager server RECORDINGINTERVAL parameter.

Interval records are written to SMF Subtype 02 records if the Data Virtualization Manager server is configured to write SMF records.

The following table lists the parameters used to configure the Interval record:

Procedure

1. To name the DB2 table that is to contain the interval, add the following parameter in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGINTERVALSTABLE) VALUE(AVZ.INTERVALS)"
```

2. Use the **MODIFY PARM** command in the hlq.SAVZEXEC(AVZSIN00) member, to enable or disable logging of SMF interval records.

- To disable the logging of Interval records to SMF, set the LOGINTERVALSSMF parameter to NO:

```
"MODIFY PARM NAME(LOGINTERVALSSMF) VALUE(NO)"
```

- To enable the logging of Interval records to SMF, set the LOGINTERVALSSMF parameter to YES, and then set the LOGRETAININTERVALS parameter interval value accordingly:

```
"MODIFY PARM NAME(LOGINTERVALS) VALUE(YES)"
"MODIFY PARM NAME(LOGRETAININTERVALS) VALUE(30)"
```

Results

The following table lists the parameters used to configure the Interval Record.

<i>Table 43. Interval Record for DB2</i>		
Parameter	Description	Valid values
LOGINTERVALS	Controls whether session interval information should be logged. Session Interval information is logged by inserting rows in to a DB2 table. One row is inserted for each session at the end of each recording interval and at session termination time.	YES Default value is YES. NO
LOGINTERVALSSMF	Controls whether session interval information should be written to SMF.	YES Default value is YES. NO
LOGINTERVALSTABLE	Specifies the name of the DB2 table that is used to log interval information. If interval recording is active, a row is inserted into this table at the end of each recording interval,	AVZ.INTERVALS Default name
LOGRETAININTERVALS	Controls the number of days to wait before automatically deleting rows from the interval summary table. That is, all rows older than the number of days are deleted. If this value is zero, rows are never automatically deleted from the interval summary table.	Number of days Default is 30 days.

AVZ.INTERVALS

The AVZ.INTERVALS table contains precise z/OS resource usage information for all SQL connections that were active for an interval. Each record in the table is associated with a starting interval time.

Column	Description
RECORD_TYPE	Describes the record type. Currently, this record type is always Summary.
TOTAL_CPU TIME	Total CPU time that is used by all connections.
SRB_CPU TIME	Time is included in TOTAL_CPU TIME.
DATABASE_CPU TIME	Total database CPU time that is used by all connections, currently consists of IMS and DB2. This field does not reflect CPU usage by RPCs that access IMS and DB2.

Column	Description
NETWORK_CPU TIME	Total network CPU time trappable within Data Virtualization Manager server. Note: Because some network CPU time cannot be trapped within Data Virtualization Manager server, this field may not reflect the total CPU time.
REXX_CPU TIME	Total CPU time that is used by running SEF REXX programs.
RPC_CPU TIME	Total CPU time that is used by RPCs. This includes CPU time that is used in DB2 and IMS if the RPCs accessed these databases.
SSL_CPU TIME	SSL processing CPU time.
ENCLAVE_CPU TIME	Total CPU time that is associated with the WLM enclave. Note: Other CPU times in this record may also be associated with the Enclave CPU time. Also, other tasks may have contributed CPU time to the same Enclave.
ZIIP_QUALIFIED	Enclave zIIP qualified CPU time.
ZIIP_CPU	Enclave zIIP CPU time.
ZIIP_ON_CP	Enclave zIIP time on CP.
OTHER_CPU TIME	All of the unaccountable CPU time that is associated with IBM Data Virtualization Manager for z/OS itself.
USER_COUNT	The number of users that were connected during this interval.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The 4-character IBM Data Virtualization Manager for z/OS subsystem ID.
INTERVAL_START	Interval start time. For Sessions records, this value is null.
CONNECT_TIME	N/A. Set to null.
BYTES_READ	Total number of bytes sent from the client connections.
BYTES_WRITTEN	Total number of bytes of data that is written down to the client workstations.
COMMIT_COUNT	Total number of commits performed.
ROLLBACK_COUNT	Total number of rollbacks performed.
SQL_COUNT	Total number of SQL queries run.
RPC_COUNT	Total number of RPCs run.
MAXIMUM_USER	Maximum number of users this interval.

Record: SQL Source

This record is used to capture SQL information for use with the Server Activity Monitor (SAM).

About this task

This record can also be used with the Dynamic-to-Static Analyzer (DSA). Recording dynamic SQL statements in the table provides a central location for extracting dynamic SQL statements for input to the DSA application.

Procedure

Use the **MODIFY PARM** command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGRETAINSQL) VALUE(30)"  
"MODIFY PARM NAME(LOGSOURCETABLE) VALUE(AVZ.SQLSOURCE)"  
"MODIFY PARM NAME(LOGSQLSOURCE) VALUE(YES)"
```

The following table lists the parameters used to configure the SQL Source record:

Parameter	Description	Valid values
LOGRETAINSQL	Specifies the number of days to wait before automatically deleting SQL from the SQL source table. That is, all rows older than the number of days are deleted. If this value is zero, then rows are never automatically deleted from the SQL source table.	Number of days Default is 30 days.
LOGSOURCETABLE	Sets the name of the DB2 table that is used to log SQL source for conversion from dynamic SQL to static SQL. Each SQL statement is stored in one or more rows of this table.	AVZ.SQLSOURCE Default name
LOGSQLSOURCE	Controls whether SQL source information should be logged. SQL source information is logged by inserting rows in to a DB2 table. When the SQL statement is processed, one row is inserted for each SQL statement. The logged SQL source is used to convert dynamic SQL to static SQL.	YES Default value is YES. NO

AVZ.SQLSOURCE

The AVZ.SQLSOURCE table contains all the dynamic SQL and CALL statements that are run by SQL.

Column	Description
USERID	User ID associated with the record.

Column	Description
GENERIC_ID	An alternative RACF/ACF2/Top Secret USER ID that an authorized Client (one serving multiple people) has passed stating that “this transaction is being run on behalf of this user.” This implementation is associated with Enterprise Auditing.
EXTENDED_USERID	User ID information that is passed by the Client and put into the SMF records.
CLIENT_SYSTEM	System machine name of client, for example, PC name or host name on UNIX.
IP_ADDRESS	For TCP/IP connections, this record is the IP address of the client workstation; otherwise, this value is NULL
LU_NAME	For LU 6.2 connections, the LU name that is used for the connection.
CONNECTION_ID	The unique Data Virtualization Manager server connection ID.
TIME_CURRENT	Timestamp for the current activity.
LOGON_TIME	Timestamp for the actual logon time.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The 4-character IBM Data Virtualization Manager for z/OS subsystem ID.
APPLICATION	Application name. This value is set by the client’s ODBC connection information.
PLAN	DB2 plan that is used.
DATABASE	DB2 subsystem to which the user is connected.
HASH_CODE	Internal Use.
CURSOR	Internal cursor name.
RETURN_CODE	Return code for non-DB2 users.
REASON_CODE	Code that is given as reason for a failure.
SQL_CODE	Return code for DB2 users.
ABEND_CODE	Abend for the mainframe session, if one occurred. For Interval records, this value is NULL.
TIMERONS	The DB2 estimated cost of the statement. A timeron is a unit of measurement used to give a rough relative estimate of the resources, or cost, required by the database server to execute two plans for the same query. The resources calculated in the estimate include weighted CPU and I/O costs.
SQL_LENGTH	The length of the SQL statement.
SQL	The actual SQL statement.
SQL_LOB	Object locator for large objects.
ROWID	DB2 row identifier.

Record: Storage

This record is used to monitor Data Virtualization Manager server storage usage.

About this task

The Storage record is written at the end of every Data Virtualization Manager server storage recording interval.

Procedure

To enable this record, use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(CHECKSTORAGEINTERVAL) VALUE(900)"  
"MODIFY PARM NAME(LOGSTORAGE) VALUE(YES)"  
"MODIFY PARM NAME(LOGSTORAGE$SMF) VALUE(YES)"  
"MODIFY PARM NAME(LOGSTORAGETABLE) VALUE(AVZ.STORAGE)"
```

The following table lists the parameters used to configure the Storage record:

Parameter	Description	Valid values
CHECKSTORAGEINTERVAL	Controls how often (in seconds) statistics for allocated storage are gathered within IBM Data Virtualization Manager for z/OS. A value of zero turns off this function.	Default value is 900. Function is turned off.
LOGSTORAGE	Controls whether storage information is logged. Storage information is logged by inserting rows in to a DB2 table.	YES Default value is YES. NO
LOGSTORAGE\$SMF	Controls whether storage usage information should be written to SMF. Storage usage information can also be written to a DB2 table.	YES NO Default value is NO.
LOGSTORAGETABLE	Sets the name of the DB2 table that is used to log storage information. A row is inserted into this table at the end of each recording interval, if storage logging is active.	AVZ.STORAGE Default name

AVZ.STORAGE

The AVZ.STORAGE is used to record the private and virtual storage that is used by the Data Virtualization Manager server address space in 15-minute intervals.

Column	Description
PRODUCT_SUBSYSTEM	The 4-character IBM Data Virtualization Manager for z/OS subsystem ID.
INTERVAL_START	The start time of summary activity.
MAXIMUM_USERS	The maximum number of users allowed.

Column	Description
SUBPOOL	Name of virtual storage information.
BELOW_16M	Amount of memory in use below 16 MB.
ABOVE_16M	Amount of memory in use above 16 MB.
SMFID	The SMFID as defined within Data Virtualization Manager server.

Record: APPC/MVS

This record is used to log APPC/MVS interval summary information by inserting rows in to a DB2 table. One row is inserted at the end of each recording level.

Procedure

To enable this record, use the **MODIFY PARM** command to add the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGAPMVSSUM) VALUE(YES)"
"MODIFY PARM NAME(LOGAPMVSSUMSMF) VALUE(YES)"
"MODIFY PARM NAME(LOGAPMVSSUMTABLE) VALUE(DVS.APMVSSUM)"
"MODIFY PARM NAME(LOGRETAINAPMVSSUM)VALUE(30)"
"MODIFY PARM NAME(MONITORAPPC/MVS) VALUE(YES)"
```

The following table lists the parameters used to configure the APPC/MVS record:

Parameter	Description	Valid values
LOGAPMVSSUM	Controls whether APPC/MVS interval summary information should be logged. APPC/MVS interval summary information is logged by inserting rows in to a DB2 table. One row is inserted at the end of each recording level.	YES NO Default value is NO.
LOGAPMVSSUMSMF	Controls whether APPC/MVS interval summary information should be written to SMF. APPC/MVS interval summary information can also be written to a DB2 table.	YES Default value is YES. NO
LOGAPMVSSUMTABLE	Sets the name of the DB2 table that is used to log APPC/MVS interval summary information. If APPC/MVS interval summary recording is active, a row is inserted into this table at the end of each recording interval.	AVZ.APMVSSUM Default name

Table 46. APPC/MVS record for DB2 (continued)

Parameter	Description	Valid values
LOGRETAINAPMVSSUM	Specifies the number of days to wait before automatically deleting rows from the APPC/MVS summary table. That is, all rows older than the number of days are deleted. If this value is zero, then rows are never automatically deleted from the APPC/MVS summary table.	0 Default value of 0. Function is turned off.
MONITORAPPC/MVS	Specifies whether to monitor APPC/MVS conversations. This parameter should be set to YES to activate the monitor.	YES Default value is YES. NO

AVZ.APMVSSUM

The AVZ.APMVSSUM table shows the APPC/MVS interval summary information.

Column	Description
RECORD_TYPE	Either Session or Interval.
TOTAL_CONV	The total number of conversations between TCP/IP and IBM Data Virtualization Manager for z/OS.
ALLOCATED_CONV	The number of conversations that are allocated to TCP/IP and IBM Data Virtualization Manager for z/OS.
NUMBER_OF_SENDS	The number of blocks of data sent.
DATA_SENT	The number of bytes of data sent.
NUMBER_OF_RECEIVES	The number of blocks of data received.
DATA_RECEIVED	The number of bytes of data received.
ACTIVE_CONV	The number of conversations in use at the end of the interval.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The 4-character IBM Data Virtualization Manager for z/OS subsystem ID.
INTERVAL_START	Interval start time. For Sessions records, this value is null.

Records: Error Log

This record is used to log system errors.

Procedure

To enable this record, use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member.

```
"MODIFY PARM NAME(LOGERRORS) VALUE(YES)"
"MODIFY PARM NAME(LOGERRORSTABLE) VALUE(AVZ.ERRORLOG)"
"MODIFY PARM NAME(LOGRETAINERRORS) VALUE(30)"
```

The following table lists the parameters used to configure the Error Log record:

<i>Table 47. Error Log records for DB2</i>		
Parameter	Description	Valid values
LOGERRORS	Controls whether error information should be logged. Error information is logged by inserting rows in to a DB2 table. When set to YES, one row is inserted for each error that is detected by the Data Virtualization Manager server address space or reported by an application running under the Data Virtualization Manager server address space.	YES Default value is YES. NO
LOGERRORSTABLE	Sets the name of the DB2 table that is used to log errors. A row is inserted into this table each time Data Virtualization Manager server detects an error. Errors can also be reported by applications running under the control of the Data Virtualization Manager server address space. Note: Error logging can be turned on and off at any time.	AVZ.ERRORLOG Default name
LOGRETAINERRORS	Controls the number of days to wait before automatically deleting rows from the error logging table. For examples, all rows older than the number of days are deleted. If this value is zero, then rows are never automatically deleted from the error logging table.	30 Number of days. Default is 30.

AVZ.ERRORLOG

The AVZ.ERRORLOG table contains records of DB2 SQL failures that result from negative SQL return codes in SQL. Only DB2 is supported by this table.

Column	Description
USERID	User ID associated with the record.
GENERIC_ID	An alternative RACF/ACF2/Top Secret USER ID that an authorized Client (one serving multiple people) passed stating that “this transaction is being executed on behalf of this user”. This implementation is associated with Enterprise Auditing, formerly our Transaction Level Security (“TLS”).
EXTENDED_USERID	User ID information that is passed by the Client and put into the SMF records.

Column	Description
CLIENT_SYSTEM	System machine name of client, for example, PC name or host name on UNIX.
IP_ADDRESS	For TCP/IP connections, this record is the IP address of the client workstation; otherwise, this value is null
LU_NAME	For LU 6.2 connections, the LU name that is used for the connection.
CONNECTION_ID	The unique Data Virtualization Manager server connection ID.
TIME_CURRENT	Timestamp for the current activity.
LOGON_TIME	Timestamp for the actual logon time.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The four-character IBM Data Virtualization Manager for z/OS subsystem ID.
APPLICATION	Application name. This value is set by the client's ODBC connection information.
PLAN	DB2 plan that is used.
DATABASE	DB2 subsystem to which the user is connected.
DB_GROUP_MEMBER	DB2 group attachment member name.
CURSOR	Internal cursor name.
STATEMENT	The statement number.
TYPE	SQL.
RETURN_CODE	Return code for non-DB2 users.
REASON_CODE	Code that is given as reason for a failure.
SQL_CODE	Return code for DB2 users.
ABEND_CODE	Abend for the mainframe session, if one occurred. For Interval records, this value is NULL.
TIMERONS	The DB2 estimated cost of the statement.
SQLCA	SQL communication area, which provides an application program with information about the processing of SQL statements within the program.
SQL	The actual SQL statement.
MESSAGE	The message describing the error.

Record: Services

The Services records are used to set the level of recording you want to use.

About this task

There are existing recording level options from which you can choose. If you are using IBM Data Virtualization Manager for z/OS logging, the level is reflected in the RECORD_TYPE field in the DB2 record. If you are using SMF Records, the level is reflected in the **Record Type** field SM18RCTY.

You can choose to use one or all of the recording options. When you choose more than one option, you get duplicate usage records, summarized at different levels. Therefore, if the records are used for billing or usage information, care must be taken to not over calculate values that are based on the same usage information.

Procedure

1. To enable this record, use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member.

```
"MODIFY PARM NAME (LOGWSTORTM) VALUE(YES)"
```

2. Optional: System page data sets should be reviewed and adjusted to hold more storage requirements. For each Web service run, 1664 bytes are needed in virtual storage. The length of time this information is retained is determined by the WSMEMORYINTERVALS parameter that defaults to 100. The frequency of the recording of Summary records is determined by the RECORDINGINTERVAL parameter. These parameter settings, in addition to the number of Web service transactions run, affect the amount of storage required.
3. Select the type of recording interval that you want to use from the following level options:
 - Interval recording using no specific criteria.
 - Interval recording at the Virtual Directory level.
 - Interval recording at the Web Service level.
 - Interval recording at the operation level.
 - End of Session recording of each Web Service.

Results

The following table lists the parameters used to configure the Service record:

Parameter	Description	Valid values
LOGWSTORTM	Enables logging Services information for Real Time Monitoring (RTM).	YES NO Default value is NO.
RECORDINGINTERVAL	Controls how often interval summary and per-client SMF and/or SQL records are created. These records show what resources were used during the current recording interval. The interval value is specified in seconds and should be a factor of one hour. The value should divide evenly into 3600.	900 (default)
WSMEMORYINTERVALS	Sets the number of intervals to retain in memory for Services interval processing	100 (default)

Services interval recording using no specific criteria

This record enables logging interval information using no specific criteria.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGWSREQUESTSTABLE) VALUE(AVZ.SERVICES)"  
"MODIFY PARM NAME(LOGRETAINWS) VALUE(30)"
```

The following table lists the parameters used to configure this record:

<i>Table 49. Services Record for DB2: Interval recording using no specific criteria</i>		
Parameter	Description	Valid values
LOGRETAINWS	Controls the number of days to wait before automatically deleting rows from the Services table. That is, all rows older than the number of days are deleted. If this value is zero, then rows are never automatically deleted from the URLs table.	Number of days Default is 30 days.
LOGWSREQUESTSTABLE	Sets the name of the DB2 table that is used to log Services information. If a recording is active, a row is inserted into this table for each Web Service.	AVZ.SERVICES

Services interval recording at the Web Services level

This record enables logging interval information at the Web Services level.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member.

```
"MODIFY PARM NAME(LOGWSSUMMARY) VALUE(YES)"  
"MODIFY PARM NAME(LOGWSSUMMARYWS) VALUE(YES)"
```

The following table lists the parameters used to configure the record:

<i>Table 50. Services Record for DB2: Interval Recording at the Web Services Level</i>		
Parameter	Description	Valid values
LOGWSSUMMARY	Enables logging Services summary SMF records to a DB2 table. This parameter contains an overall summary of all Web Services requests, which are written at the end of a specified interval.	YES NO Default value is NO.

Table 50. Services Record for DB2: Interval Recording at the Web Services Level (continued)

Parameter	Description	Valid values
LOGWSSUMMARYWS	Enables logging Services Web Service summary SMF records to a DB2 table.	YES NO Default value is NO.

Services interval recording at the Web Services directory level

This record enables logging interval information at the Web Services directory level.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGWSSUMMARYVDIR) VALUE(YES)"
```

The following table lists the parameters used to configure this record:

Parameter	Description	Valid values
LOGWSSUMMARYVDIR	Enables logging Services Virtual Directory summary SMF records to a DB2 table.	YES NO Default value is NO.

Services interval recording at the operation level

This record enables logging interval information at the operation level.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGWSSUMMARYOPER) VALUE(YES)"
```

The following table lists the parameters used to configure for this record:

Parameter	Description	Valid values
LOGWSSUMMARYOPER	Enables logging Services Operation summary SMF records to a DB2 table.	YES NO Default value is NO.

Services End of Session recording for each web service

This record enables logging interval information for each web service.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGWSREQUESTS) VALUE(YES)"
```

The following table lists the parameters used to configure this record:

Table 51. Services for DB2: End of Session record

Parameter	Description	Valid values
LOGWSREQUESTS	<p>Enables logging Services information. If recording is active, a row is inserted into a table for each Web Service. This may not be practical for large volumes of requests. Consider logging WS Summary records.</p> <p>Note: LOGWSREQUESTS should be set to YES in order to see the URL and Namespace data.</p>	<p>YES NO</p> <p>Default value is NO.</p>

AVZ.SERVICES

The AVZ.SERVICES table shows statistics information for Services.

Column	Description
ORIGINAL_URL	The URL where the Web Service is stored.
CLIENT_USERID	The user ID of the client that sent the request.
CLIENT_SYSTEM	The system that sent the request.
CLIENT_AUTH_USAGE	The authentication mechanism used.
REQUEST_TYPE	The type of service request: Web Service, Terminal Service, or WSCICSCONN request.
RECORD_TYPE	The type of record this is: Interval Summary, Virtual Directory Summary, Web Service Summary, Operation Summary, or Session Detail.
RECORD_COUNT	Transaction count for Summary records.
TOTAL_CPU TIME	Total CPU time used. CPU time may also be accounted for in the Enclave CPU time.
SRB_CPU TIME	SRB CPU time. Time is included in TOTAL_CPU TIME.
DATABASE_CPU TIME	Total database CPU time that is used by IMS and DB2. This field does not reflect CPU usage by RPCs that access IMS and DB2. CPU time may also be accounted for in the Enclave CPU time.
NETWORK_CPU TIME	<p>Total network CPU time trappable within Data Virtualization Manager server. CPU time may also be accounted for in the Enclave CPU time.</p> <p>Note: Network CPU time cannot be trapped within Data Virtualization Manager server. Therefore, this field may not reflect the total CPU time.</p>
REXX_CPU TIME	Total CPU time that is used by running SEF REXX programs. CPU time may also be accounted for in the Enclave CPU time.
USER_PGM_CPU TIME	Total CPU time that is used by RPCs. This time includes CPU time that is used in DB2 and IMS if the RPCs accessed these databases. CPU time may also be accounted for in the Enclave CPU time.

Column	Description
SSL_CPUTIME	Total CPU time that is used in processing SSL routines. CPU time may also be accounted for in the Enclave CPU time.
ENCLAVE_CPUTIME	Total CPU time that is associated with the WLM enclave. Note: Other CPU times in this record may also be associated with the Enclave CPU time. Also, other tasks may have contributed CPU time to the same Enclave.
ZIIP_QUALIFIED	Enclave zIIP qualified CPU time.
ZIIP_CPU	Enclave zIIP CPU time.
ZIIP_ON_CP	Enclave zIIP time on CP.
OTHER_CPUTIME	All of the unaccountable CPU time. CPU time may also be accounted for in the Enclave CPU time.
CONNECTION_ID	The unique Data Virtualization Manager server connection ID.
GMT_LOGON	Time the user logged on in Greenwich Meridian Time.
LOGON_TIME	Time the user logged on.
CONNECT_TIME	The number of seconds the user was connected.
INTERVAL_START	The adjusted interval start time.
BYTES_READ	Total bytes read.
BYTES_WRITTEN	Total bytes written.
HTTP_STATUS	The HTTP Status code that is returned for this request.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The name of the IBM Data Virtualization Manager for z/OS subsystem ID.
SERVER_CODE	Overall return code for the transaction.
SERVER_REASON	Associated reason code, if any.
SERVER_ABEND	Abend code for the transaction, if one occurred.
IP_ADDRESS	For TCP/IP connections, this is the IP address of the client workstation; otherwise, this value is null.
PORT	TCP/IP port number for this connection.
VIRTUAL_DIRECTORY	Alternate name (50 characters max) used to uniquely identify a virtual directory.
WEB_SERVICE	The name of the Web Service invoked.
NAME_SPACE	The namespace of the Web Service invoked.
OPERATION	The Operation of the Web Service invoked.
TARGET_SYSTEM	The name of the Target System used to process this request.
SOAP_FAULT	Some of the text of the Soap Fault message generated, if any.

Record: Streams

This record is used to write one SMF records for each Streams task, during each interval.

About this task

A 'Final' row is written covering a partial interval when a Streams task terminates. These can be distinguished by the SM19RCTY field of the SMF record, or the RECORD_TYPE column of the table. Some columns only apply to source tasks, and some columns apply only to destination tasks, so the remaining columns are null in any given row.

Procedure

Use the MODIFY PARM command to set the following parameters in the hlq.SAVZEXEC(AVZSIN00) member:

```
"MODIFY PARM NAME(LOGPUBINTERVALSTABLE) VALUE(AVZ.STREAMS)"  
"MODIFY PARM NAME(LOGPUBINTERVALS) VALUE(YES)"  
"MODIFY PARM NAME(LOGRETAINPUB) VALUE(30)"
```

The following table lists the parameters used to configure the Streams record:

Parameter	Description	Valid values
LOGPUBINTERVALS	Enables logging of Streams interval information. If interval recording is active, a row is inserted into a table at the end of each recording interval,	YES NO Default value is NO.
LOGPUBINTERVALSTABLE	Sets the name of the DB2 table that is used to log Streams interval information. If interval recording is active, a row is inserted into this table at the end of each recording interval.	'AVZ.STREAMS'
LOGRETAINPUB	Controls the number of days to wait before automatically deleting rows from the Streams log table. That is, all rows older than the number of days are deleted. If this value is zero, then rows are never automatically deleted from the Streams log table.	Number of days. Default is 0.

AVZ.STREAMS

The AVZ.STREAMS is used for capturing statistics for Streams usage.

Column	Description
RECORD_TYPE	Record type, either Interval record or Final Interval record (possibly a partial interval).
PUBLISH_TYPE	Record is for a Source or a Destination.

Column	Description
TASK_TYPE	Task type, either DB2 source, IMS source, CICS source, Adabas source, IDMS source, VSAM source, HTTP destination, MQ destination, or MQ Broker destination.
TASK_NAME	The name that is given to the task.
TOTAL_CPUETIME	Total CPU time. CPU time may also be accounted for in the Enclave CPU time.
SRB_CPUETIME	SRB CPU time. Time is included in TOTAL_CPUETIME.
DATABASE_CPUETIME	Total database CPU time that is used by IMS and DB2. Some CPU time may also be accounted for in the Enclave CPU time.
NETWORK_CPUETIME	Total network CPU time trappable within Data Virtualization Manager server. CPU time may also be accounted for in the Enclave CPU time. Note: Network CPU time cannot be trapped within Data Virtualization Manager server. Therefore, this field may not reflect the total CPU time.
REXX_CPUETIME	Total CPU time that is used by running SEF REXX programs. CPU time may also be accounted for in the Enclave CPU time.
SSL_CPUETIME	SSL processing CPU time.
ENCLAVE_CPUETIME	Total CPU time that is associated with the WLM enclave. Note: Other CPU times in this record may also be associated with the Enclave CPU time. Also, other tasks may have contributed CPU time to the same Enclave.
ZIIP_QUALIFIED	Enclave zIIP qualified CPU time.
ZIIP_CPU	Enclave zIIP CPU time.
ZIIP_ON_CP	Enclave zIIP time on CP.
OTHER_CPUETIME	All of the unaccountable CPU time. CPU time may also be accounted for in the Enclave CPU time.
SMFID	The SMFID as defined within Data Virtualization Manager server.
PRODUCT_SUBSYSTEM	The name of the IBM Data Virtualization Manager for z/OS subsystem ID.
INTERVAL_START	Interval start time.
ELAPSED_TIME	The length of the interval.
BYTES_WRITTEN	The number of bytes written to TCP/IP or MQ.
SO_EVENTS_CAPTURED	The number of events the source task captured.
SO_EVENTS_IGNORED	The number of events the source task was told to ignore.
SO_RULES_RUN	The number of times a source task ran a rule.
SO_RULES_FAILED	The number of times a source task ran a rule that failed.
SO_EVENTS_QUEUED	The number of events queued to a destination task.
SO_BYTES_CAPTURED	The number of events the source task captured.

Column	Description
SO_BYTES_QUEUED	The number of bytes in internal format that a source task queued to a destination.
DE_EVENTS_READ	The number of events that a destination task read.
DE_EVENTS_SHIPPED	The number of events that a destination task successfully shipped.
DE_BYTES_SHIPPED	The number of bytes in internal format that a destination task successfully shipped.
DE_EVENTS_FAILED	The number of events that a destination task had permanent failures.
DE_CONNECTIONS	The number of connections that a destination task established.
DE_RETRIABLE_FAILS	The number of events that a destination task had re-triable failures.

Chapter 15. Sysplex Function

Sysplex function in the IBM Data Virtualization Manager for z/OS ISPF menu helps you access different Data Virtualization Manager server, execute queries and modify various parameters across the subsystems running on different LPARs within the same Sysplex.

Currently you can access different IBM Data Virtualization Manager for z/OS servers running on same LPAR. Using Sysplex function you can access different IBM Data Virtualization Manager for z/OS servers running on the same Sysplex.

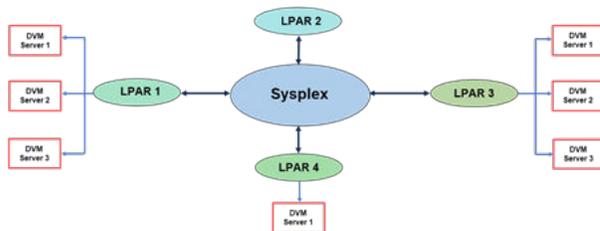


Figure 1. Sysplex Function

In the figure, the Sysplex has four LPARs and each LPAR is connected to multiple IBM Data Virtualization Manager for z/OS Servers. A change to a parameter or a setting on one server is replicated across the selected servers that are attached in a Sysplex.

Defining and enabling Sysplex Function

To enable Sysplex Function, you must modify the parameter **SYSPLEX** in IN00 configuration file.

Before you begin

Before defining Sysplex and SIS/XCF configuration in the same Server, ensure that the **PLEXID** in Sysplex Definition and **SISID** in SIS/XCF definition are different.

About this task

Follow the steps in this section to enable and define multiple sub-systems in a Sysplex group.

Procedure

1. Use the `MODIFY PARM` command to set the **SYSPLEX** parameter to YES as:

```
MODIFY PARM NAME(SYSplex) VALUE(YES)
```

2. Add the following `DEFINE` statement located in the AVZSIN00 configuration member:

```
"DEFINE SYSplex",
"CONNID(AVZS)",
"PLEXID(AVZS)",
"PLEXGRP(PLEXAVZS)"
```

Note: Add this definition in the configuration members of all the servers that are part of the Sysplex Functions feature.

Accessing Sysplex Function

After enabling and defining the Sysplex Function you can access the feature from the primary menu option.

About this task

Use the Sysplex Functions option is available under IBM Data Virtualization Manager for z/OS's Primary Option Menu to view various functions available in the Sysplex function.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, enter option **S** to select **Sysplex Function** and press **ENTER**. The *Sysplex Command Scope selection* screen appears.
2. Type one of the following line commands, and press **ENTER**:

Command	Description
1	AVZ Select
2	AVZ Execute

AVZ Select

Use this option to list all the IBM Data Virtualization Manager for z/OS sub-systems belonging to the same Sysplex group as defined in the IN00 configuration file.

About this task

On selecting this option the member name, name of the sub-systems, and whether the subsystem is in service or not is displayed. You can select any IBM Data Virtualization Manager for z/OS sub-system as per the requirement.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select **Sysplex Function** and press **ENTER**. The *Sysplex Command Scope Selection* screen appears.
2. Enter option **1** to select AVZ **select** and press **ENTER**. A list of subsystems running appears.
3. Under **Sel** column, enter Y to select the sub-system.
Enter N if you do not want to select the subsystem.

Note:

- If you enter Y against an active server, the changes made to the default server parameters through the Sysplex Function will reflect in the selected sub-system.
- The **In Svc** column displays whether the server is currently active or not. You can only access or make changes to the active sub-systems.
- The first server appearing in the list is current or active server, the value in the **Sel** column is always **D** for this server as it is the default server and the corresponding **In Svc** value will be **Yes**.

AVZ Execute

Use this option to modify the parameters for all the selected sub-system running on the same Sysplex.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select **Sysplex Function** and press **ENTER**.

The *Sysplex Command Scope Selection* screen appears.

2. Enter option **2** for AVZ **Execute** and press **ENTER**.

The *Sysplex Command Option Selection* screen appears.

3. Type one of the following line commands, and press **ENTER**:

Command	Description
C	AVZ Admin
D	Rules Management

AVZ Admin

You can manage the started task parameters for IBM Data Virtualization Manager for z/OS.

About this task

Changes made to the started task parameters associated with the current IBM Data Virtualization Manager for z/OS sub-system will reflect in all the selected sub-systems in the Sysplex group.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select **Sysplex Function** and press **ENTER**. The *Sysplex Command Scope Selection* screen appears.
2. Enter option **2** for AVZ **Execute** and press **ENTER**. The *Sysplex Command Option Selection* screen appears.
3. Enter option **C** for AVZ **Admin** and press **ENTER**.
4. Enter option **2** for AVZ **Parms** and press **ENTER**. A list of Parameter Group appears.
5. Scroll down and find the required parameter group, to the left of the parameter group, type **D** and press **ENTER**.
All the parameters and their default values in the selected parameter group appears.
6. Scroll down and find the required parameter, to the left of any particular parameter, type **D** to display more information.
The **Parameters** panel with more information for the selected parameter appears.
7. Under **Parameter Value** column, enter a value that needs to be changed.
8. Use the **END** command, or press the **F3** key, to return to the **Parameters** panel.
The change appears across the selected IBM Data Virtualization Manager for z/OS sub-systems.
For more information on viewing and editing the parameters, see [“Viewing Details about a Parameter” on page 3](#).

Rules Management

This option displays detailed SEF rule for the current IBM Data Virtualization Manager for z/OS sub-system.

Before you begin

For using this option all the participating IBM Data Virtualization Manager for z/OS sub-systems must be using the same SEF rule datasets.

About this task

You can modify, enable, disable and reset auto-enable SEF rules under Rules Management option in Sysplex Function. If any changes are made to a IBM Data Virtualization Manager for z/OS sub-system, the changes will reflect in all the selected sub-systems instantly.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select **Sysplex Function** and press **ENTER**. The *Sysplex Command Scope Selection* screen appears.
2. Enter option **2** to select AVZ **Execute** and press **ENTER**. The *Sysplex Command Option Selection* screen appears.
3. Enter option **E** for **Rules Mgmt.** and press **ENTER**. The *Event Facility (SEF) Control* screen appears.
4. Enter option **2** for **SEF Rule Management**.
5. Enter **VTB** against **Display Only the Ruleset Named** to view all rulesets. A list of available VTB rules appear.

You can enable or disable the VTBs. For more information on rules and events, see [Chapter 4, “Rules and events,”](#) on page 31.

Chapter 16. Monitoring

Data Virtualization Manager server provides powerful diagnostic tools that can record critical events for individual transactions. This information can be used to diagnose, debug, and correct problems.

Data Virtualization Manager server provides the following trace options:

- Server Trace
- Instrumentation Server (IS)
- Server Trace Archival Facility
- SQL Tracing

Server Trace

The Server Trace adds Data Virtualization Manager server trace records to a trace buffer maintained in virtual storage. When the session is finished, the trace records are automatically saved in a VSAM data set.

Trace records are written for the following actions:

- SQL operations
- IMS calls
- CICS calls
- Communication events (LU 6.2, TCP/IP, and messages)
- Thread attach and detach events
- Remote Procedure Call (RPC) events
- Message events
- Errors (abends)

A Remote Procedure Call (RPC) can add its own trace messages to the trace for diagnostic purposes.

Using Trace Browse, you can perform the following actions:

- Display formatted columns of information, such as user ID and time
- Use FIND and LOCATE commands to search for data or a specific time and date
- Use the DISPLAY command to display additional columns of information
- Use the STATUS command to display the Trace Browse status area

In general, the Server Trace can accommodate the complete record of all client/server processing for several days. However, using hierarchical storage management, you can maintain an unlimited history of data. The Server Trace data collection routines support collection of all the data required for auditing, capacity planning, and trend analysis of usage patterns. You can set security for the Server Trace filter functionality to prohibit viewing of sensitive data by a non-authorized user.

Instrumentation Server

Using the Instrumentation Server (IS), you can run multiple instances of the server in a sysplex and route trace information to a single repository so that you have a global view of all activity.

Server Trace Archival Facility

Use the Server Trace Archival Facility to back up, or archive, active trace information. The archive consists of a large block of virtual storage, which can be backed up by a data-in-virtual (DIV) linear data set. This block of virtual storage is sub-divided into the following parts:

- The status area occupies the first 4 KB page of the virtual storage and contains checkpoint information about the trace area and information about the most recent trace archive.
- Event blocks begin in the second 4 KB page of the virtual storage area. Each event block occupies 896 bytes of storage. Each server event is recorded in the next available slot, beginning with the first slot, continuing to the end of the event blocks, and wrapping around to the beginning of the event block.
- Vector tables each begin on a 4 KB page boundary, and are located after the event blocks in the trace storage. Each vector table contains index information that allows views of the trace to be filtered without searching through the entire virtual storage area occupied by each individual event block.

SQL Trace

The SQL Trace program provides details about all of the SQL statements that applications issue. The information that is displayed in the SQL Trace program is derived from the main SDB log by using connection IDs as the selection criterion.

When you select an active session, the SQL Trace displays the current information. To refresh the information, press `Enter`.

Monitoring Data Virtualization Manager client response time

Client response time is the time between when a query starts on the Data Virtualization Manager server and when data is returned to the application.

Procedure

1. Use the `MODIFY PARM` command to add the following parameter that is located in the `AVZSIN00` configuration member:

```
"MODIFY PARM NAME(MONRESPONSETIME) VALUE(YES)"
```

The following table lists the parameter for configuring response time monitoring:

Parameter	Description	Valid values
MONRESPONSETIME	Controls whether to monitor the client response time for applications that are defined by the <code>RTMONAPP</code> parameter in the <code>AVZSIN00</code> configuration file. SMF Subtype 14: Client Response Time Records	YES Monitoring occurs. NO Default value is NO.

2. For each application that you want to monitor, add the following `DEFINE` statement to the `AVZSIN00` configuration member:

```
"DEFINE RTMONAPP APPLICATION(appname)", "TIME(time)"
```

Where:

- *appname* is the application name, internal name, or module name.
 - *time* is the response-time goal, in milliseconds.
3. Restart the Data Virtualization Manager server so that the changes take effect.

Monitoring streams with Server Trace

You can turn on tracing in Server Trace, but typically you do not need to refer to it unless instructed to by Technical Support.

Procedure

Enable tracing in Server Trace by using the `MODIFY PARM` command to set the following parameters that are located in the Data Virtualization Manager configuration member, `AVZSIN00`:

```
"MODIFY PARM NAME(TRACEPUBLISH) VALUE(YES)"
"MODIFY PARM NAME(TRACEPUBLISHCAPTURE) VALUE(YES)"
"MODIFY PARM NAME(TRACEPUBLISHDATA) VALUE(NO)"
"MODIFY PARM NAME(TRACEPUBLISHEVENTIO) VALUE(NO)"
"MODIFY PARM NAME(TRACEPUBLISHFLOW) VALUE(YES)"
"MODIFY PARM NAME(TRACEPUBLISHWORKIO) VALUE(NO)"
```

The following table lists the parameters for configuring basic Trace Browse support:

Parameter	Description	Valid values
TRACEPUBLISH	Controls tracing of Streams servers.	YES (default) All calls are traced. NO
TRACEPUBLISHCAPTURE	Controls tracing of the Streams event capturing.	YES NO (default) Streams events tracing is disabled.
TRACEPUBLISHDATA	Controls whether the full publish data for publish events is traced.	YES NO (default) Full publish data is not traced..
TRACEPUBLISHEVENTIO	<i>(Non-DB2 users only)</i> Controls tracing of the Streams input/output to its event capture files.	YES NO (default) Input/output events to the event capture database are not traced.
TRACEPUBLISHFLOW	Controls tracing of the Streams module flow.	YES (default) The module flow is traced. NO
TRACEPUBLISHWORKIO	Controls tracing of the Streams input/output to its work files.	YES NO (default) Input/output to the Streams work file is not traced.

Instrumentation Server

The Instrumentation Server presents a global view of the Trace Browse facility by running many servers in a single logical partition (LPAR) or across a sysplex, and routing Trace Browse information to one repository. This is accomplished using Cross System Coupling Facility (XCF) services.

When multiple servers run in a single LPAR, XCF converts the call to a z/OS cross memory call so that XCF is not used.

The SIS ERRORONLY server limits the trace data being sent to the Instrumentation Server to contain only 'significant' information. This means that the IS Trace Browse wraps less frequently and is more useful. You use the SIS ERRORONLY server when you only want to trace error messages or if you are running in a high-volume environment.

The Instrumentation Server is not recommended for a high-volume environment where large amounts of tracing is occurring. The Instrumentation Server address space increases CPU usage by your IBM Data Virtualization Manager for z/OS environment based on the number of messages being processed. CPU increases could range around .0003 seconds of CPU per Trace Browse message. However, this number is relative to the type and speed of the processor where the IBM Data Virtualization Manager for z/OS subsystems are installed and should not be used as a definitive calculation. The Instrumentation Server may also affect transaction response time relative to the number of messages that are generated per transaction and the extra CPU costs for those messages.

Instrumentation Server benefits

There are many benefits to using the Instrumentation Server:

- The Instrumentation Server decreases the virtual storage requirements for the main Data Virtualization Manager server address spaces servicing transaction requests by removing the storage requirements for Trace Browse.
- The Instrumentation Server provides a global view of tracing activity for all your Data Virtualization Manager server instances in a single z/OS LPAR or across a sysplex into a single Instrumentation Server.
- The Instrumentation Server provides a larger trace data set to be used than the main Data Virtualization Manager server address space. This also allows for more trace data to remain in the Trace Browse display.
- You may also want to have one IS per IBM Data Virtualization Manager for z/OS subsystem to take advantage of the preceding first and third bullets.

Reducing the amount of tracing

Before you implement the Instrumentation Server, you should review your current Data Virtualization Manager server tracing values and reduce unnecessary tracing parameters to minimize the rate that messages are sent to the Instrumentation Server.

Procedure

To reduce the amount of tracing for a Data Virtualization Manager server and still benefit from tracing, use these recommendations:

- If you are doing a new install using the sample Data Virtualization Manager configuration member, AVZSIN00, shipped with the Data Virtualization Manager server distribution, ensure that all trace parameters are disabled by setting each parameter to NO. If you have trace options that have been added to AVZSIN00 from an existing installation and that are not included in the distributed sample AVZSIN00, then comment these out to use the default values.
- Consider disabling the following parameters by adding them to the Data Virtualization Manager configuration member, AVZSIN00:
 - TRACEIBMOEVENTS disables all TCP/IP tracing events.

- TRACETEXTEVENTS disables various informational messages that are associated with other trace events. This includes more logon messages, DB2 thread token value messages, and DB2 DBRM tracing messages.
- TRACESQMEVENTS disables the IBM Data Virtualization Manager for z/OS logging tracing used for logging activity in a set of predefined logging tables.
- TRACERRSEVENTS disables detailed RRS messages when using either the DB2 RRS interface or RRS two-phase commit support with any supported interface.
- TRACERPCEVENTS disables the tracing of the IBM Data Virtualization Manager for z/OS ODBC CALL RPC APIs. If you disable default trace options, be aware that Technical Support may require certain disabled trace options to be re-enabled to assist with any reported issues.

Installing the Instrumentation Server

To use either the Instrumentation Server or Instrumentation Server ERRORONLY, create a separate address space.

About this task

The sole responsibility of this address space is to act as the Instrumentation Server manager. Do not use this separate IS address space to run any client transactions other than IBM Data Virtualization Manager for z/OS Administrator functions. Use job AVZGNSIS located in the *hlq*.SAVZCNTL data set to help in performing the following steps 1 and 2. To install IS and IS ERRORONLY, take the following steps:

Procedure

1. Create a new VSAM data set for the Instrumentation Server address space.
Increase the size of the Trace Browse data set to a minimum of 1750 cylinders for a 3390 device. This enables up to one million lines of tracing to be maintained in a single viewable Trace Browse.
2. Use job AVZGNSIS located in the *hlq*.SAVZCNTL data set to copy the following data sets:

```
hlq.ATH.SAVZEXEC
hlq.CMD.SAVZEXEC
hlq.EXC.SAVZEXEC
hlq.GLV.SAVZEXEC
hlq.PUB.SAVZEXEC
hlq.RPC.SAVZEXEC
hlq.SQL.SAVZEXEC
hlq.TOD.SAVZEXEC
```

3. Create a new startup JCL procedure for the Instrumentation Server address space. The AVZS member of the *hlq*.SAVZCNTL library contains sample JCL procedures for running the main address space (started task) as an Instrumentation Server address space.
Add the AVZS PROC to the SYS1.PROCLIB or to another procedure library. You can change the name of the procedure to reflect the Instrumentation Server, and you can change the **SSID** parameter in the startup procedure to reflect another valid subsystem name.
4. Define the new started task to the security product.
5. Depending on the communication protocol that you use, obtain or define a new TCP/IP port or VTAM application ID.
6. Create a new Data Virtualization Manager configuration member for the Instrumentation Server address space. The Data Virtualization Manager configuration member is a REXX program that is used to set product parameters. A sample Data Virtualization Manager configuration member, AVZSIN00, is shipped with the product and can be customized to configure your Instrumentation Server address space. If you use a IBM Data Virtualization Manager for z/OS subsystem name other than AVZS for Instrumentation Server, be sure to rename this member to include the first four characters of the subsystem name.

The sample AVZSIN00 contains only the required parameters for the Instrumentation Server. Make the following changes to the SIS AVZSIN00 configuration member:

- a) Define the TCP/IP Port number to be used.
- b) Set up the SEF data sets.
- c) Set up the data set definitions.
- d) Update the **BROWSEMAX** parameter to match the desired size of the retained Trace Browse. You need 175 cylinders per 100000 Trace Browse messages.
- e) Define the Instrumentation Server parameters, as follows:

- Use the **MODIFY PARM** command to set the following parameter that is located in the Data Virtualization Manager configuration member, AVZSIN00:

```
"MODIFY PARM NAME(SIS/XCF) VALUE(YES)"
```

- For each Data Virtualization Manager server you connect to the Instrumentation Server, add the following DEFINE statement that is located in the AVZSIN00 configuration member:

```
DEFINE SISXCF
  "CONNID(SYS1AVZS)",
  "SISID(AVZS)",
  "SISXCFGRP(SISAVZS)",
  "ERRORONLY(YES)", (for SIS ERRORONLY server)
  "MANAGER(NO)"
```

- f) Make the following changes to each Data Virtualization Manager configuration member, AVZSIN00 that is connected to and sending Trace Browse messages to the Instrumentation Server. Use the **MODIFY PARM** command to set the following parameter:

```
"MODIFY PARM NAME(SIS/XCF)VALUE(YES)"
```

Note: Only one Data Virtualization Manager server can send data on a particular SISID-SISXCFGRP pair to an Instrumentation Server manager. This combination must be unique in the sysplex. For the Instrumentation Server manager, the CONNID must be unique in that Data Virtualization Manager server.

7. Create a AVZSINEF member in the SYSEXEC concatenation, where AVZS refers to the new Instrumentation Server subsystem name you are creating. A sample AVZSINEF is included in the distributed *hlq.SAVZEXEC* data set.
8. Start the Instrumentation Server address space.
9. Recycle all the Only one IBM Data Virtualization Manager for z/OS Servers connected to the Instrumentation Server address space. After these are recycled, view the transmitted Trace Browse messages from any Only one IBM Data Virtualization Manager for z/OS Server ISPF application by specifying the Instrumentation Server subsystem name in the SIS SSID field on the Only one IBM Data Virtualization Manager for z/OS Server Primary Option Menu.

Using the Instrumentation Server in a sysplex

The Cross System Coupling Facility (XCF) handles communication between Logical Partitions (LPARs) in a sysplex and is only used when setting up the Instrumentation Server to connect IBM Data Virtualization Manager for z/OS Servers and the Instrumentation Server across members of a sysplex. Information such as workload, status, and data transmission can be passed through the coupling facility. The information sharing is constant and continuous, allowing the independent z/OS images to know detailed information about the status of all images in the sysplex.

About this task

To use this most effectively, size the coupling facility for the Instrumentation Server that is based on the number of expected messages that are processed by the coupling facility. The sizing depends the Instrumentation Server's ability to read messages out of the coupling facility as the other IBM Data Virtualization Manager for z/OS Servers send messages to the coupling facility. If using the Instrumentation Server in a sysplex environment, you might have to resize the coupling facility to handle the increase in traffic for the Data Virtualization Manager server trace messages.

Use the IBM Resource Measurement Facility (RMF) to monitor the number of rejected messages in the XCF reports. If the reports contain many rejected messages, increase the size of the buffers. For more information about SCF reports, see [Parallel Sysplex performance: XCF performance considerations at Parallel Sysplex Performance: XCF Performance Considerations](#).

There are no required structure definitions for the IS coupling facility. The Instrumentation Server uses the XCF signaling services, which use the paths that are defined in the COUPLEXX member of the system PARMLIB. These can be CTCs or CF structures.

If the member of the sysplex that hosts the Instrumentation Server terminates, all the IBM Data Virtualization Manager for z/OS Servers that are clients of the Instrumentation Server begin tracing locally. After the member of the sysplex and the Instrumentation Server are restarted, the IBM Data Virtualization Manager for z/OS Servers begin transmitting Trace Browse messages. Trace Browse messages that were written locally are not sent to the Instrumentation Server.

Monitoring and managing RRS transactions

IBM Data Virtualization Manager for z/OS Enterprise Transactions is a licensed add-on component of the IBM Data Virtualization Manager for z/OS product suite. This component supports the monitoring and management tasks of RRS (Resource Recovery Services) transactions.

With the RRS monitor and control options, you can use the Data Virtualization Manager server RRS manager to view and manage all in-progress two-phase commit protocol transactions that are managed by IBM Data Virtualization Manager for z/OS.

Note: IBM Data Virtualization Manager for z/OS Enterprise Transaction was designed and written to the Open Group XA-Distributed Transaction Protocol specification.

RRS Manager display

The Resource Manager program provides information about the Data Virtualization Manager server RRS Resource Manager. RRS is a z/OS component that uses two-phase commit protocol to manage transaction processing across multiple data sources. When two-phase commit support is enabled, IBM Data Virtualization Manager for z/OS registers its Resource Manager with RRS. The Resource Manager must be connected to RRS for two-phase commit transaction processing to occur.

Enabling two-phase commit transaction processing

When two-phase commit support is enabled, the Data Virtualization Manager Server registers its Resource Manager with RRS. The Resource Manager must be up and connected to RRS for two-phase commit transaction processing to occur.

About this task

To invoke RRS Manager information:

Procedure

1. From the Primary Option Menu, select AVZ **Admin** and press Enter.
2. From the **Server Management** menu, select **RRS** and press Enter.
3. From the **Server RRS Monitor** menu, select **Resource Manager** and press Enter.
4. Use the available line commands that are described in the following section to perform the appropriate functions.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
D	Disables the Resource Manager.
E	Enables the Resource Manager.
F	Formats the information.
P	Prints the control block.
S	Displays the control block for the selected row.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description
RRS RESOURCE MANAGER NAME	The name of the RRS Resource Manager.
RRS STATUS	The status of the RRS Resource Manager program. Valid values are: <ul style="list-style-type: none"> • ACTIVE • DOWN • NO RRS (the RRS parameter is not been selected).
STATUS	The status of the RRS connection. <ul style="list-style-type: none"> • ENABLED allows normal transaction processing. • DISABLED means new RRS transactions are prevented from starting.
TRANSACTIONS STARTED	The number of transactions started successfully.
COMMITTS NORMAL	The number of transactions committed normally.
COMMITTS RECOVERY	The number of transactions committed by using the XA recover command.
COMMITTS PANEL	The number of transactions committed manually.
ROLLBACKS NORMAL	The number of transactions rolled back normally.
ROLLBACKS RECOVERY	The number of transactions rolled back by using the XA recover command.
ROLLBACKS PANEL	The number of transactions rolled back manually.

Viewing active two-phase commit transactions

The Data Virtualization Manager server Active Transaction Control program allows you to view all active RRS transactions that are running in IBM Data Virtualization Manager for z/OS.

About this task

To invoke active RRS transaction control information:

Procedure

1. From the Primary Option Menu, select AVZ **Admin** and press Enter.

2. From the **Server Management** menu, select **RRS** and press Enter.
3. Select **Active Transactions** from the Data Virtualization Manager server **RRS Monitor** menu and press Enter.

Three panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

4. Use the available line commands that are described in the following section to perform the appropriate functions.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
F	Formats the information for the selected row.
P	Prints the control block for the selected row.
S	Displays the control block for the selected row.
T	Displays the Server Trace data that is related to this RRS XID.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
GTRID LENGTH	The length of the XA global transaction ID.	
GLOBAL TRAN ID	The first 32 bytes of the XA global transaction ID.	GTRID
TRXN BEGIN TIME	The date and time when the transaction began running.	START
CLIENT USERID	The user ID passed by the client.	USERID
RRS STATE	The state of the transaction according to RRS.	TMTYPE
TRANSACTION TYPE	The type of transaction manager on the client side that is coordinating this transaction: MTS.	TMTYPE
NUMBER OF THREADS	The number of z/OS threads (1 - 8) participating in the transaction.	TMTYPE
RRS URID	The RRS-assigned Unit of Recovery (UR) ID for the first or only thread of this transaction. It can be used to correlate this transaction with an RRS UR.	URID

Column name	Description	Sort name
XID TOKEN	The assigned token that is associated with this transaction. It can be used with PROFILE and DISPLAY in the Server Trace Facility.	XTOKEN
BQUAL LENGTH	The length of the XA branch qualifier.	XTOKEN
BQUAL VALUE	The first 32 bytes of the XA branch qualifier value, up until the last valid byte.	XTOKEN
GTRID 2ND HALF	The second 32 bytes of the XA global transaction ID, up until the last valid byte.	XTOKEN

Viewing indoubt two-phase commit transactions

The Indoubt Transaction program displays RRS transactions that are in the indoubt state and allows the user to commit or rollback these transactions.

About this task



Warning: The RRS transactions that run under IBM Data Virtualization Manager for z/OS on z/OS can be one part of a larger transaction that is coordinated by the client side transaction manager (MTS). Issuing a COMMIT or ROLLBACK could leave the overall transaction, as well as its data, in an inconsistent state. Extreme care must be used with these commands.

To invoke indoubt RRS transaction information:

Procedure

1. From the Primary Option Menu, select AVZ **Admin** and press Enter.
2. From the **Server Management** menu, select **RRS** and press Enter.
3. Select **Indoubt Transactions** from the **RRS Monitor** menu and press Enter.

Three panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

4. Use the available line commands that are described in the following section to perform the appropriate functions.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
C	Commits the transaction (see warning).
F	Formats the information for the selected row.
P	Prints the control block for the selected row.
R	Commits the transaction (see warning).
S	Displays the control block for the selected row.

Line commands	Description
T	Displays the Server Trace data that is related to this RRS XID.



Warning: Using the C or R line commands can leave the overall client transaction and the data in an inconsistent state.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
GLOBAL TRAN ID	The XA global transaction ID assigned by the client-side transaction manager.	GTRID
TRXN BEGIN TIME	The date and time when the transaction began running on IBM Data Virtualization Manager for z/OS on the MVS system.	START
CLIENT USERID	The user ID passed by the client.	USERID
RRS STATE	The state of the transaction according to RRS.	USERID
TRANSACTION TYPE	The type of transaction manager on the client side that is coordinating this transaction: MVS.	TMTYPE
NUMBER OF THREADS	The number of MVS (1 - 8) threads participating in the transaction.	TMTYPE
RRS URID	The RRS-assigned Unit of Recovery (UR) ID for the first or only thread of this transaction.	URID
XID TOKEN	The token that is associated with this transaction. It can be used with PROFILE and DISPLAY in the Server Trace Facility.	XTOKEN
BQUAL LENGTH	The length of the XA branch qualifier.	XTOKEN
BQUAL VALUE	The first 32 bytes of the XA branch qualifier value, up until the last valid byte.	XTOKEN
GTRID 2ND HALF	The second 32 bytes of the XA global transaction ID, up until the last valid byte.	XTOKEN

Displaying information about failed two-phase commit transactions

The Recovery Table program displays RRS transactions that are stored in the RRS recovery table because of a failure while the transaction was in progress.

About this task

To invoke the recovery table display:

Procedure

1. From the Primary Option Menu, select AVZ **Admin** and press Enter.
2. From the **Server Management** menu, select **RRS** and press Enter.
3. Select **Recovery Table** from the **RRS Monitor** menu and press Enter.

Three panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

4. Use the available line commands that are described in the following section to perform the appropriate functions.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
F	Formats the information for the selected row.
P	Prints the control block for the selected row.
S	Displays the control block for the selected row.
T	Displays the Trace Browse data that is related to this RRS XID.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
GLOBAL TRAN ID	The XA global transaction ID assigned by the client-side transaction manager.	GTRID
TRXN BEGIN TIME	The date and time when the transaction began running.	START
CLIENT USERID	The user ID passed by the client.	USERID
RRS STATE	The state of the transaction according to RRS.	TMTYPE
TRANSACTION TYPE	The type of transaction manager on the client side that is coordinating this transaction: MTS.	TMTYPE

Column name	Description	Sort name
NUMBER OF THREADS	The number of MVS (1 - 8) threads participating in the transaction.	TMTYPE
RRS URID	The RRS-assigned Unit of Recovery (UR) ID for the first or only thread of this transaction.	URID
XID TOKEN	The token that is associated with this transaction. It can be used with PROFILE and DISPLAY in Server Trace.	XTOKEN
BQUAL LENGTH	The length of the XA branch qualifier.	XTOKEN
BQUAL VALUE	The first 32 bytes of the XA branch qualifier value, up until the last valid byte.	XTOKEN
GTRID 2ND HALF	The second 32 bytes of the XA global transaction ID, up until the last valid byte.	XTOKEN

Invoking the RRS Units of recovery information

The Units of Recovery program displays the RRS Units of Recovery (URs) associated with this instance of the Data Virtualization Manager server.

About this task

To invoke the Units of Recovery program:

Procedure

1. From the Primary Option Menu, select AVZ **Admin** and press Enter.
2. From the Server Management Menu, select **RRS** and press Enter.
3. Select **Unit of Recovery** from the **RRS Monitor** menu and press Enter.

Three panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

4. Use the available line commands that are described in the next section to perform the appropriate functions.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
F	Formats the information for the selected row.
P	Prints the control block for the selected row.
S	Displays the control block for the selected row.

Line commands	Description
T	Displays the Server Trace data that is related to this RRS XID.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
GLOBAL TRAN ID	The XA global transaction ID assigned by the client-side transaction manager.	GTRID
TRXN BEGIN TIME	The date and time when the transaction began.	START
CLIENT USERID	The user ID passed by the client.	USERID
RRS STATE	The state of the transaction according to RRS.	USERID
TRANSACTION TYPE	The type of transaction manager on the client side that is coordinating this transaction: MTS.	TMTYPE
NUMBER OF THREADS	The number of MVS (1 - 8) threads participating in the transaction.	TMTYPE
RRS URID	The RRS-assigned Unit of Recovery (UR) ID for the first or only thread of this transaction.	URID
XID TOKEN	The token that is associated with this transaction. It can be used with PROFILE and DISPLAY in Server Trace.	URID
BQUAL LENGTH	The length of the XA branch qualifier.	URID
BQUAL VALUE	The first 32 bytes of the XA branch qualifier value, up until the last valid byte.	URID
GTRID 2ND HALF	The second 32 bytes of the XA global transaction ID, up until the last valid byte.	URID
TCB ADDRESS	The initial TCB on which the UR ran. (Zero if the UR was from the recovery table.)	TCB
VCID	The VCID value that is assigned when this UR started.	VCID
PREPARE RET CODE	The return code from the RRS PREPARE operation. (N/A if the PREPARE operation is not done).	VCID

Column name	Description	Sort name
COMMIT RET CODE	The return code from the RRS COMMIT operation. (N/A if the RRS COMMIT operation is not done).	VCID
ROLLBACK RET CODE	The return code from the RRS ROLLBACK operation. (N/A if the RRS ROLLBACK operation is not done).	VCID
FORGET RET CODE	The return code from the RRS FORGET operation. (N/A if the RRS FORGET operation is not done).	VCID
TRANSACTION FLAG 1	Transaction flag 1. Used for diagnostics.	VCID
TRANSACTION FLAG 2	Transaction flag 2. Used for diagnostics.	VCID
TRANSACTION FLAG 3	Transaction flag 3. Used for diagnostics.	VCID
TRANSACTION FLAG 4	Transaction flag 4. Used for diagnostics.	VCID
TRANSACTION FLAG 5	Transaction flag 5. Used for diagnostics.	VCID
DIAGNOSTIC FLAG 1	Diagnostic flag 1. Used for diagnostics.	VCID
DIAGNOSTIC FLAG 2	Diagnostic flag 2. Used for diagnostics.	VCID

Chapter 17. Working with error messages

This section contains information about viewing and modifying the error messages issued by Data Virtualization Manager server.

Viewing details about an error message

You can view information such as severity, explanation, and action item about an error message.

Procedure

To display more information about an error message:

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, enter the option **C AVZ Admin** and press ENTER
2. From the **Server Management Menu**, select **2 AVZ Params** and press ENTER.
3. Scroll down and find the parameter group **PRODMESSAGES**.
4. Type **D** to display the information about the group.

The information such as description, severity/parameter value of the error messages under the group is displayed.

```
----- Parameters ----- Scr 1 Row 1 of 1777
LCs: D Display E Edit F Format P Print CB S Show CB

Parameter          Parameter
Description        Value
SEVERITY OF MESSAGE ID <SSID>0001      'U'
SEVERITY OF MESSAGE ID <SSID>0002      'S'
SEVERITY OF MESSAGE ID <SSID>0003      'S'
SEVERITY OF MESSAGE ID <SSID>0004      'S'
SEVERITY OF MESSAGE ID <SSID>0005      'S'
SEVERITY OF MESSAGE ID <SSID>0006      'S'
SEVERITY OF MESSAGE ID <SSID>0007      'S'
SEVERITY OF MESSAGE ID <SSID>0008      'S'
SEVERITY OF MESSAGE ID <SSID>0009      'S'
SEVERITY OF MESSAGE ID <SSID>0010      'H'
SEVERITY OF MESSAGE ID <SSID>0011      'W'
SEVERITY OF MESSAGE ID <SSID>0012      'W'
SEVERITY OF MESSAGE ID <SSID>0013      'S'
SEVERITY OF MESSAGE ID <SSID>0014      'S'
SEVERITY OF MESSAGE ID <SSID>0015      'S'
Command ==>>>                               Scroll ==>> PAGE
```

- Type **D** next to the corresponding error message to view information such as description and action item about the message: . In the following example, information about the error message **0194** is displayed.

```
***** Top of Data *****
MESSAGE <SSID>0194W, SEF start-up procedure %1 ended with special RC (8
) - auto-enable of SEF rules bypassed - no SEF rule are enabled

EXPLAIN The xxxxINEF start-up procedure ended with RC=8, indicating
rule enablement should be bypassed. The server will bypass
rule enablement. You can enable rules later, manually, but
until you do so, no event related processing is performed
by the SEF facility.

ACTION Determine if the warning is anticipated. If not, correct
the xxxxINEF start-up procedure and restart the server.
***** Bottom of Data *****
```

- Type **F** next to an error message to view the configured values of the message. In the following example, information about the error message 0193 is given below.

```

BROWSE      Parameter Information                               Line 0000000000 Col 001 066
***** Top of Data *****
Parameter Name      <SSID>0193
Description Text    SEVERITY OF MESSAGE ID <SSID>0193
Group Name         PRODMESSAGES
Updatable Parameter Y
Read-Only Parameter N
Maximum Value      0
Minimum Value      0
Parameter Counter  0
Last Update Timestamp
Set During Initialization 0
Changed During Initialization 0
Set After Initialization 0
Changed After Initialization 0
Last Update Userid
Parameter Value     'W'
***** Bottom of Data *****

```

The following table [Table 53 on page 380](#) describes the description of the message.

Column name	Sort name	Column definition
Description	DESCRIPTION	1 to 50-character parameter description
Value	VALUE	Parameter value
Name	NAME	1 to 20-character parameter name
Updatable	UPDATABLE	Value can be changed
Output only	ONLY	Parameter cannot be set
Minimum value	MINIMUM	Minimum value for numeric parameters
Maximum value	MAXIMUM	Maximum value for numeric parameters
-Init-- Set	SETINIT	Number of times set during initialization
-Init-- Chg	CHANGEINIT	Number of times value changed during init
-After- Set	SETAFTER	Number of times set after initialization
-After- Chg	CHANGEAFTER	Number of times value changed after init.
Timestamp	CHANGED	Last time the parameter was changed.
Last Userid	USERID	Last userid that changed the parameter.

Sorting an error message

You can sort an error message based on the description and the parameter value in an ascending order or a descending order.

Procedure

To sort:

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select the option **C AVZ Admin** and press ENTER.
2. From the **Server Management Menu**, select **2 AVZ Params** and press ENTER.
3. Scroll down and find the parameter group **PRODMESSAGES**.
4. Type **D** to display the information about the group.

```
----- Parameters ----- Scr 1 Row 1 of 1777
LCs: D Display E Edit F Format P Print CB S Show CB

Parameter          Parameter
Description         Value
SEVERITY OF MESSAGE ID <SSID>0001      'U'
SEVERITY OF MESSAGE ID <SSID>0002      'S'
SEVERITY OF MESSAGE ID <SSID>0003      'S'
SEVERITY OF MESSAGE ID <SSID>0004      'S'
SEVERITY OF MESSAGE ID <SSID>0005      'S'
SEVERITY OF MESSAGE ID <SSID>0006      'S'
SEVERITY OF MESSAGE ID <SSID>0007      'S'
SEVERITY OF MESSAGE ID <SSID>0008      'S'
SEVERITY OF MESSAGE ID <SSID>0009      'S'
SEVERITY OF MESSAGE ID <SSID>0010      'H'
SEVERITY OF MESSAGE ID <SSID>0011      'W'
SEVERITY OF MESSAGE ID <SSID>0012      'W'
SEVERITY OF MESSAGE ID <SSID>0013      'S'
SEVERITY OF MESSAGE ID <SSID>0014      'S'
SEVERITY OF MESSAGE ID <SSID>0015      'S'
Command ==>>>                               Scroll ==>> PAGE
```

5. To sort:

- Based on the description: type **sort description desc** or **sort description asc** to sort in a descending order or in an ascending order, respectively.
- Based on the severity value or the parameter value: type **sort value desc** or **sort value asc** to sort in a descending order or in an ascending order, respectively.

Changing the severity of an error message

You can change the severity of an error message by performing the following task.

Procedure

1. From the IBM Data Virtualization Manager for z/OS's Primary Option Menu, select the option **C AVZ Admin** and press ENTER.
2. From the **Server Management Menu**, select **2 AVZ Params** and press ENTER.
3. Scroll down and find the parameter group **PRODMESSAGES**.
4. Type **D** to display the information about the group.

```

----- Parameters ----- Scr 1 Row 1 of 1777
LCs: D Display E Edit F Format P Print CB S Show CB

Parameter Description Parameter Value
SEVERITY OF MESSAGE ID <SSID>0001 'U'
SEVERITY OF MESSAGE ID <SSID>0002 'S'
SEVERITY OF MESSAGE ID <SSID>0003 'S'
SEVERITY OF MESSAGE ID <SSID>0004 'S'
SEVERITY OF MESSAGE ID <SSID>0005 'S'
SEVERITY OF MESSAGE ID <SSID>0006 'S'
SEVERITY OF MESSAGE ID <SSID>0007 'S'
SEVERITY OF MESSAGE ID <SSID>0008 'S'
SEVERITY OF MESSAGE ID <SSID>0009 'S'
SEVERITY OF MESSAGE ID <SSID>0010 'H'
SEVERITY OF MESSAGE ID <SSID>0011 'W'
SEVERITY OF MESSAGE ID <SSID>0012 'W'
SEVERITY OF MESSAGE ID <SSID>0013 'S'
SEVERITY OF MESSAGE ID <SSID>0014 'S'
SEVERITY OF MESSAGE ID <SSID>0015 'S'
Command ===> Scroll ===> PAGE

```

5. Overwrite the parameter value and press ENTER.

Chapter 18. Migrating maps

Use the Map Migration utility to move your virtual table maps from a development environment to a test or production environment or from one release to another.

Before you begin

Before using the Map Migration utility, make sure that the following prerequisites have been met:

- **IBM Data Virtualization Manager studio requirements**

If migrating DB2 virtual tables, target systems used by each table must be defined in the target server using one of the following definitions:

- If you want to use the same target system name, define the target system name on the target server.
- If you want to use a different target system name, then define the new target system name, and use the `TSYS=OLD_TSYS,NEW_TSYS` parameter in the AVZGNMPM batch migration utility.

- **Data Virtualization Manager server requirements**

Make sure that both the origin and destination servers have been started.

- **Data Virtualization Manager server security requirements**

The following table summarizes the security permissions required to use the migration utility:

	JCL library	Map export PDS	Server map data set
	The location where the JCL resides.	The PDS library to which the exported metadata objects are unloaded.	The AVZMAPP DD data set, which must be the first data set in the concatenation if the parameter NEW_MAP_DSN is not set.
Batch user ID	UPDATE	CREATE READ	N/A
Server user ID	N/A	UPDATE	UPDATE READ

About this task

The Map Migration utility facilitates change control of the virtual table maps. Change control is the process of moving the virtual table maps defined in a development environment to a test or production environment or from one release to another.

You can use the AVZGNMPM member located in your `hlq.SAVZCNTL` data set for migrating virtual table maps. See the AVZGNMPM member for a list of parameters available for use when migrating virtual table maps.

You can use the AVZGNMPM member to perform the following tasks:

- Migrate one or multiple virtual table maps from one server to another.
- Change the virtual table map definition using the optional parameters. See the comments in the sample job for more details.

Procedure

1. Customize the migration utility job, AVZGNMPM, for the requirements at your site.
2. Submit the AVZGNMPM batch job. Utility job AVZGNMPM extracts the contents of the maps, stores the metadata objects in the map export PDS library, and creates the batch job that is used to rebuild the maps on the target server.
3. Submit the batch JCL that was created in the previous step to rebuild the maps on the target server.

Results

The utility extracts the content of the map export PDS and rebuilds the map on the target server.

Chapter 19. Virtualizing mainframe data

You can virtualize and access mainframe data using batch processing, the ISPF Server Data Mapping Facility, or the IBM Data Virtualization Manager studio.

- *Batch* is typically used in a production lifecycle for adding and updating maps in your production environment. Batch provides an audit trail for monitoring mainframe changes.
- The *Data Virtualization Manager server ISPF* interface provides interface facilities for accessing data sources and a Data Mapping Facility for creating maps.
- The *IBM Data Virtualization Manager studio* allows you to connect to data sources and map data. In IBM Data Virtualization Manager studio data maps are referred to as virtual tables and virtual collections. For more information, see *Creating server metadata*.

Virtual tables (maps)

Mapping data means that the source data's definition is used to create a virtual table that matches the definition of the source data. In the IBM Data Virtualization Manager studio data maps are referred to as virtual tables for SQL solutions or virtual collections for NoSQL solutions.

The data definition depends on the programming language that compiles it. For example:

- For COBOL, it is a file definition or data definition.
- For PL/I, it is a Data Control Language (DCL) statement.

The information (length, format, and field elements) is extracted from the data definition and made available to the Data Virtualization Manager server. The data maps refresh process is governed by the **Auto Refresh** parameter that is specified by using the Data Mapping Defaults Options. To access this feature in ISPF, select D Data Mapping from the **Primary Option Menu**. Select 0 Map Defaults from the **Server Data Mapping Facility Menu** and then set the **Auto Refresh** parameter to Yes.

Note: For batch job use the refresh control card in the SYSIN DD

Once created, a data map is called by using a parameter that is passed with an ODBC/JDBC SQL statement. The data map controls the parsing and formatting of the result set, including the names that are assigned to columns. By calling different maps, the Data Mapping Facility (DMF) can return different views or subsets of the data.

Data maps are created by using a series of ISPF panels that allow you to specify a data set containing a compile listing of a program that contains a data definition. The information (length, format, type, and offset, for example) about each field element is extracted from the data definition and made available to Data Virtualization Manager server.

Applications that use Data Virtualization Manager server through a Data Virtualization Manager client, JDBC, and ODBC can use the data maps to manipulate or view the logical or physical data.

Note: The extracts for COBOL and PLI data maps are also available in batch. The AVZMFAP member is included in the server distributed *hlq.SAVZCNTL* data set as a sample JCL for extracting these types of maps in batch.

When you use the DMF, follow these guidelines:

- Use one server as a test server and another server as a production server.
- Use the DD statement AVZMAPP as part of your initial setup to identify the data sets that contain the maps for your production server.
- For each server, allocate one or more data sets, as needed. To facilitate central control of the production map data set, allocate a “staging” data set for interim maps.

Restrictions for column extraction

The DMF can process up to 7,500 columns for a result set. If more than 7,500 columns are extracted, the extract process continues, but it is recommended that you disable any unwanted columns to reduce the total to 7,500.

Multiple schema support

Create and manage multiple schemas for virtual tables.

With the IBM Data Virtualization Manager for z/OS, you can create and manage multiple schemas. This feature supports the use of multiple virtual schemas for virtual tables in Data Virtualization Manager. You can create schemas in the server using JCL batch files, you can manage the schemas using the JCL batch files and Data Virtualization Manager Studio.

With this feature you can:

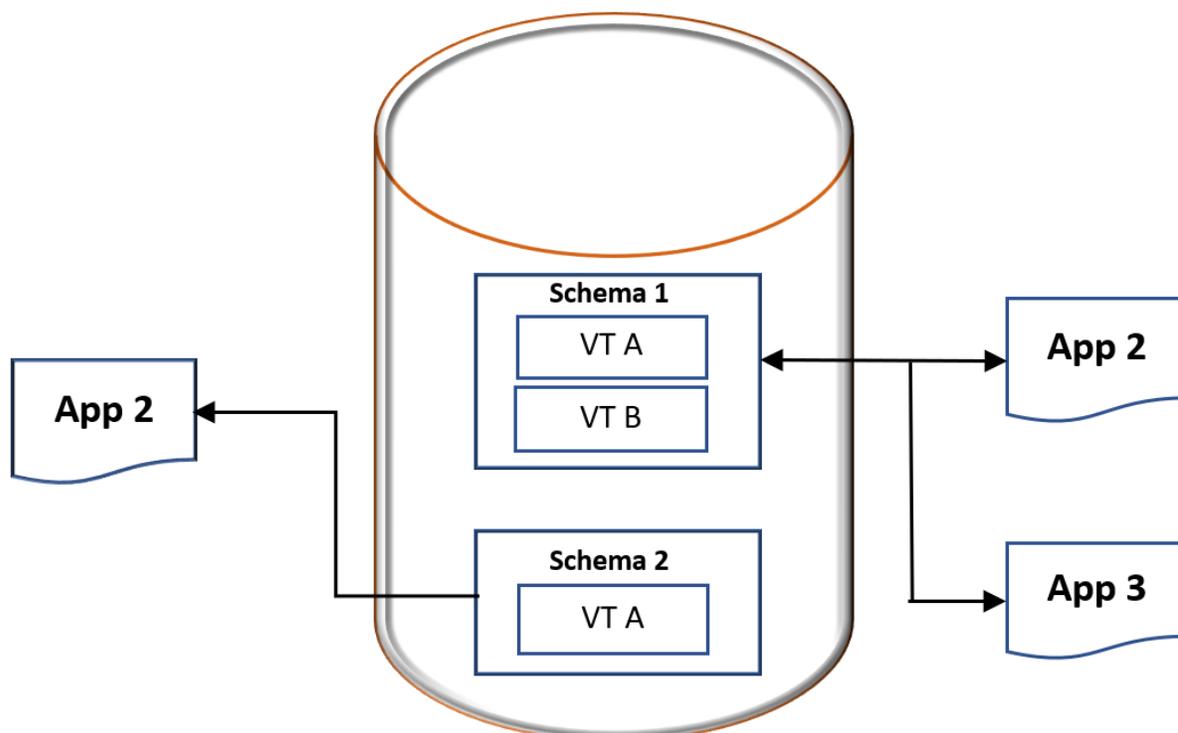
- create multiple virtual tables with the same names in different schemas,
- create tables with same name for different departments or functions in an organization, and
- create virtual tables with the same names that references to different data sources.

Enhanced security

Support for multiple schemas enhances the security by giving you granular control and letting you configure the security at a schema level instead of just at the table level.

You can now additionally configure the security at the schema level and at the schema data set level. You can configure a schema such that only specified users can run queries from the Studio, JDBC, ODBC, DSCClient on the tables that are part of that schema. You can secure the data set of a schema so that only specified users can create virtual tables or have read access to the virtual tables.

The following diagram illustrates an example where two different schemas have virtual tables with similar names and are used by multiple applications:



Related concepts

[“Prerequisites for adding virtual tables to a schema” on page 387](#)

Ensure to follow these prerequisites before adding virtual tables to a schema.

Related tasks

[“Creating schema maps on the server” on page 387](#)

Create new schema maps on the server using the AVZMSCHM member and updating the JCL statements in the control file.

Prerequisites for adding virtual tables to a schema

Ensure to follow these prerequisites before adding virtual tables to a schema.

You can only allocate the virtual tables that you can query using SQL or use with the SQL engine to a schema. You cannot add a virtual table that uses legacy interfaces to a schema. This limitation is because all the legacy interfaces use 31-bit storage, and the schema support requires 64-bit storage.

Here are the prerequisites for creating schemas:

- Every schema must have an associated data set that holds the virtual tables for that schema. The data set is not defined in the started task JCL and is allocated and deallocated dynamically when a server discovers a schema.
- You must associate only one data set per schema.
- You can create a data set by an IEFBR14 step in the batch job or by the studio wizard. If any other process creates a data set, Data Virtualization Manager can use the dataset, but you must ensure that the data set is correct.
- A schema map has a schema name and a map name. The schema name must not exceed 64 bytes. The map name defaults to the first 50 bytes of the schema name if it is not specified. If this default map name is not unique, you might come across an error, and you must provide a map name.
- Currently, you can create only 200 schemas and virtual directories per server.

Related concepts

[“Multiple schema support” on page 386](#)

Create and manage multiple schemas for virtual tables.

Creating schema maps on the server

Create new schema maps on the server using the AVZMSCHM member and updating the JCL statements in the control file.

About this task

A new member called AVZMSCHM is added to the JCL statement in the control file. The job contains instructions of the required modifications and information about the differences in the statement.

Update the JCL with the required information using the SYSIN statement.

Procedure

1. Modify the JCL according to the instructions provided in the h1q.CNTL (AVZMSCHM) member. Update the SYSIN statements as follows:

```
SYSIN DD*
SCHEMA = NAME_OF_YOUR_SCHEMA
SCHEMA DATASET = MAP_DATA_SET_FOR_MAPS_IN_THIS_SCHEMA SCHEMA MAP
SAVE OPTION = SAVE
REFRESH OPTION = REFRESH
```

Option	Description
SCHEMA	Name of your schema that you can use in the SQL query.
SCHEMA DATASET	Data set name for the schema.
SCHEMA MAP	Map name for the schema.

2. Change the existing migration jobs by updating the h1q.CNTL (AVZGNMPM) member. Add a new SYSIN statement as follows:

```
OPT VT SCHEMA=
VT SCHEMA=OLD_SCHEMA,NEW_SCHEMA
VT SCHEMA=OLD_SCHEMA_1,NEW_SCHEMA_1
```

Option	Description
VT SCHEMA	The schema name in which the virtual table must reside. The format is old schema, new schema. Add only one change pair per line.

To move a map that does not have a schema to a schema, use the ADD option as follows:

```
VTSHEMA=ADD,NEW_SCHEMA
```

Note: Only one add statement is allowed per migration.

Related concepts

[“Multiple schema support” on page 386](#)

Create and manage multiple schemas for virtual tables.

Related tasks

[“Moving the existing maps or virtual tables to a schema” on page 388](#)

Move the existing maps or virtual tables to a specific schema using the h1q.CNTL (AVZGNMPM) member. You can also use the standalone utility using the h1q.CNTL (AVZDMUT) member to move existing maps to a schema.

[“Viewing schema maps from the ISPF panel” on page 389](#)

In the ISPF panel, view all the schema maps by navigating from **Data Mapping** to **Map Display**.

Moving the existing maps or virtual tables to a schema

Move the existing maps or virtual tables to a specific schema using the h1q.CNTL (AVZGNMPM) member. You can also use the standalone utility using the h1q.CNTL (AVZDMUT) member to move existing maps to a schema.

About this task

Move the existing maps or virtual tables to specific schemas using either the SYSIN statement in the h1q.CNTL (AVZGNMPM) member or the standalone utility in the h1q.CNTL (AVZDMUT) member.

Procedure

1. Moving the maps using the SYSIN statement in the h1q.CNTL (AVZGNMPM) member. Use the SYSIN statements as follows:

```
SYSIN DD *
VT SCHEMA = ADD,NEW_SCHEMA
MAP=NAMES_OF_MAPS_TO_PUT_IN_THIS_SCHEMA
```

Note: Using the migration job with the same SOURCE and TARGET SSID, the virtual table is copied to the specified schema and the original map is also available in the default schema.

2. Moving the maps using the standalone utility in the h1q .CNTL (AVZDMUT) member: You can run the standalone batch utility outside Data Virtualization Manager. Take your current map library and specify the schema map library that you want to use. Ensure that schema map library name matches with schema name that you defined when you created the schema. Any data that is not defined to be moved to the schema map library is transferred to the NEW_MAP_LIBRARY (refer to the sample job for details). This NEW_MAP_LIBRARY must be used in the JCL server's started task under the AVZMAPP DD name. Additionally, the original data set under AVZMAPP must be renamed and kept as a backup. If you want to revert the changes, point the AVZMAPP DD to the backup (the original CURRENT_MAP_LIBRARY). After completing the changes to the map data set, the server must be recycled.

Use the instructions that are described in the job as follows:

```
//STEP1 EXEC PGM=AVZDMUT,REGION=0M
//STEPLIB DD DISP=SHR,DSN=&LOADLIB
//SYSEXEC DD DISP=SHR,DSN=&REXXLIB
//AVZMAPP DD DISP=SHR,DSN=CURRENT_MAP_LIBRARY
//SCHEMAMP DD DISP=SHR,DSN=SCHEMA_MAP_LIBRARY
//NEWMAPP DD DISP=SHR,DSN=NEW_MAP_LIBRARY
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ADDSHEMA SCHEMA_NAME
MAPNAME
MAPNAME
/*
```

You can use a single * (asterisk) as a wildcard to move multiple VTs between the schemas.

For example,

- TRADE* will move all the VT names starting with TRADE, such as TRADEXYZ, and TRADEABC;
- *TRADE will move all the VT names irrespective of the text after *; and
- TRA*DE* will search and move all the VTs with name TRA*DE because, the first wildcard from the right would be considered in case of multiple wildcards.

Note: Only VSAM, SEQUENTIAL, VIEWS, ADABAS, DB2, IDMS and IMS VIEWS can be moved using the AVZDMUT) batch job.

3. After performing the changes, recycle the server.

Viewing schema maps from the ISPF panel

In the ISPF panel, view all the schema maps by navigating from **Data Mapping** to **Map Display**.

About this task

View all the schema maps by navigating from **Data Mapping** to **Map Display**. A list of schema maps is displayed. DVSQ1 is the default schema. The existing maps are part of the DVSQ1 schema. You can open a schema and display the list of maps that are part of the schema.

Note: On the ISPF panel, the structure name field displays the map name.

Using batch JCL jobs

Batch processing is typically used in a production lifecycle for adding and updating maps in your production environment. Batch JCL jobs provide an audit trail for monitoring mainframe changes.

About this task

There are different ways to use batch processing:

- You can create a virtual table in the IBM Data Virtualization Manager studio and use a batch job to copy the map into your production environment. For more information on creating virtual tables using IBM Data Virtualization Manager studio, see Creating virtual tables section.

- You can use the batch job to create the new virtual table that is then put directly into your production environment.

Note: Using the IBM Data Virtualization Manager studio is the recommended method to create virtual tables.

Procedure

You still must use a compiled listing to perform the extract.

A COBOL listing with OPT(FULL) cannot be processed to produce a map. Keywords for this process define the same elements that you would specify on the ISPF panels. Batch extract does not support alternate indexes for VSAM.

Using the ISPF application

IBM Data Virtualization Manager for z/OS supports access to many data sources.

IBM Data Virtualization Manager for z/OS Interface for ACI

The Advanced Communication Interface (ACI) enables applications that are written in COBOL, Assembler, PL/I, or Natural and running in remote transaction processing (TP) environments to communicate with the desktop.

ACI allows developers to create applications that can run services in their transaction processing (TP) environments. In this case, the IBM Data Virtualization Manager for z/OS Interface for ACI provides access to transactions in a CICS or Batch environment using the ACI API.

This interface also provides data access to JDBC and ODBC clients, web browser clients, and *n*-tier applications. It allows any JDBC- or ODBC-enabled application to use standard JDBC or ODBC facilities to make requests directly to a COBOL, Assembler, PL/I, or Natural program. A relational result set is returned to the application running in its native transaction processing environment.

The ACI Interface Facilities option on the Data Virtualization Manager server - Primary Option Menu provides access to the Server ACI Facility features.

Option	Description
ACI Server Definition	Create ACI server map information
Natural Extract	Extract from Natural source
COBOL Extract	Extract from a COBOL listing
PL/I Extract	Extract from a PL/I listing
ACI Map Display	Display ACI server map information
Map Display	Display all map information
Map Copy	Copy maps
Map Refresh	Refresh maps
Active Server Display	Display active ACI servers
ACI Error Create	Create ACI error processing definitions
ACI Error Display	Display ACI error processing definitions
ACI Execution Errors	Display ACI execution errors
CICS Global ACI Count	Monitor CICS global ACI counters
ACI Buffer Pools	Display ACI buffer pool information

ACI server map information

Before you can use the CALL DVS_ACI request for data, you must first define a map to the server by using the ACI Server Definition option in the ACI Facility.

Using this option, you can define an ACI server map. Users can create ACI server maps (service definitions) for the following types of servers:

- CICS servers
- Batch servers
- Stored procedures

A stored procedure is a started task that runs as a stored procedure for ACI.

The map defines and stores the definition of a remote service application. The definitions are retrieved when referenced in the second parameter of the CALL DVS_ACI request.

Defining an ACI server map

ACI server maps can be created in either of the following ways:

- Creating an ACI server map in batch
- Creating an ACI server map by using the **Server ACI Facility** panel

Creating an ACI server map in batch

Use the AVZACMP1 member (located in the *hlq.SAVZCNTL* data set) for sample JCL that you can use to create ACI server maps in batch.

Creating an ACI server map using the Server ACI Facility panel

Data Virtualization Manager server allows you to create ACI server maps using the ISPF panel.

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. From the Server ACI Facility panel, select **ACI Server Definition** and press **Enter**.

The following sections guide you through creating an ACI CICS server definition and an ACI batch server definition.

Creating an ACI server definition for CICS

The Data Virtualization Manager server ISPF panel allows you to create an ACI batch server definitions for CICS.

About this task

Use the following procedure to create the ACI server definition for CICS using the Server ACI Facility.

Note: You can use the AVZACMP2 member (located in the *hlq.SAVZCNTL* data set) for sample JCL that you can use to create ACI CICS server data maps in batch.

Procedure

1. From the **Server ACI Extract** menu, select **Create ACI CICS Server Definition** and press Enter.
2. Complete the following fields:

Note: The (R) or (O) at the end of each field indicates whether the field is Required or Optional.

- Server Name: This value must correspond to the service information defined in the service application.
- Server Service Class: This value must correspond to the service information defined in the service application.

- **Server Service:** This value must correspond to the service information defined in the service application.

Note:

- The combination of the server name, server service class, and server service identifies a service.
- If part of the name is changed while a service is active, all ACI services that are associated with the former name are treated as orphan services because an ACI service with that name no longer exists in the system. The ACI services that are associated with the former name still appear in the active ACI server maps and continue to display until they time out or until they are manually terminated.

- **Persistent Connection:** Y (Yes) allows persistent connections, and N (No) allows non-persistent connections. The following differences distinguish persistent connections from non-persistent connections:

- A persistent connection allows ongoing conversational requests and responses. The server is "assigned" to the client until the client issues an end-of-conversation (EOC) request, at which point, the ACI server program deregisters the service and terminates. When another connection requests the same ACI service, a new ACI server is started. This implies that the client and service are in conversation mode.

It is also possible for a persistent ACI server to be reused by different client connections after the EOC request by deregistering and registering.

Note: Reusing persistent connections improves performance and reduces overhead. For information about how to create a program to reuse persistent connections, see [“Reusing persistent connections”](#).

- A non-persistent connection is one in which a single request is issued and a single response is received. The server is available for use by any client on a receive request. The service can be used by any incoming client connection with the Data Virtualization Manager Server.
- **Secure this Service:** Y (Yes), restricts the connection to the user who has a SAF resource for the ACI service. Only a user ID with a valid resource defined for the ACI service is allowed to start and connect to that service during its life. This field defaults to N (No), which allows any user ID to start and connect to that started ACI service. The format of the resource name is:

```
ACI.aci-mapname
```

Note: To secure a persistent ACI connection, you must edit and submit one of the following sample jobs (depending on your security type) located in your *hlq.SAVZCNTL* library to specify the map name to be used.

- AVZRAACI for RACF security
- AVZA2ACI for CA ACF2 security
- AVZTSACI for CA Top Security security

- **Mirror Transaction:** The name of the CICS transaction that corresponds to the EXCI mirror program DFHMIRS.
- **Connection Name:** The name of the CICS connection, as defined in CICS and the Data Virtualization Manager configuration member IN00, for DPL requests in CICS.
- **Transaction Name:** The user transaction to start in the CICS region.

Note: The Mirror Transaction, Connection Name, and Transaction Name are configured by the user so the transaction can be invoked under CICS. They are previously defined in the Data Virtualization Manager configuration member.

- **Unit of Work Participant (for persistent connections only):** Indicates whether this transaction can process units of work. If so, the transaction must also support a persistent connection (the **Persistent Connection** field must be set to Y (Yes)).

- **Maximum UOW Buffer Size** (for UOW participants only): The maximum buffer size that UOW transactions can accommodate for any single call (the maximum size of data that the ACI interface can send to the ACI service at any one time). The buffer size is rounded to 1000-byte increments and the maximum is 32,000. If 32,000 is specified, the ACI interface reduces that number to 31,767 bytes at execution time to comply with the maximum size the ACI service can receive at any one time.

Note: The client can send any size data, including SQL_LONGVARCHAR and SQL_LONGVARBINARY data types, which can be greater than 31,767 bytes long. The data that is received from the client is buffered on the Data Virtualization Manager Server until a UOWLAST or UOWONLY request is received [“Query syntax”](#), at which time it is sent to the ACI service in size increments that do not exceed the maximum UOW buffer size value.

- **Max Execution Time:** The maximum time, in seconds, that an ACI service can run before the ACI service is set to TIMEOUT status, at which point, the client is released and receives notification that the ACI service timed out. If you do not set this value, the client non-activity timer value is used.

For more information about the timeout values, see [“Timeout values”](#).

- **Secure Server to Userid:** This value defaults to N (No), which allows any user ID to reconnect to that started ACI service. To restrict the connection to the user who started the ACI service, set this value to Y (Yes). Then only the user ID that started the ACI service is allowed to reconnect to the service during its life.
- **Auto Start:** service Indicates to the Data Virtualization Manager server that it can start this service.
- **Max. No. Allowed:** The maximum number of concurrent servers of this definition type that can run at any given time.

Note: In cases where CICS is slow or has performance problems, a client request can be submitted to multiple ACI services. To prevent a client request from being submitted to all ACI services, you can limit the number of ACI services that the client request can be submitted to [“Using submission limit checking”](#). When the registration is requested by a CICS program running online (not started by the Data Virtualization Manager server), the Max Allowed setting does not take effect. See [“Running a CICS program not started by Data Virtualization Manager server”](#)

- **Auto Terminate** (for non-persistent connections only): A number 0 - 99999 to indicate the number of receives to accept before the system automatically terminates the server and the service deregisters itself. If this field is blank, the default value is 0 (zero). Complete this field only if you specified N (No) for the **Persistent Connection** field.

Note: This field limits the number of times that the server can receive requests after which it is terminated to protect storage resources.

- **Client Non-Activity Timer:** The non-activity timer, which has the following functions:
 - It is the amount of time that the client waits for the service to return before it times out.
 - If the max execution time value is not set, it is used for the time that an ACI service can run before timing out and releasing the client.
 - If the maximum wait for server timer value is not set, it is used for the time that a client can wait for an ACI service to be assigned before timing out.
 - For persistent services, it is also the amount of time that the service remains idle waiting for a client to converse with the service (the amount of time that is allowed for a client to interface with a service).

Note: For persistent services only, if the ACIPERSISTTIMEOUT (ACI PERSISTENT SERVER TIMEOUT) parameter is set to SERVER, the Server shutdown non-activity timer value is used for all of the functions that are listed in the client non-activity timer description.

However, the ACIPERSISTTIMEOUT would not yet apply for servers that are still in a pending registration state. Prior to registration, the Data Virtualization Manager uses the max wait for server. If that is not set, the Data Virtualization Manager uses the client non-activity timer. If the timer is not set, the DV uses a default of 15 seconds. For more information about the timeout values, see [“Timeout values”](#).

- **Server Shutdown Non-Activity Timer (for non-persistent connections only):** The amount of time the service can be non-active before the Data Virtualization Manager Server requests it to terminate. This field allows the less frequently used servers to stop, freeing up storage for the more frequently used servers, improving the use of available resources.

Note: For persistent services, by default, the ACIPERSISTTIMEOUT (ACI PERSISTENT SERVER TIMEOUT) parameter is set to CLIENT, so this field is not used, and the client non-activity timer value is used. For more information about the timeout values, see [“Timeout values”](#).

- **Maximum Wait for Server Timer:** The maximum time that a client can wait for an ACI service to be assigned before the request is timed out and the client is released. If this value is not set, the client non-activity timer value is used. For more information about the timeout values, see [“Timeout values”](#).
 - **SDCIFEN Information:** If using SDCIFEN as the program associated with the CICS transaction defined in the **Transaction Name** field to pass data to the transaction, enter the SDCIFEN information. The following information is required:
 - The name of the program to which SDCIFEN transfers control.
 - The items to be passed to the transaction using the COMMAREA.
3. Press Enter to complete the ACI server definition. If it was successfully created, the system displays the `Service is now defined` message.
 4. Type the END (or press F3) to return to the **Server ACI Facility** options menu.
 5. Select **Map Refresh** to refresh the data maps.

Results

To view the ACI server definitions after creating it, see [“Displaying ACI server map information”](#).

Creating an ACI batch server definition

The Data Virtualization Manager server ISPF panel allows you to create an ACI batch server definition.

Procedure

1. From the **Server ACI Extract** menu, select **Create ACI Batch Server Definition** and press Enter.
2. Provide the following information for the Batch ACI server definition.
 - **Map Name:** The name for the map.
 - **Server Name:** This value must correspond to the service information defined in the service application.
 - **Server Service Class:** This value must correspond to the service information defined in the service application.
 - **Server Service:** This value must correspond to the service information defined in the service application.

Note:

- The combination of the server name, server service class, and server service identify a service.
- If any part of the name is changed while a service is active, all ACI services that are associated with the former name are treated as orphan services because an ACI service with that name no longer exists in the system. The ACI services that are associated with the former name still appear in the active ACI server maps display until they time out or until they are manually terminated.
- **JCL DSN:** Type of JCL DSN if submitting the service as a batch job.
- **Console Command:** Type of console command if submitting the service as a started task.
- **Max Allowed:** The maximum number of concurrent servers of this definition type that can run at any given time.

- In cases where CICS is slow or has performance problems, a client request can be submitted to multiple ACI services. To prevent a client request from being submitted to all ACI services, the IBM Data Virtualization Manager for z/OS Interface for ACI limits the number of ACI services that the client request can submit to as described in [“Using submission limit checking”](#).
- The Max Allowed setting does not take effect when the registration is requested by a CICS program running online (not started by the server). See [“Running a CICS program not started by Data Virtualization Manager server”](#).
- Persistent Connection: Y (Yes) allows persistent connections, and N (No) allows non-persistent connections. The following differences distinguish persistent connections from non-persistent connections:
 - A persistent connection allows ongoing conversational requests and responses. The server is "assigned" to the client until the client issues an end-of-conversation (EOC) request, at which point, the ACI server program unregisters the service and terminates. A new ACI server is started when another connection requests the same ACI service. This implies that the client, and service are in conversation mode.

It is also possible for a persistent ACI server to be reused by different client connections after the EOC request by deregistering and registering.

Note: Reusing persistent connections improves performance and reduces overhead. For information about how to create a program to reuse persistent connections, see [“Reusing persistent connections”](#).

 - A non-persistent connection is one in which a single request is issued and a single response is received. The server is available for use by any client on a receive request. The service can be used by any incoming client connection with the Data Virtualization Manager server.
- Secure this Service: Y (Yes), restricts the connection to the user who has a SAF resource for the ACI service. Only a user ID with a valid resource defined for the ACI service is allowed to start and connect to that service during its life. This field defaults to N (No), which allows any user ID to start and connect to that started ACI service. The format of the resource name is:

```
ACI.aci-mapname
```

- Note:** To secure a persistent ACI connection, you must edit and submit one of the following sample jobs (depending on your security type) located in your *hlq.SAVZCNTL* library to specify the map name to be used.
- AVZRAACI for RACF security
 - AVZA2ACI for CA ACF2 security
 - AVZTSACI for CA Top Security security
- Auto Start service: Indicates to the DV that it may start this service.
 - Unit of Work Participant (for persistent connections only): Indicates whether this transaction can process units of work. If so, the transaction must also support a persistent connection (the **Persistent Connection** field must be set to Y (Yes)).
 - Maximum UOW Buffer Size (for UOW participants only): The maximum buffer size that UOW transactions can accommodate for any single call (the maximum size of data that the ACI interface can send to the ACI service at any one time). The buffer size is rounded to 1000-byte increments and the maximum is 32,000. If 32,000 is specified, the ACI interface reduces that number to 31,767 bytes at execution time to comply with the maximum size the ACI service can receive at any one time.
- Note:** The client can send any size data, including SQL_LONGVARCHAR and SQL_LONGVARBINARY data types, which can be greater than 31,767 bytes long. The data that is received from the client is buffered on the Data Virtualization Manager server until a UOWLAST or UOWONLY request is received (see [“Query syntax” on page 410](#)), at which time it is sent to the ACI service in size increments that do not exceed the maximum UOW buffer size value.

- Auto Terminate (for non-persistent connections only): A number 0 - 99999 to indicate the number of receives to accept before the system automatically terminates the server and the service deregisters itself. If this field is blank, the default value is 0 (zero). Complete this field only if you specified N (No) for the **Persistent Connection** field.

Note: This field limits the number of times that the server can receive requests after which it is terminated to protect storage resources.

- Secure Server to Userid: This value defaults to N (No), which allows any user ID to reconnect to that started ACI service. To restrict the connection to the user who started the ACI service, set this value to Y (Yes). Then only the user ID that started the ACI service is allowed to reconnect to the service during its life.
- Client Non-Activity Timer: The non-activity timer, which has the following functions:
 - It is the amount of time that the client waits for the service to return before it times out.
 - If the max execution time value is not set, it is used for the time that an ACI service can run before timing out and releasing the client.
 - If the maximum wait for server timer value is not set, it is used for the time that a client can wait for an ACI service to be assigned before timing out.
 - For persistent services, it is also the amount of time that the service remains idle waiting for a client to converse with the service (the amount of time that is allowed for a client to interface with a service).

Note: For persistent services only, if the ACIPERSISTTIMEOUT (ACI PERSISTENT SERVER TIMEOUT) parameter is set to SERVER, the Server shutdown non-activity timer value is used for all of the functions that are listed in the client non-activity timer description.

However, the ACIPERSISTTIMEOUT would not yet apply for servers that are still in a pending registration state. Before registration, the DV uses the max wait for server. If that is not set, the DV uses the client non-activity timer. If the timer is not set, the DV uses a default of 15 seconds. For more information about the timeout values, see [“Timeout values” on page 408](#).

- Server Shutdown Non-Activity Timer (for non-persistent connections only): The amount of time the service can be non-active before the Data Virtualization Manager server requests it to terminate. This field allows the less frequently used servers to “die,” freeing up storage for the more frequently used servers, improving the use of available resources.

Note: For persistent services, by default, the ACIPERSISTTIMEOUT (ACI PERSISTENT SERVER TIMEOUT) parameter is set to CLIENT, so this field is not used, and the client non-activity timer value is used. For more information about the timeout values, see [“Timeout values” on page 408](#).

- Maximum Wait for Server Timer: The maximum time that a client can wait for an ACI service to be assigned before the request is timed out and the client is released. If this value is not set, the client non-activity timer value is used. For more information about the timeout values, see [“Timeout values” on page 408](#).

3. Press Enter to complete the ACI server definition. If it was successfully created, the system displays the `Service is now defined` message.
4. Type the END command (or press F3) to return to the **Server ACI Facility** menu.
5. Select **Map Refresh** to refresh the data maps.

Results

To view the ACI server definition after you create it, see [“Displaying ACI server map information”](#).

Extracting ACI data map information

You can extract information from a Natural source, a COBOL source, or a PL/I listing. This extraction provides information about the characteristics of the program's input and output requirements.

Extracting a map from a Natural listing

You can extract a Natural listing by using either of the following methods:

- Using the AVZMFPAR member
- Using the DMF Parser

Using the AVZMFPAR member

To use this method, run the AVZMFPAR member that is located in your *hlq.SAVZCNTL* data set as a sample JCL for extracting Natural maps.

For information about the available parameters in the AVZMFPAR member, see Using batch JCL jobs section.

Using the DMF Parser

The Data Virtualization Manager server console allows you to extract the ACI data mapping information using DMF parser.

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. From the **Server ACI Facility** menu, select **Natural Extract** and press Enter.
The **DMF Map Creation Utility** panel displays.
3. Specify the following information:
 - **Source Library Name:** The data set and member name that contains the source code for the map you want to create.
 - **Nat PGM:** The program name of your batch Natural nucleus.
 - **Load Lib:** The name of the library where the Natural nucleus resides. If you are required to concatenate multiple libraries to resolve all modules that are used during Natural execution, the library names may be used.
 - **PARM:** The Natural nucleus parameters that are required by your installation.
 - **TEMP DSN:** The name of a temporary data set that is used as a work file by this execution.
 - **Temp DSN Space:** Define this value large enough to contain the Natural object listing.
 - **Logon:** The Natural library to be logged on to.
 - **List:** The Natural object to be listed.
 - **ADARUN:** Defines the Adabas execution requirements if not linked in the Natural nucleus. After you enter the information about the panel, press Enter. The system displays a second DMF Map Creation Utility panel.
4. Provide the following information
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set, if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.
 - **Edit Object Listing:** Edit the object listing before the data is parsed and the data map is created.

- **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem.

Note: After migration, the AVZMAPP server must point to the new map data set.

Then, press Enter. The batch Natural nucleus is run to list the object you selected. Then, the ISPF editor may be invoked, depending on the Edit Object Listing selection, so that you can delete or modify information in the object listing.

5. Delete or modify information in the object listing, as appropriate. You can delete lines, fields, or information you do not want to be extracted. Leave any data elements that you want to be extracted in the editor.

The first three lines in the ISPF editor must be deleted, even if all of the other information is required for the extract. Line 1 must be the first line as input into the extract; therefore, preceding lines must be deleted. If this is not done, the Not Valid Source message appears.

6. Type the END command (or press PF3) after you complete all of your edits. The data remaining in the ISPF editor is parsed and the data map is created. An Extract Successful message appears on the extract screen.
7. Type the END command (or press PF3) to return to the **Server ACI Facility** menu.
8. Select **Map Refresh** to add your map to the map display list.

Extracting a map from a COBOL source or COBOL/PLI listing

You can extract a map from a COBOL source, or a COBOL or PLI listing, using either of the following methods:

- Using the AVZMF PAR member
- Using the DMF parser

Using the AVZMF PAR member

The Data Virtualization Manager server ISPF panel allows you to extract a map from a COBOL source or COBOL/PLI list using AVZMF PAR.

Run the AVZMF PAR member that is located in your *hlq.SAVZCNTL* data set as a sample JCL for extracting COBOL and PL/I maps.

For information about the available parameters that are located in the AVZMF PAR member, Using batch JCL jobs section.

Using the DMF Parser

The Data Virtualization Manager server ISPF panel allows you to extract a map from a COBOL source or COBOL/PLI list using DMF parser.

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. From the **Server ACI Facility** menu, select **COBOL Extract** and press Enter.
The **DMF Map Creation Utility** panel displays.
3. Specify the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you want to create.
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set, if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.

- **Convert Var to True:** Select Y (Yes) to convert VAR fields to TRUE VAR fields. TRUE VAR fields are fields that have a 2-byte length of data field that precedes the data.
 - **Flatten Arrays:** Determines whether arrays are flattened. Valid values depend on the product:
 - For IBM Data Virtualization Manager for z/OS SQL, you can specify C (COMPATIBLE) or Y (YES).
 - For IBM Data Virtualization Manager for z/OS Streams, you can specify C (COMPATIBLE) only.
 - For IBM Data Virtualization Manager for z/OS SQL 92, you can specify C (COMPATIBLE), Y (YES), or N (NO).

Note: The C (COMPATIBLE) value is provided for backwards compatibility with an older mapping architecture. When C is specified, OCCURS fields are flattened in the map and OCCURS DEPENDING ON fields generate an error message.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem.
4. Type the END command (or press PF3). An **Extract Successful** message appears on the extract screen.
 5. Type the END command (or press PF3) to return to the **Server ACI Facility** menu.
 6. Select **Map Refresh** to add your map to the map display list.

Displaying ACI server map information

Once the ACI servers are defined, you can view them using the ACI Map Display option. The **ACI Data Mapping Block** panel displays ACI data mapping information only. The data maps displayed in this panel represent the service, or remote application, characteristics.

About this task

To access the **ACI Data Mapping Block** panel:

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. Select **ACI Map Display** from the **Server ACI Facility** menu. Press Enter.

The system displays the **ACI Data Mapping Block** panel.

Note: Some active ACI server information can be returned in a result set using a simple query with the IBM Data Virtualization Manager for z/OS driver. For more information, see [“Using a query”](#) on page 404.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
P	Prints map.
S	Shows map.
D	Disables map.
E	Enables map.
M	Modifies/displays map.

The M command can be used to display or modify an ACI server definition. If no fields are changed on the panel that is displayed, the map is not saved. If any field is changed, the map is saved and a refresh is required to make the changes active. Changes cannot be viewed until a refresh is done.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description
STRUCTURE NAME	The ACI server map name.
STATUS	The active status of the ACI server.
MAX. NO. SERVERS	The maximum number of services that can be running concurrently.
ACTIVE SERVERS	The number of services that are currently running.
HIGH WATER SERVER USAGE	High water marks concerning service usage.
REGISTER COUNT	The number of times a service registers. Incremented when a REGISTER completes.
DEREG COUNT	The number of times a service deregisters. Incremented when a DEREGISTER completes.
SEND COUNT	The number of buffers a service has sent to Data Virtualization Manager server. Incremented when a SEND completes.
RECEIVE COUNT	The number of requests a service has received from a client. Incremented when a RECEIVE or RCV ON SND completes. Note: RECV READY is not counted in this count because a RECV READY means that the program is waiting for a client. As soon as the client connects, a RECEIVE occurs and this is when the count is incremented.
TIMEOUT COUNT	The number of times a client has timed out waiting for a server (see “Timeout values” on page 408). Incremented when a client request (CALL DVS_ACI) times out while waiting for an ACI server to be available.
ABEND COUNT	The number of times a server has terminated abnormally. Incremented when an ACI server service abends.
WAIT COUNT	The number of times a client has been waiting for an available server. Incremented each time a CALL DVS_ACI request waits for an ACI service (for example, a WAITING FOR THE SERVER occurrence).

Note: The counts that are displayed on this panel are reset when the server is restarted.

If an error definition is defined see [“Displaying CICS global ACI counters”](#) on page 405, the columns that are described in the following table also contain information.

Column name	Description
SUSPEND COUNT	The number of times a server has been suspended because of an error.
LAST SUSPENDED DATE TIME	The last time a server was suspended.
SUSPEND ERROR	The error that caused the server to abend.
SUSP_SEC REMAINING TOTAL	The time (in seconds) until the server resumes.
TOTAL ERRORS	The total number of errors that are received by the server.
INACTIVE TIMEOUTS	The number of inactive timeouts.
MODIFICATION DATE TIME	The date and time the map was modified.
USERID	The user ID of the map creator.

Displaying all map information

The **Data Mapping Block** panel displays all data maps that are defined to this Data Virtualization Manager server.

About this task

To display map information:

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. Select **Map Display** from the **Server ACI Facility** menu. Press Enter.

The system displays the **Data Mapping Block** panel. Several panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
D	Disables the map so it is unavailable for use.
E	Enables the map for use.
K	Deletes a map, making it unavailable for use.
P	Prints the associated control block for the selected row.
S	Displays the associated control block for the selected row.
X	Displays the map elements for the selected row.

Column names

The following table provides a description and sort name (if available) for each column name on the ISPF panels.

Column name	Description	Sort name
STRUCTURE NAME	The data map name.	NAME
TYPE	The type of data map.	TYPE
STATUS	The status of the service: <ul style="list-style-type: none">• Enabled• Disabled• Deleted	STATUS
MR	The MapReduce status. Y indicates enabled and N indicates disabled.	MR
LANGUAGE	The language type from which this map was generated.	LANGUAGE
AT	Attachments (OPDWs) present in the map (Yes/No)	AT
MODIFICATION DATE TIME	The creation date and time of this map.	DATE
USERID	The user ID of the map creator.	USERID
CREATION DATASET	The data set from which the map was extracted.	DATASET

Copying ACI maps

Data Virtualization Manager server allows you to copy or move the ACI maps.

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI**.
2. Then, select **Map Copy** from the **Server ACI Facility** panel. Press Enter.
The system displays the Move/Copy Utility panel.
3. Type one of the following commands in the **Option** field:
 - C to copy
 - CP to copy and print
 - M to move
 - MP to move and print
4. In the From ISPF Library fields, provide the information for the data set, including values for the Project, Group, and Type information. If the data set is partitioned, type a member name in the **Member** field:
 - To move, copy, or promote a single member, type the member name.
 - To move, copy, or promote all members, type * (asterisk).
 - To request a member selection list, leave the member name blank or specify a pattern.

Alternatively, for any other partitioned or sequential data sets, you can specify the From Other Partitioned or **Sequential Data Set** field. Type the data set name and volume serial number. Press Enter.

Note: If you forget to enter a password for a data set that requires one, or if you enter the password incorrectly, the system prompts you in standard TSO (line) mode. On TSO/TCAM systems, you may need to press the CLEAR key before responding to the password prompt. If you enter the password incorrectly or encounter any other problems, you may be prompted again to enter the password until you reach a system limit of attempts.

Displaying active ACI server information

You can display active ACI server information in either of the following ways:

- [“Using the active server display” on page 403](#)
- [“Using a query” on page 404](#)

The information that displays represents active services or remote applications that are running in the system. These services are registered to this Data Virtualization Manager server instance and are assigned a server ID. The server name is the same as that defined in the service definition.

Using the active server display

The Data Virtualization Manager server console displays active services or remote applications that are running in the system.

About this task

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. Select **Active Server Display** from the **Server ACI Facility** panel display and press Enter. The system displays the **ACI Servers** panel.

Two panels comprise this program. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

Note: Some active ACI server information can be returned in a result set by using a query. For more information, see [“Using a query” on page 404](#).

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
P	Prints the map.
S	Shows the map.
K	Terminates the map.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description
SERVER ID	The server ID.
SERVER NAME	The name of the server as defined in the service definition.

Column name	Description
STAT	<p>The status of the service:</p> <ul style="list-style-type: none"> • 0: Waiting for work from a client. • 1: Busy or assigned conversationally to a client. • 2: Registered but not assigned. • 3: Deregistered but not released. • 5: Waiting for the program to terminate or reset. • 6: EOC issued waiting for service to process command. • 7: Start issued waiting for service to register. <p>Note: Status 4 is not used.</p>
LAST ACTIVE	<p>This value depends on the status of the service:</p> <p>For status 0, this is the number of seconds that the service has been idle.</p> <p>For other status values, this is the number of seconds that the service has been in use.</p>
CONN ID	The CICS connection name or load balancing name
TRAN ID	The CICS transaction name running in the CICS region for this service.
TASK ID	The CICS task ID running in the CICS region for this service.
MAXIMUM LAST ACTIVE	<p>The high-water mark for the LAST ACTIVE count.</p> <p>Note: The MAXIMUM LAST ACTIVE column displays on the next panel. Use the RIGHT scroll commands (or PF11 key) to scroll to the right.</p>
AVG SEND BUFFER	The send buffer size in bytes.
AVG RECV BUFFER	The receive buffer size in bytes.

Using a query

Using the driver, your application can return active ACI server information in a result set by using a query.

About this task

The syntax of the query is:

```
CALL DVS_INFO('ACTIVEACISERVERS','optional-filters')

```

where:

- ACTIVEACISERVERS (Required) causes the query to return a result set with all active ACI servers listed.
- *optional-filters* (Optional) Specifies a filter for the query. Valid filters are:
 - NAME (*server-name*): Obtains results for the server name specified. For example:

```
CALL DVS_INFO('ACTIVEACISERVERS', 'NAME(SDCIFEN)')

```

- CONNECTION (*connection-name*): Obtains results for the CICS connection name or load balancing name specified. For example:

```
CALL DVS_INFO('ACTIVEACISERVERS', 'CONNECTION(EXCS)')
```

- PERSIST (YES | NO | ALL): Obtains results for servers with the persistent status specified:
 - YES selects persistent servers.
 - NO selects non-persistent servers.
 - ALL (Default) selects all servers (persistent and non-persistent).

For example:

```
CALL DVS_INFO('ACTIVEACISERVERS', 'PERSIST(ALL)')
```

Displaying CICS global ACI counters

With the CICS Global ACI Count option, you can display the current values of MAXTASKS and the number of ACI services that are running for each CICS that is running ACI services. You can also display the last service to update the counter. If the counter is determined to be inaccurate, the counter can be updated by typing over the counter-value.

About this task

To display CICS Global ACI Counters:

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **ACI** and press Enter.
2. Select **CICS Global Count** from the **Server ACI Facility** panel and press Enter. The system displays the **Global ACI Counters Display** panel.

Use the LEFT and RIGHT scroll commands (or PF keys) to shift between the two panels that display the global ACI counters.

Note: The maximum number of ACI services started is managed globally among all Data Virtualization Manager servers. This limits the number of ACI services even when the ACI configuration would otherwise allow more services to start. Refer to the DEFINE CONNECTION statement, MAXTCUSHION parameter. This defines a value that is used to further limit the number of ACI services that can be started. The MAXTCUSHION value is subtracted from the MAXTASKS value found in CICS, and used to reserve some tasks for non-ACI work in CICS.

Use the X line command to view all of the ACI Servers for a CICS APPLID.

Converting program data types to ODBC

COBOL conversions

COBOL conversions describe how COBOL data types are converted to ODBC data types.

<i>Table 56. Conversions of COBOL data types to ODBC data types</i>	
COBOL	ODBC
Alphanumeric	SQL_CHAR
Floating Point	(If 4 bytes) SQL_FLOAT (If 8 bytes) SQL_DOUBLE
Integer	(If 2 bytes) SQL_SMALLINT (If 4 bytes) SQL_INTEGER

Table 56. Conversions of COBOL data types to ODBC data types (continued)

COBOL	ODBC
Numeric	SQL_NUMERIC
Packed	SQL_DECIMAL

Natural conversions

Natural conversions describe how Natural data types are converted to ODBC data types.

Natural	ODBC
A-Alphanumeric	SQL_CHAR
B-Binary	(If 2 bytes) SQL_SMALLINT (If 4 bytes) SQL_INTEGER
C-Attribute Control	N/A
D-Date	*SQL_DECIMAL
F-Floating Point	(If 4 bytes) SQL_FLOAT (If 8 bytes) SQL_DOUBLE
I-Integer	(If 1 byte) SQL_BINARY (If 2 bytes) SQL_SMALLINT (If 4 bytes) SQL_INTEGER
L-Logical	SQL_BINARY
N-Numeric	SQL_NUMERIC
P-Packed	SQL_DECIMAL
T-Time	*SQL_DECIMAL

Note: Although the IBM Data Virtualization Manager for z/OS Interface for Adabas supports the conversion of ODBC date and time to the Natural date and time format, the IBM Data Virtualization Manager for z/OS Interface for Natural only allows the passing of the internal format for date and time (P6 and P12, respectively).

Reusing persistent connections

Persistent connections can be reused so that the ACI service can be used by different client connections. The number of times that a service can be reused is controlled by the application, not by the IBM Data Virtualization Manager for z/OS Interface for ACI.

Once the client issues the end-of-conversation (EOC) request, the ACI service and its CICS task normally become unavailable. For the persistent service to be reused, the program must perform one of the following actions:

- Deregister and register again. Then, go into the RECV READY state.
- Enter the RECV READY state by using `CONV-ID = 'NEW'`. In this case, the IBM Data Virtualization Manager for z/OS Interface for ACI implicitly issues a Deregister/Register on the client's behalf.

Note: Although the ACI service is reused, the server ID of the service is changed each time because of the Deregister/Register calls.

Using submission limit checking

The ACI interface limits the number of ACI services that can be submitted for the client request. When a request to start an ACI server is received from a client, the ACI interface attempts to start the ACI server or waits for a short interval depending on the following criteria:

- The active ACI server queue is searched for an available ACI server that matches the ACI service definition, as configured in the data map. If found, that server is assigned to this request and processing continues.

For more information about creating an ACI service definition, see [“Defining an ACI server map” on page 391](#)

- If either of the following situations occur, the request waits for a short interval:
 - The number of active ACI servers is greater than the Max Allowed setting of the ACI service definition, which specifies the maximum number of concurrent servers allowed.
 - The number of start attempts for this request is greater than the maximum allowed (five).

Note: The first criterion that is met determines the action taken.

If no start attempts have been made for this request, a start attempt is made. The current registration count for this ACI service definition is saved.

If the wait interval is less than a second, the request waits for another short interval.

If a start attempt is made, but the current registration count for this ACI service definition has changed, another requestor may have obtained control of the ACI server. A new ACI server is started and the current registration count for this ACI service definition is saved.

For other situations, the request waits for a short interval as defined. The interval time that a request waits starts at 0.25 seconds and doubles for each waiting interval until it reaches a maximum of 5 seconds.

Note: The WAITING FOR SERVER message does not appear until after the interval reaches five seconds.

The maximum amount of time that a request waits for an ACI server to be assigned is defined by the Maximum Wait For Server Timer value in the ACI service definition; otherwise, the Client Non-Activity Timer value is used.

When the maximum amount of time is reached for the client to wait for an available server, the request terminates with an error, depending on whether any servers are active for this ACI service definition:

- If at least one server is active for this ACI service definition:

```
DVS_ACI ERROR HAS OCCURRED RC -1062; TIMEOUT EXCEEDED, ALL SERVERS ARE BUSY
```

- If no servers are active for this ACI service definition:

```
DVS_ACI ERROR HAS OCCURRED RC -1081; NO ACI SERVICE AVAILABLE / CANNOT  
START ACI SERVICE - CHECK FOR SERVER FAILURE
```

Timeout values

Timeout values describe the amount of time to wait before timing out in the ACI server definition, how each timeout value is set, and the resulting error message that is returned to the requesting application.

<i>Table 57. ACI timeout values</i>		
Event Description	Method of control	Client error code returned
The timeout for a client waiting for an available server (the amount of time that the client waits for a service connection).	<ul style="list-style-type: none"> Maximum Wait for Server Timer If the Maximum Wait for Server Timer value is not specified, the Client Non-Activity Timer value is used. 	<p>If no server is active:</p> <pre>DVS_ACI ERROR HAS OCCURRED RC -1081; NO ACI SERVICE AVAILABLE / CANNOT START ACI SERVICE - CHECK FOR SERVER FAILURE</pre> <p>If a server is active, but unavailable:</p> <pre>DVS_ACI ERROR HAS OCCURRED RC -1062; TIMEOUT EXCEEDED, ALL SERVERS ARE BUSY</pre>
The timeout value for a client waiting for a server to return (the time that is allowed for a service to complete a unit of work before the result is sent to the client).	Client Non-Activity Timer	<pre>DVS_ACI ERROR HAS OCCURRED RC -1065; SERVER HAS NOT RESPONDED, TIMEOUT</pre>
The maximum server execution time (the time that is allowed for a server to run).	<ul style="list-style-type: none"> Max Execution Time If the Max Execution Time is not specified, the Client Non-Activity Timer is used. 	<pre>DVS_ACI ERROR HAS OCCURRED RC -1065; SERVER HAS NOT RESPONDED, TIMEOUT</pre>
The timeout value for an idle server waiting for a client to make a request (the amount of time the service can be non-active before Data Virtualization Manager server requests the service to terminate).	<ul style="list-style-type: none"> Non-Persistent Connections: Controlled by the Server Shutdown Non-Activity Timer. Persistent Connections: Controlled by the Client Non-Activity Timer. 	

Note: For persistent connections, the method of controlling timeout values depends on the value of the ACIPERSISTTIMEOUT (ACI PERSISTENT SERVER TIMEOUT) parameter:

- If the parameter ACIPERSISTTIMEOUT = CLIENT (default) is defined, the client non-activity timer value is used.
- If the parameter ACIPERSISTTIMEOUT = SERVER is defined, the server shutdown non-activity timer value is used for all of the client non-activity timer functions.

Handling interrupted connections

Interrupted connections affect the following items:

- ACI service status
- Client error codes

ACI service status

When the ACI service is busy in status 1, but the connection is interrupted while issuing a CALL DVS_ACI, the IBM Data Virtualization Manager for z/OS Interface for ACI ensures that the situation is handled appropriately by marking the connections as timed out. This allows the server to clean up and deregister. The server is placed in status 5, which indicates that it is waiting for the application to terminate or to reset.

The ACI service remains in status 5 until the application responds by using a SEND/RECEIVE command. Once the SEND/RECEIVE is received, the application receives a TIMEOUT error code (#ETBCB-ERROR-CODE = TIMEOUT). The application then issues a DEREGISTER, and the ACI service is cleaned up.

Client error codes

The client receives an appropriate error code, depending on which of the following occurrences caused the interrupted connection:

- **Connection Timing Out:** If the application reaches the timeout setting while waiting for a server to return or waiting for server execution (see “Creating an ACI server definition for CICS” on page 391 for description) while issuing a CALL DVS_ACI, the application receives the following message:

```
DVS_ACI ERROR HAS OCCURRED RC -1065; SERVER HAS NOT RESPONDED, TIMEOUT
```

Note: In the case of persistent services, subsequent calls to this service get the following message:

```
DVS_ACI ERROR HAS OCCURRED RC -1065; SERVER HAS NOT RESPONDED, TIMEOUT
```

The service that is assigned to the client must be terminated so the client can restart another persistent service and start a new conversation. Once the service is terminated, any subsequent calls to this service receive the following message:

```
DVS_ACI ERROR HAS OCCURRED RC -1071; CONVERSATION HAS NOT BEEN ESTABLISHED OR IS TIMED OUT BY SERVICE
```

The client must start a new conversation.

- **Terminated Connection:** If the connection was terminated, the client receives the following message:

```
Host Communication Failed
```

Connections can be terminated by the following methods:

- Data Virtualization Manager server FAILxxxxxTIME parameter, which terminates the connection if the connection exceeds the value specified.
- Kill line command of the Remote User program (accessed from the Data Virtualization Manager server - Primary Option Menu).

Running a CICS program not started by Data Virtualization Manager server

CICS programs that are not started by Data Virtualization Manager server can register with Data Virtualization Manager server by using the SDBRTX table.

Note: The Max Allowed setting in the ACI service map does not take effect when registration is requested by a program that is not started by Data Virtualization Manager server because this setting limits the number of CICS transactions (the number of programs) that can be started by Data Virtualization Manager server. The MAX NO SERVERS and MAX ACTIVE SERVERS counts in the ACI server maps display do not apply for this type of registration scenario.

When registration is requested by a CICS program that is not started by Data Virtualization Manager server, registration process performs the following actions:

- Determines the Data Virtualization Manager server subsystem. The SDBRTX table is checked to see if an entry with the transaction name matches the transaction name under which the program is running:

- If a match is found, the registration goes to the Data Virtualization Manager server subsystem specified in this entry.
- If no match is found, the subsystem name on the default entry is used.
- Determines the ACI service. The ACI service is determined in the following ways:
 - If ACIDEFAULTCONNNAME ((ACI DEFAULT CONNECTION NAME) is not set, the IBM Data Virtualization Manager for z/OS Interface for ACI bypasses connection name checking. The first ACI service with a triple name that matches the triple name that is specified by the program is used for the registration process.
 - If ACIDEFAULTCONNNAME is set, the IBM Data Virtualization Manager for z/OS Interface for ACI enables connection name checking, which means that an ACI service is used for the registration process only if the triple name matches the triple name that is specified by the program and the connection name matches the value of ACIDEFAULTCONNNAME. If no match is found, the registration request receives an ACI error code of 01000100.

Query syntax

The syntax of a query is:

```
CALL DVS_ACI('function','datamaps','data1',...,'dataN')
```

where

function is SEND, SOC, EOC, UOWFIRST, UOWMIDDLE, UOWLAST, or UOWONLY:

- SEND: The data strings are sent to the server defined in the server data map.
 - Note:** The SEND function implies that you receive information in return.
- SOC: Start of conversation. This function is required for persistent servers. It is used to obtain an existing service or to start a server and lock a server from use by the client.
- EOC: End of conversation. This function is required for persistent service. It is used to notify the service that it is no longer registered to a client.
- UOWFIRST: Indicates that this message is the first part of a unit of work (UOW). The messages are accumulated on the Data Virtualization Manager server until a request indicates a function of UOWLAST is received.
- UOWMIDDLE: Indicates that this message is not the first part or the last part of a UOW. The messages are accumulated on the Data Virtualization Manager server until a request indicates a function of UOWLAST is received.
- UOWLAST: Indicates that this message is the last part of a UOW. When this is received, the Data Virtualization Manager server processes the entire UOW, sending the messages to the ACI service in the size increments it desires.
- UOWONLY: Indicates that this is a one-message UOW. When this is received, the Data Virtualization Manager server processes the UOW, sending the messages to the ACI service in the size increments it desires.

Note: For a UOW, the size of the message segments that are sent by the client are not dependent on the size that the ACI service can accept. Any size segment can be sent by the client by using the SQL_LONGVARCHAR and SQL_LONGVARBINARY data types.

datamaps are the data maps (up to three data map names):

- (Required) Specifies the map that defines the server to which the request is being assigned. The map name is required.
- (Optional) Client map in (CMI). Defines the client input (data1-dataN) presented to the server. The CMI represents the data format expected by the service.

If CMI is coded, the data parameters data1-dataN are validated as described in [“Data validation”](#).

Note:

- For special considerations on passing numeric data with CMI, see [“Passing numeric data”](#).
- CMI is not supported for UOW calls. If a CMI is specified for a UOW call, the following message is generated:

```
DVS_ACI ERROR HAS OCCURRED RC -1086; INPUT DATA MAP NOT ALLOWED FOR UNIT
OF WORK TRANSACTIONS
```

- (Optional) Server map output (SMO). Describes the data as presented by the server. If SMO is coded, the data buffer output from the source is presented as a result set described by the SMO data map.

If the SMO is not specified, the Data Virtualization Manager server cannot determine the maximum size of the row in the result set until the first SEND call of each CALL DVS_ACI invocation is made. Once the first SEND call is issued, the Data Virtualization Manager server uses the length of the first SEND call to establish the maximum size of the row in the result set.

Note the following guidelines:

- If an optional CMI and SMO are specified, separate them by commas.
- If a CMI is omitted and an SMO is specified, use a comma as a placeholder for the CMI.
- You can run a simple CALL statement to return metadata for the CMI or SMO.

data1-dataN describes the data input to the server. If more than one data area is coded, a CMI is required.

CMI considerations

When passing an ACI input map (CMI), remember the following considerations:

- Data validation
- Passing numeric data

Data validation

If CMI is coded, the data parameters data1-dataN are validated in the following ways:

- If only one data parameter is given, the CMI is used for validation of data types only; that is, numeric fields are numeric.
- If more than one data parameter is given, the CMI is used to validate and buffer the data components as input to the server.

Passing numeric data

The following considerations exist for passing numeric data with a CMI:

- **Packed Decimal Fields.** If a field is defined as Packed Decimal in the ACI input map, the following guidelines apply:
 - If the value passed has a scale that is too long, the size of the scale in the ACI input map is used. The IBM Data Virtualization Manager for z/OS Interface for ACI allows the value if the adjusted precision is less than or equal to the precision in the ACI input map, and the scale is truncated.
 - Although the IBM Data Virtualization Manager for z/OS Interface for ACI allows you to pass a string to a Packed Decimal field, it does not allow the decimal point to be specified in the string (the decimal is based on what is defined in the ACI input map). Also, if the length of the string exceeds the precision of the field, the leading digits in the string are truncated. For these reasons, it is not recommended to pass a string value to a packed field.
- **SmallInt Fields.** The IBM Data Virtualization Manager for z/OS Interface for ACI allows a value that is passed as an integer to a field defined in the ACI input map as SmallInt. Ensure that the value is less than or equal to 32767. Otherwise, the data is truncated.

Using a CALL statement to obtain map metadata

The IBM Data Virtualization Manager for z/OS Interface for ACI allows users to view metadata information for CMI and SMO maps on the client with a simple CALL statement. This allows a user to pass all the input

parameters with the correct data types as required by the CMI or SMO without having to go into the server ISPF panels to locate this information.

The format of the CALL statement is:

```
CALL DVS_MAP('DESCRIBE',mapname')
```

where *mapname* is the name of the map.

The CALL statement returns a result set with a single column named FORMAT. This column contains details on the fields of the map. The following table describes the FORMAT column types and their SQL equivalents.

<i>Table 58. FORMAT column types and the SQL equivalent</i>	
FORMAT types	SQL types
CHARACTER	SQL_CHARACTER
NUMERIC	SQL_NUMERIC
DECIMAL	SQL_DECIMAL
INTEGER	SQL_INTEGER
SMALLINT	SQL_SMALLINT
FLOAT	SQL_FLOAT
DOUBLE	SQL_DOUBLE
DATE	SQL_DATE
TIME	SQL_TIME
TIMESTAMP	SQL_TIMESTAMP
VARCHAR	SQL_VARCHAR
LONGVARCHAR	SQL_LONGVARCHAR
BINARY	SQL_BINARY
VARBINARY	SQL_VARBINARY
LONGVARBINARY	SQL_LONGVARBINARY
UNICODE	SQL_UNICODE
UNICODE_VARCHAR	SQL_UNICODE_VARCHAR
UNICODE_LONGVARCHAR	SQL_UNICODE_LONGVARCHAR

Adabas

The IBM Data Virtualization Manager for z/OS Interface for Adabas allows ODBC, JDBC, and Web clients to access Adabas data in a relational model by using simple SQL-based queries. This interface can be used with traditional client/server applications, desktop productivity tools that use ODBC, and 2-tier and 3-tier Web implementations. Using the IBM Data Virtualization Manager for z/OS Interface for Adabas, any ODBC- or JDBC-enabled application can use standard ODBC or JDBC facilities to make SQL requests directly to Adabas. The result is a relational result set, with no host programming required.

The Adabas Interface Facilities option on the Data Virtualization Manager server - Primary Option Menu provides access to the Server Adabas Data Mapping Facility features.

<i>Table 59. Server Adabas Data Mapping Facility</i>	
Option	Description
Map Defaults	Set map options
Map Create	Create maps
Map Display	Display all map information
Map Copy	Copy maps
Map Refresh	Refresh maps

Creating Adabas virtual tables using the Data Mapping Facility in batch

To extract and import Adabas data, use the sample JCL in the AVZMFPAR member.

Member AVZMFPAR, which is in the *hlq.SAVZCNTL* data set, contains sample JCL for extracting Adabas virtual tables.

For information about the available parameters in the AVZMFPAR member, see Using batch JCL jobs section.

IBM Data Virtualization Manager for z/OS Interface for CICS/TS

The CICS/TS interface offers quick and easy access to CICS/TS, making CICS data directly available to non-mainframe users.

The Data Virtualization Interface for CICS/TS provides a tool for making quick and easy CICS/TS queries. IBM Data Virtualization Manager for z/OS, in conjunction with this interface, allows existing and new CICS programs to be rapidly integrated into client/server applications with minimum modification to the code.

The Data Virtualization Manager server component of IBM Data Virtualization Manager for z/OS connects to CICS by using the IBM EXCI (External CICS Interface). In IBM Data Virtualization Manager for z/OS, a CICS program behaves like a stored procedure. Applications can call the procedure by using industry-standard mechanisms, passing in the input data as part of the call. The CICS program returns data by using the COMMAREA to the Data Virtualization Manager server. The Data Virtualization Manager server, through the Data Mapping Facility, formats the display of the returned results.

The CICS Interface Facilities option on the IBM Data Virtualization Manager for z/OS - Primary Option Menu provides access to the Server CICS Control Facility features.

<i>Table 60. Server CICS Control Facility</i>	
Option	Description
CICS Connections	Monitor and Control CICS Connections
CICS Sessions	Monitor and Control CICS Sessions
CICS Data Mapping	Create CICS Map Information
CICO	CICS Global Information Control Block
EXCI	EXCI Global Information Control Block
CICS Load Balancing	Monitor CICS Load Balancing
CICS Global ACI Cnt	Monitor CICS Global ACI Counters

Choosing a connectivity method

Before you use the IBM Data Virtualization Manager for z/OS Interface for CICS/TS, decide which of the following connectivity methods best fits your needs:

- Stored Procedures (metadata support)

- Custom RPC, a sample of which is DVSCIBSC
- ODBC CICS RPC, a sample of which is SDCOCIRP

Stored procedures

A stored procedure contains the metadata for input and output fields, as well as other required parameters for accessing CICS transactions. It stores this information in the Data Mapping Facility.

To use a stored procedure to perform a CICS/TS query, see [“Creating a stored procedure by using the CICS Control facility”](#) on page 414.

Remote Procedure Calls (RPCs)

The RPC types provide the following benefits:

- Custom RPC, DVSCIBSC (sample) provides users a way to invoke multiple CICS programs and return a customized result set to the client in only one network round trip.
- ODBC CICS RPC, SDCOCIRP (sample) provides users a way to eliminate the restriction of a 32 KB COMMAREA.

Custom RPC, DVSCIBSC (sample)

The sample Custom RPC, DVSCIBSC, can be run with a CALL statement from any ODBC- or JDBC-compliant application, such as Visual Basic, PowerBuilder, and MS-Access. Because the applications are developed by using APIs provided on the host, they can return an ODBC or a JDBC result set to the application. Using custom RPCs, the RPCs run and end similar to the way a normal DB2 query runs and ends. The results of the RPC are accessed in the same manner as the results of a DB2 query.

The source for the sample RPC, DVSCIBSC, is on the z/OS host in the *hlq.SAVZSAMP* data set. Compiled copies of these samples are also provided in the *hlq.RPCLIB* data set.

Note: This call is under the restriction of the 32 KB limitation of the COMMAREA.

To use the custom RPC to perform a CICS/TS query, see [“Using the custom RPC, DVSCIBSC \(sample\)”](#) on page 419.

ODBC CICS RPC, SDCOCIRP (sample)

The ODBC CICS RPC, SDCOCIRP, is a remote procedure call that runs in the CICS address space and uses IBM Data Virtualization Manager for z/OS APIs to avoid the restrictions of the 32 KB COMMAREA. ODBC CICS RPCs enable non-CICS applications that run non-z/OS operating systems to access enterprise data and applications that are managed by CICS.

Using this connection option, the driver can make a single network roundtrip by using multiple programs, and return a single customized result set of the combined data. For example, if you are running five programs, and you want specific columns of data in each program, the ODBC CICS RPC delivers only one set of with the specified data. You can also return zero or multiple result sets from an ODBC CICS RPC.

In addition, this connection option allows you to connect to different CICS regions from a single Data Virtualization Manager server.

To use the ODBC CICS RPC to perform a CICS/TS query, see [“Using the ODBC CICS RPC, SDCOCIRP \(sample\)”](#) on page 419.

Creating a stored procedure by using the CICS Control facility

You can use the CICS Control facility to create a map extract by using COBOL or PL/I and generate a stored procedure.

Extracting a map from a COBOL and/or PLI listing

You can extract a COBOL and/or PLI listing using either of the following methods:

- Using the AVZMFPAR member

- Using the DMF parser

Using the AVZMF PAR member

To use this method, run the AVZMF PAR member that is located in your *hlq.SAVZCNTL* data set as a sample JCL for extracting CICS maps.

For information about the CICS parameters that are located in the AVZMF PAR member, see Using batch JCL jobs section.

Using the DMF parser

About this task

Use the DMF parser to extract a map from COBOL. The procedure is similar for the PLI extract. For information about extracting a VSAM map, see [“Creating data maps for VSAM file access”](#).

Procedure

1. From the Primary Option Menu, select **CICS** and press Enter.
2. Select **CICS Data Mapping** from the CICS Control Facility Display and press Enter.
3. Make sure the Parser Version is set to N (New).
4. Return to the **CICS Data Mapping** menu and select **Extract Cobol** and press Enter.
5. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you want to create.
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set, if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero. Otherwise, the offset starts at the offset of the field in the structure.
 - **Flatten Arrays:** Determines whether arrays are flattened. Valid values depend on the product:
 - For IBM Data Virtualization Manager for z/OS SQL, you can specify C (COMPATIBLE) or Y (YES).
 - For IBM Data Virtualization Manager for z/OS Stream, you can specify C (COMPATIBLE) only.
 - For IBM Data Virtualization Manager for z/OS SQL 92 Engine, you can specify C (COMPATIBLE), Y (YES), or N (NO).

Note: The C (COMPATIBLE) value is provided for backwards compatibility with an older mapping architecture. When C is specified, OCCURS fields are flattened in the map and OCCURS DEPENDING ON fields generate an error message.
6. Press Enter. If the extract completes with no errors, the **Extract Successful** message appears in the upper right corner of the panel. Both the map library and the Data Virtualization Manager server contain the mapped structure definition.
7. Return to the **CICS Data Mapping** panel and select **Map Refresh** to add your map to the map display list.

Note: The output from the extract is a virtual table definition that is placed in the named map library. The virtual table library member name is the name that is associated for this virtual table by Data Virtualization Manager server.

After virtual table generation, you must disable virtual table fields that are not used by your application by using the Map Display option of the Data Virtualization Manager server CICS Data Mapping panel.

Generating a stored procedure by using data map(s)

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **Generate a Stored Procedure from Maps** and press Enter.

2. Provide the following information:

- **Map Data Set Library/Other Map Data Set Name:** The DSNAME of the map data set. This is the location of the input and output maps that are used to create the stored procedure and where the stored procedure map is created.
- **Map Names:** The input and the output maps that were created. In this example, the same map is used for input and output.
- **Procedure Name:** The name of the CICS procedure to create. This is written to the DMF.
- **Interface Type:** Type C (CICS).
- **CICS Transaction ID:** A mirror transaction name. If you are unsure of the mirror transaction name, type EXCI, the default value. This name is used to communicate with the CICS address space.

Note: If you specify a null value for the CICS Transaction ID, IBM Data Virtualization Manager for z/OS issues the DPL request with a NULL pointer for the transaction ID address to allow for CICS transaction dynamic routing. In addition, this implementation allows your application to specify NULL in the client setting Transaction Name (TRNA). If this keyword is set, any CICSEX request that is issued by the application results in a DPL request with a NULL pointer for the transaction ID address that is passed to CICS.

- **CICS Transaction ID2:** The TRANID2 associated with CICS transaction, if applicable.
- **CICS Connection Name** or **CICS Load Balancing Group Name:** The CICS connection name or CICS load balancing group name. This name is used for the DPL request at run time.

Note: If you do not specify the connection name or load balancing group name when creating the map, the name is picked up from the EXCICONNECTIONNAME parameter (this is the EXCI DEFAULT CONNECTION NAME parameter in the IBM Data Virtualization Manager for z/OS started task parameter PRODCICS group).

- **CICS Program Name:** The name of the CICS program to be invoked. This program must have a valid PPT entry that is defined to CICS.
- **Dynamic Connection** and **TRANID Support:** An optional map that is used to dynamically change the default connection name or load balancing name and TRANID at execution.

By default, the stored procedure that is defined uses the CICS connection name or load balancing name and TRANID defined for the stored procedure. This definition can be changed at client connection time by using the ODBC/JDBC client CNNA and TRNA connection keywords. Using the client CNNA and TRNA keywords to override the defaults requires that a new connection be established. To change these defaults at execution time, you can add input parameters either at the beginning (PREFIX) or end (SUFFIX) of the stored procedure input parameter list. By specifying PREFIX or SUFFIX, input parameters CICS_CONNECTION and CICS_TRANSACTION are automatically added to the stored procedures input map definition (the original input map that is supplied is not modified.). These are optional parameters and if set to "" (a 0-length string), existing defaults are used. If set to another value at execution time, the new value is used to run the CICS stored procedure.

3. Return to the **CICS Data Mapping** panel and select **Map Refresh** to add your map to the Map Display list.

The pseudo-stored procedure name is returned by the ODBC function call SQLProcedures when directed at the Data Virtualization Manager server address space. The contained metadata is returned by SQLProcedureColumns so that a SQLExecute can be used to invoke the procedure. IBM Data Virtualization Manager for z/OS maps all application parameters as mapped by the previous extract and passes them to the CICS program. All output is passed according to the previously mapped output map.

The procedure generation process uses the value that is specified in the IBM Data Virtualization Manager for z/OS started task parameter CICSPROCOWNER. The default for the CICS stored procedure owner CICSPROCOWNER is CICSEX. CICSPROCOWNER can be found in the IBM Data Virtualization Manager for z/OS started task parameter group PRODCICS.

- Return to the **CICS Data Mapping** panel and select **Map Display** to view the map display list and press Enter.

This example shows the first of two panels for displaying existing data maps. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

To generate the stored procedure using the Studio, see Generating a CICS stored procedure section.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
P	Prints the associated control block for the selected row.
S	Displays the associated control block for the selected row.
D	Causes map to be unavailable for use.
E	Enables the map for use.
K	Deletes a map, also making it unavailable for use.
X	Displays map elements for the selected row.

Type the command to the left of the line and press Enter.

Column names

The following table describes each column name on the ISPF panels and provides a sort name.

Column name	Description	Sort name
STRUCTURE NAME	The member names in the map data set.	NAME
TYPE	TYPE <ul style="list-style-type: none"> • ADABAS • Input • Output • Screen • LPTBL • Header • USER 	TYPE
STATUS	The status of this map (Enabled, Disabled, or Deleted).	STATUS
LANGUAGE	The language of the extracted map. This value is determined at the time of the extract.	LANGUAGE

Column name	Description	Sort name
AT	Attachments (OPDWs) present in the map (Yes/No)	AT
MODIFICATION DATE TIME	The date and time the map was last modified.	DATE
USERID	The user ID of the map creator. Used only for informational purposes.	USERID
CREATION DATASET	The data set from which the map was extracted.	DATASET

Generating a stored procedure by using studio

You can also create a stored procedure for CICS using studio.

About this task

Perform the following steps to generate a stored procedure in studio.

Procedure

1. On the **Server** tab, expand **SQL > Stored Procedures > CICS**.
2. Right click **CICS** and choose **Create Virtual Stored Procedure**.
3. In the **New Stored Procedure Wizard**, provide the following information:
 - **Name:** The name of the CICS procedure to create. This is written to the DMF.
 - **Metadata Library:** The DSNAME of the map data set. This is the location of the input and output maps that are used to create the stored procedure and where the stored procedure map is created.
 - **Description:** The description you want to provide for the created procedure.
4. Click **Next**.
5. Select an input map from **COBOL** or **PL/I** listing.
6. Select the input parameters for the selected input map.
7. Click **Next**.
8. Select the result/output map and output columns and click **Next**.
9. Provide the following information.
 - **Transaction ID:** The TRANID2 associated with CICS transaction, if applicable.
 - **CICS Connection Name:** The CICS connection name or CICS load balancing group name. This name is used for the DPL request at run time.

Note: If you do not specify the connection name of the load balancing group name when creating the map, the name is picked up from the EXCICONNECTIONNAME parameter (this is the EXCI DEFAULT CONNECTION NAME parameter in the started task parameter PRODCICS group).
 - **Mirror Transaction:** A mirror transaction name. If you are unsure of the mirror transaction name, type EXCI, the default value. This name is used to communicate with the CICS address space.
 - **Program Name:** The name of the CICS program to be invoked. This program must have a valid PPT entry that is defined to CICS.
 - **Dynamic:** An optional map that is used to dynamically change the default connection name or load balancing name and TRANID at execution. By default, the stored procedure uses the CICS connection name or load balancing name and TRANID defined for the stored procedure. This definition can be changed at client connection time by using the ODBC/JDBC client CNNA and TRNA connection keywords. Using the client CNNA and TRNA keywords to override the defaults requires that a new connection be established. To change these defaults at execution time, you can

add input parameters either at the beginning (PREFIX) or end (SUFFIX) of the stored procedure input parameter list. By specifying PREFIX or SUFFIX, input parameters CICS_CONNECTION and CICS_TRANSACTION are automatically added to the stored procedures input map definition (the original input map that is supplied is not modified.). These are optional parameters and if set to "" (a 0-length string), existing defaults are used. If set to another value at execution time, the new value is used to run the CICS stored procedure.

10. Click **Finish**

Query syntax: accessing a CICS program by using a stored procedure

The syntax of a query is:

Call *owner.procedure-name*(*parm1,parm2,...*)

where:

- *owner* is the name that is defined in the Started Task parameter CICSPROCOWNER. In the following example, CICSEX is the owner.
- *procedure-name* is the stored procedure name that invokes the transaction. In the following example, the stored procedure is TESTPROC.
- *parm1,parm2* defines the parameters to be passed to the CICS transaction. These parameters are defined by using the map extraction procedure in the IBM Data Virtualization Manager for z/OS ISPF interface. These parameter descriptions are passed to the program by using the SQLProcedureColumns function call.

For this example, the actual call would be:

```
Call CICSEX.TESTPROC(parm1,parm2,...)
```

Using the custom RPC, DVSCIBSC (sample)

Run the sample RPC sample program, DVSCIBSC. To do this, compile the COBOL member, which is located in *hlq.SAVZSAMP* data set.

Note: Use the LE370 COBOL compiler. If this compiler is not available, use the COBOL II compiler.

The PART transaction that is delivered by IBM with IMS is used by the sample program and must be installed in your IMS system.

Query syntax

The syntax of the query is:

Call DVSCIBSC

Using the ODBC CICS RPC, SDCOCIRP (sample)

Procedure

1. Compile and link the program SDCOCIRP, found in *hlq.SAMP* file.

Note: Use the LE370 COBOL compiler. If this compiler is not available, use the COBOL II compiler.

2. When you link your CICS program, include SDCPCR, the name of the ODBC CICS RPC API Support Module as shown in the following link step of the CICS program:

```
//LKED      EXEC PGM=IEWL,REGION=&REG,
//          PARM='&LNKPARM',COND=(5,LT,COB)
//SYSLIB     DD      DSN=&INDEX..SDFHLOAD,DISP=CICS load lib.
//          DD      DSN=&LE370HLQ..SCEELKED,DISP=SHRLE library (if using COBOL
for MVS)
//          DD      DISP=SHR,DSN=HLQ.LOAD=Product load lib.
//SYSLMOD    DD      DSN=&PROGLIB(&PROGRAM),DISP=SHRCICS application lib.
//SYSUT1     DD      UNIT=&WORK,DCB=BLKSIZE=1024,
```

```

//          SPACE=(1024,(200,20))
//SYSPRINT DD      SYSOUT=&OUTC
//SYSLIN   DD      DSN=&&LOADSET,DISP=(OLD,DELETE)
//          DD      DSN=&&COPYLINK,DISP=(OLD,DELETE)
//          DD      DDNAME=SYSIN
//*
//STEP01          PEND
//ONE            EXECDFHYITVL,PROGRAM=SDCOCIRP
//LKED.SYSIN DD *
//              INCLUDE SYSLIB(SDCPCR) CICSRPC API support module
//              NAME SDCOCIRP(R)
//*
```

Query syntax

The syntax of the query is:

Call CICSRPC.sdccirp('100100')

where:

CICSRPC is the prefix that identifies the RPC as an ODBC CICS RPC.

sdccirp is the program name in CICS that is the RPC.

'100100' is one or more parameters for the ODBC CICS RPC.

Monitoring and managing the CICS environment

The CICS Control Facility allows you to view and manage your CICS connections and determine the status of CICS connections and sessions.

Invoking the CICS Connections Control Facility

Procedure

1. From the Primary Option Menu, select **CICS** and press Enter.
2. Select **CICS Connections** from the **CICS Control Facility** menu and press Enter.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
F	Formats the information for the selected row.
S	Displays the CMLI control block for the selected row.
P	Prints the CMLI control block for the selected row.
R	Displays the sessions for the selected row.
I	Changes the desired status of the CICS to the INSERVICE status.
O	Changes the desired status of the CICS to the OUTSERVICE status.

Type the command to the left of the line and press Enter.

Column names

The following table provides a description and sort name (if available) for each column name on the ISPF panels.

Column name	Description	Sort name
CONNECTION NAME	The name of the connection that is defined in Data Virtualization Manager.	CONNECTION
NET NAME	The NETNAME defined in Data Virtualization Manager.	NET
GROUP NAME	The name of the group that is defined in Data Virtualization Manager.	GROUP
SECURITY NAME	The security name that is defined in the connection.	SECURITY
ACCESS METHOD	The protocol that is used for communications (EXCI).	ACCESS
IN SERVICE	The status of the service: <ul style="list-style-type: none">• YES indicates the link is available for work• NO indicates the link is unavailable for work• UNKNOWN indicates the link is in transition	
TOTAL SESSIONS	The total number of sessions for the connection.	TOTAL
ACTIVE SESSIONS	The number of active sessions for the connection.	ACTIVE
PROTOCOL TYPE	The protocol being used for the connection.	PROTOCOL
QUEUE DEPTH	The number of requests waiting for a session.	QUEUE
QUEUE TIME	The amount of time the request has been waiting for a session.	TIME
LB CNCT GROUP	The name of the load balancing connection group.	LBGRP
TOT TX COUNT	The number of transactions executed.	
RUN TX COUNT	The number of transactions executed from the start of the refresh interval.	
CONVERT TO ACI	Converts CICS EXCI requests to ACI.	

Invoking the CICS Sessions Control Facility

Procedure

1. From the Primary Option Menu, select **CICS** and press Enter.
2. From the **CICS Control Facility** menu, select **CICS Sessions** and press Enter.

This example shows the first of two panels for displaying the connections control facility. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
F	Formats the information for the selected row.
S	Displays the CMLI control block for the selected row.
P	Prints the CMLI control block for the selected row.
I	Changes the desired status of the CICS to the INSERVICE status.
O	Changes the desired status of the CICS to the OUTSERVICE status.

Type the command to the left of the line and press Enter.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
SESSION NAME	The name of the session.	SESSION
SESSION ID	A unique identifier for each session.	ID
GROUP NAME	The name of the group that is defined in Data Virtualization Manager.	GROUP
CONNECTION NAME	The name of the connection that is defined in Data Virtualization Manager.	CONNECTION
PROTOCOL TYPE	The protocol that is used for the connection.	PROTOCOL
RECEIVE PREFIX	The prefix that is used for receive sessions.	RECEIVE
RECEIVE COUNT	The number of receive sessions to be created for the connection.	COUNT
SEND PREFIX	The prefix that is used for send sessions.	PREFIX

Column name	Description	Sort name
SEND COUNT	The number of send sessions to be created for the connection.	COUNT
IOAREALN MINIMUM	The minimum length of input/output area.	MINIMUM
STATUS	The status of the session (IN SERVICE or OUT OF SERVICE).	STATUS
RESTART COUNT	The number of times the session has been restarted.	
TRANSACTION COUNT	The number of transactions that are executed on this session.	

DB2

The IBM Data Virtualization Manager for z/OS Interface for DB2 offers access to DB2-z/OS data, providing maximum performance for organizations that need to integrate DB2 data with distributed or Web applications without sacrificing flexibility, reliability, or security.

Regardless of how the data is initially represented, the IBM Data Virtualization Manager for z/OS Interface for DB2 can integrate DB2 data and stored procedures without custom coding. In addition, one Data Virtualization Manager server can access many DB2 subsystems.

The DB2 Interface Facilities option on the Primary Option Menu provides access to the Server Database Control feature. The Server Database Control application allows you to view and modify the product Server Database table. This table maps database names to entries in the Link table, which can be displayed using the Link Control application. You can associate a database name with a new host name (link) using a line command.

The DESCSTAT option determines whether Db2 builds a SQL descriptor area (SQLDA) when binding static SQL statements. This SQLDA is where Db2 returns names of result table columns for a DESCRIBE CURSOR statement.

If the value of DESCSTAT is changed from YES (default) to NO, you will receive the error "Output parameter 5 returned data type 0x01F5 incompatible with bound data type 0x01C0 No column labeled "ORDINAL_POSITION" in this result set".

Perform the following steps to change the value of DESCSTAT from YES to NO:

1. BIND/REBIND the packages for the impacted tools.
2. REBIND the packages for the IBM Data Server Driver for JDBC and SQLJ.
3. REBIND the CLI driver that was installed in the DSNTIJC job.

Database control program

The Database control program allows you to view the DB2 databases and group attachment names known to the server, and to reset the logging request queue. The entries in this table are referenced for DB2 thread collection.

Invoking the DB2 control program

The IBM Data Virtualization Manager for z/OS ISPF panel displays first of two connections control facility panels.

Procedure

From the Primary Option Menu, select **DB2** and press Enter.

The Database Control program displays the first of two connections control facility panels. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

Available commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
C	Clears the pending logging requests.
F	Formats database information for the selected row.
P	Prints the associated control block for the selected row.
S	Displays the control block for the selected row.

Column names

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
SUBSYSTEM NAME	The name of the database as it will be referred to in application programs.	NAME
SUBSYSTEM TYPE	The type of database management system.	TYPE
DATABASE VERSION	The version of the database management system	VERSION
DATABASE STATUS	The status of the database management system.	STATUS
MEMBER OF GROUP	Database is a member of group attachment.	GROUP
COMPLETED REQUESTS	The number of completed requests for the database management system.	COMPLETED REQUESTS
PENDING REQUESTS	The number of pending requests for the database management system.	PENDING REQUESTS
SSCT ADDRESS	The address of the Subsystem Communication Table (SSCT) for this database management system.	SSCT ADDRESS
RIB ADDRESS	The address of the Release Information Block (RIB) for this database management system.	RIB
DB MODE	Database operational mode. Valid values are: <ul style="list-style-type: none">• CM: compatibility mode.• ENFM: enable new function mode.• NFM: new function mode.	DB MODE

IBM Data Virtualization Manager for z/OS Interface for IMS DB: support for DBCTL

IMS support for DBCTL accesses IMS data by using DL/I data calls through the CCTL (coordinator controller). The CCTL provides communications for the DBCTL environment and consists of a subsystem that contains a database resource client (DRA).

The DBTCL (database control) is an environment that allows full-function and data entry databases (DEDBs) to be accessed from a management system.

The IMS Interface Facilities option on the Primary Option Menu provides access to the Server IMS Data Mapping Facility features.

Option	Description
Facilities	General IMS Facilities Menu
IMS Data Mapping	Create IMS Map Information
ODBA	Open Database Access Menu

Choosing a connectivity method

The IBM Data Virtualization Manager for z/OS Interface for IMS DB allows access to IMS data when used with the Data Virtualization Manager client, JDBC, or ODBC.

Using the IBM Data Virtualization Manager for z/OS Interface for IMS (CCTL/DBCTL), you can access data by using the method that is described in the following section.

SQL access to IMS DB

The IBM Data Virtualization Manager for z/OS Interface for IMS CCTL/DBCTL allows you to access IMS data by using SQL.

- Logical DBDs are not supported.
- The IBM Data Virtualization Manager for z/OS does not necessarily use the first PCB. It finds the best PCB within the PSB that will satisfy the query. If it is a SELECT statement, it will use a PROCOPT=GO PCB in preference to one with update capability.
- IMS can have separate positioning for each PCB. The IBM Data Virtualization Manager for z/OS can use multiple PCBs within a PSB. For each cursor opened on a database, IBM Data Virtualization Manager for z/OS will require a PCB in the PSB to use. If you open and hold open more than one cursor on the database concurrently, an additional PCB is required in the PSB for each open cursor.

Note: If there are not enough PCB available within the PSB to support the number of cursors open at the same time in the database, you will receive the following error message:

Unable to find available PCB.

If you set the **setAutoCommit** parameter to FALSE, to avoid being unable to find an available PCB, you should consider closing a cursor as soon as possible after the completion of the SQL associated with the cursor.

The process of enabling access to an IMS database involves extracting database information and issuing a query. For more information, see [“Using the method for SQL access to IMS DB” on page 433](#).

Extracting database information

You can extract information about the database from the following sources:

- IMS Database Description (DBD)
- Program Specification Block (PSB)
- Segment detail definitions

Data Virtualization Manager server maintains segment detail definitions in the Virtualization Facility. The primary segment information can be obtained from the IMS DBD for a specific database. The DBD contains segment definitions, which can be viewed as individual segment descriptions. Segment definitions contain information that describes the relationships between segments (parent/child relationships), as well as the information access path.

IMS Database Description (DBD)

To access an IMS database, the IBM Data Virtualization Manager for z/OS Interface for IMS DB/SQL requires that the Database Description (DBD) be extracted to create a DMF data mapping entry for every DBD/segment combination.

Program Specification Block (PSB)

Program Specification Block (PSB) controls the access to IMS databases, and each PSB has one or more Program Control Blocks (PCBs) that define the access to specific databases and database segments within the IMS subsystem. To enable SQL access, a PSB must exist or be built that contains PCBs with the necessary access to the IMS data mapped in virtual tables. The source for IMS PSBs is extracted using the Data Studio tool and related to IMS virtual tables for SQL access.

While executing an SQL query, a client connection to the server defines an implicit database transaction. In IMS, a PSB schedule defines an IMS transaction. For update transactions, only one PSB can be scheduled at any point of time to ensure transactional consistency. When auto-commit is on, every SQL statement is committed, PSB is terminated, and IMS transaction is completed. When auto-commit is off, any updates issued, which are a part of a transaction, must be supported by the PSB scheduled for the transaction. Attempts to perform an update within a single transaction that would require a different PSB (from one already scheduled) returns an error.

If you are updating multiple IMS databases within a single transaction (with auto-commit off), all virtual tables participating in the transaction must use the same PSB and that PSB must include PCBs for all databases to be updated by the transaction.

The implicit database connection is ended under the following conditions:

- After each SQL statement execution when auto-commit is on. In this case, each execute instruction commits the transaction.
- After commit/rollback request when auto-commit is off.
- After the client is disconnected when auto-commit is off. In this case, a commit is issued.

Note: The auto-commit flag is set in the application code and not set within Data Virtualization Manager.

When you extract the PSB, remember the following considerations:

- Segment sensitivity considerations. Access is allowed to all segments contained in the first PCB for a PSB.
- Field sensitivity considerations. If field sensitivity is defined, WHERE clauses are allowed in the query.
- PCB considerations. Based on the SQL statement, the first DB PCB that has the necessary segment sensitivity and the processing options is selected for database access.

Segment detail definitions

Sometimes, database segments are not defined fully in the DBD. Segment layout detail definitions can be obtained from other sources, such as COBOL copybooks. To use segment detail definitions, they must be extracted to create DMF entries, which must be linked to the associated DBD segment.

When extracting the segment detail definitions, remember the following considerations:

- **Field Sensitivity:** If field sensitivity is defined, WHERE clauses are allowed in the query.
- **REDEFINES:** Redefinitions are used to change the information that is accessed by the IBM Data Virtualization Manager for z/OS Interface for IMS DB/SQL into a customized format, depending on how the information is to be presented.

For example, assume PART-KEY is redefined as PART-PREFIX and PART-NUMBER:

```

01 PART-REC
03 PART-KEY PIC X(17).
03 PART-KEY-DETAIL REDEFINES PART-KEY.
05 PART-PREFIX PIC X(02).
05 PART-NUMBER PIC X(15).
03 FILLER

```

In this case, the following SELECT statement is valid for column selection:

```
SELECT PART-PREFIX, PART-NUMBER FROM DI21PART.DFSSAM03_PARTROOT
```

- **OCCURS:** The Data Mapping Facility does not support OCCURS clauses that contain the DEPENDING ON clause. When the OCCURS clause is used, it appends a numeric suffix to the corresponding column.

For example, if you executed the following OCCURS clause on PART-PREFIX:

```
05 PART-PREFIX OCCURS 3 TIMES
```

You would see the following column names:

```

PART-PREFIX-1
PART-PREFIX-2
PART-PREFIX-3

```

Database information

Database information is contained in the following parts:

- DBD: DI21PART
- PSB: DFSSAM03

DI21PART and DFSSAM03 are samples to demonstrate how IMS support works. The samples represent how data shown in a hierarchical model is virtualized in tables.

Database definition (DBD)

This example is the DI21PART DBD of the PART sample database, represented in an IMS view in [Figure 2 on page 428](#).

```

DBD NAME=DI21PART, ACCESS=(HISAM, VSAM)
DATASET DD1=DI21PART, DEVICE=3380, OVFLW=DI21PARO,
        SIZE=(2048, 2048), RECORD=(678, 678)
SEGM   NAME=PARTROOT, PARENT=0, BYTES=50, FREQ=250
FIELD  NAME=(PARTKEY, SEQ), TYPE=C, BYTES=17, START=1
SEGM   NAME=STANINFO, PARENT=PARTROOT, BYTES=85, FREQ=1
FIELD  NAME=(STANKEY, SEQ), TYPE=C, BYTES=2, START=1
SEGM   NAME=STOKSTAT, PARENT=PARTROOT, BYTES=160, FREQ=2
FIELD  NAME=(STOCKEY, SEQ), TYPE=C, BYTES=16, START=1
SEGM   NAME=CYCCOUNT, PARENT=STOKSTAT, BYTES=25, FREQ=1
FIELD  NAME=(CYCLKEY, SEQ), TYPE=C, BYTES=2, START=1
SEGM   NAME=BACKORDR, PARENT=STOKSTAT, BYTES=75, FREQ=0
FIELD  NAME=(BACKKEY, SEQ), TYPE=C, BYTES=10, START=1
DBDGEN
FINISH
END

```

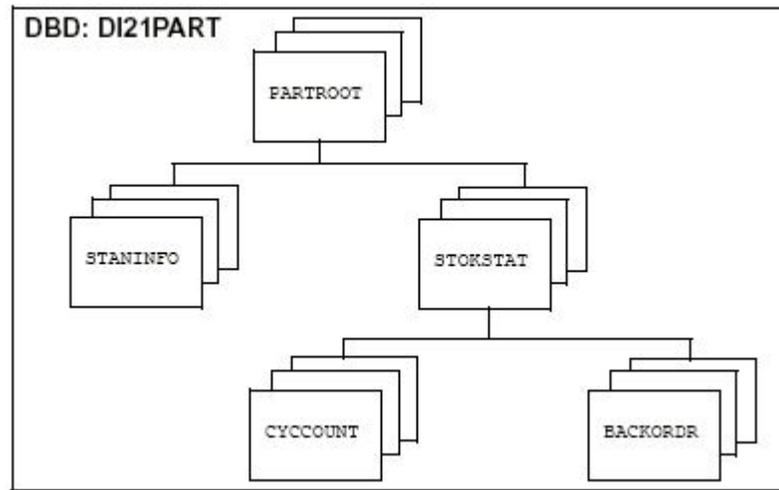


Figure 2. IMS database representation

Program Specification Block (PSB)

This example is the DFSSAM03 PSB of the PART sample database:

```

DBPCB01 PCB      TYPE=DB, DBDNAME=DI21PART, PROCOPT=G, KEYLEN=43
          SENSEG  NAME=PARTROOT, PARENT=0
          SENSEG  NAME=STANINFO, PARENT=PARTROOT
          SENSEG  NAME=STOKSTAT, PARENT=PARTROOT
          SENSEG  NAME=CYCCOUNT, PARENT=STOKSTAT
          SENSEG  NAME=BACKORDR, PARENT=STOKSTAT
          PSBGEN  LANG=COBOL, PSBNAME=DFSSAM03
          END
  
```

Extracting the data

After the maps of the DBD and PSB are extracted, you can use the Data Mapping Facility to navigate through the data.

Because IMS does not maintain a catalog that describes client information for each segment, Data Virtualization Manager server maintains the information in the Data Mapping Facility. An IMS database segment map definition is created based on the SQL statement processing requirements.

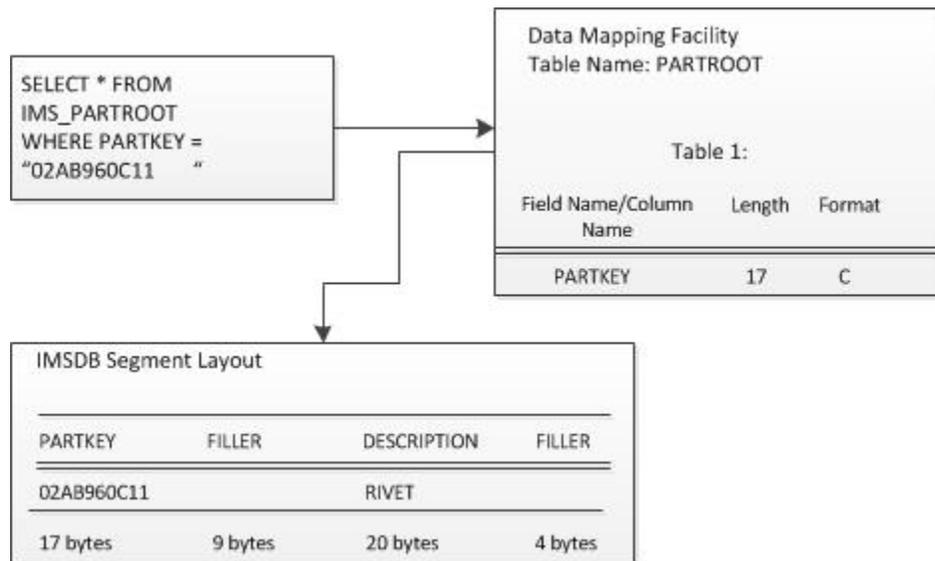


Figure 3. Using the Data Mapping Facility with the IBM Data Virtualization Manager for z/OS Interface for IMB DB/SQL

Data access paths

Data can be accessed in or across hierarchical boundaries. For the DBD shown, all of the SELECT statements that are shown in this section are valid.

The database representation of the DBD shown can be combined with the PSB and divided into specific data paths.

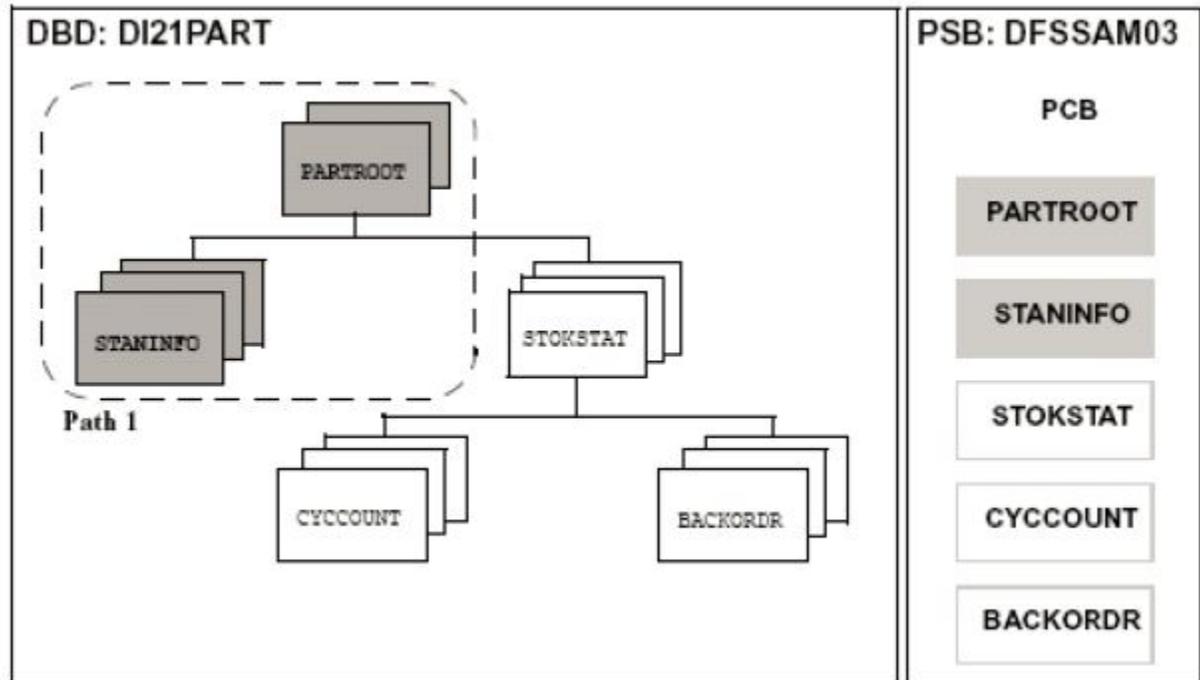


Figure 4. Data access path 1

The following SELECT statements are valid for the data access path that is shown in [Figure 4](#):

```
SELECT * FROM IMS_PARTROOT
SELECT * FROM IMS_PARTROOT P, IMS_STANINFO C
WHERE P.CHILD_ID = C.PARENT_ID
AND P.PARTKEY='02AB960C11'
```

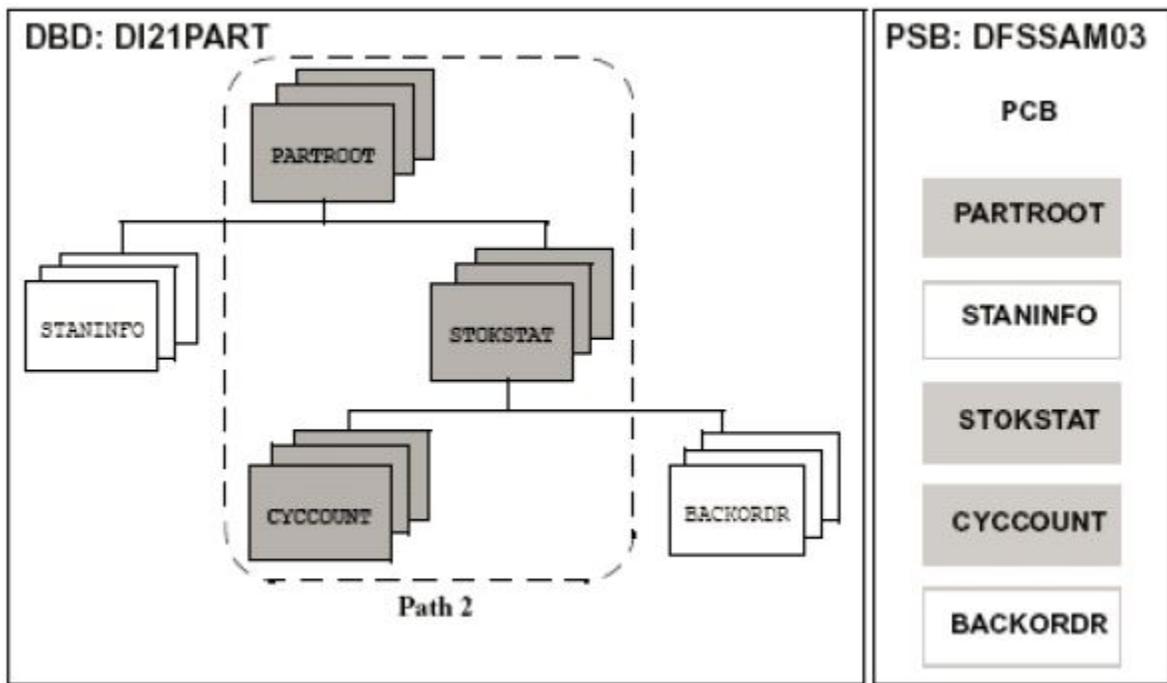


Figure 5. Data access path 2

The following SELECT statements are valid for the data access path that is shown in [Figure 5](#):

```

SELECT * FROM IMS_PARTROOT
SELECT * FROM IMS_STOKSTAT
SELECT * FROM IMS_CYCCOUNT
SELECT * FROM IMS_PARTROOT P, IMS_STOKSTAT C
  WHERE P.CHILD_ID = C.PARENT_ID
  AND PARTKEY='02AB960C11'
SELECT * FROM IMS_PARTROOT P, IMS_STOKSTAT C1,
  IMS_CYCCOUNT C2
  WHERE P.CHILD_ID = C1.PARENT_ID
  AND C1.CHILD_ID = C2.PARENT_ID
  AND PARTKEY='02AB960C11'

```

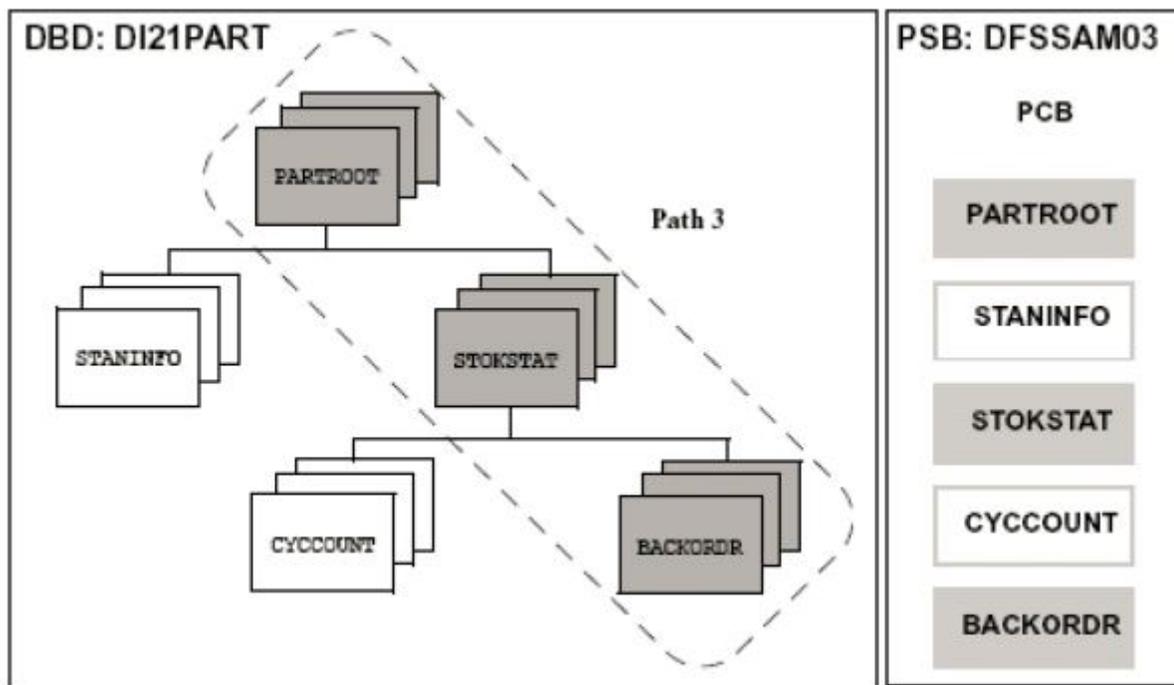


Figure 6. Data access path 3

The following SELECT statements are valid for the data access path that is shown in Figure 6:

```

SELECT * FROM IMS_PARTROOT
SELECT * FROM IMS_STOKSTAT
SELECT * FROM IMS_BACKORDR
SELECT * IMS_PARTROOT P, IMS_STOKSTAT C
  WHERE P.CHILD_ID = C.PARENT_ID
  AND PARTKEY='02AB960C11'
SELECT * FROM IMS_PARTROOT P,
  IMS_STOKSTAT C1,IMS_BACKORDR C2
  WHERE P.CHILD_ID = C1.PARENT_ID
  AND C1.CHILD_ID = C2.PARENT_ID
  AND PARTKEY='02AB960C11'

```

The following statements are not valid because they produce a Cartesian product (or Cartesian join):

```

SELECT * FROM IMS_PARTROOT, IMS_STANINFO
SELECT * FROM IMS_PARTROOT, IMS_STOKSTAT,DI21PART.
  DFSSAM03_CYCCOUNT

```

Running a statement that produces a Cartesian product results in a 1002 error code.

Note: To select from two different tables, a WHERE clause must be specified.

Selecting data

The IBM Data Virtualization Manager for z/OS Interface for IMS DB/SQL code parses the SELECT statement, optimizes it, and processes the data by using the path that is determined by the optimizer. The optimizer examines the SELECT criteria, and combines and sorts it. It also validates the access path.

For generic selections (SELECT *), all enabled columns in the data map for the segments listed in the FROM clause are returned to the client. Selected columns can be requested from any segment in a given path.

PSB security checking

The IBM Data Virtualization Manager for z/OS Server interface for IMS DBCTL/ODBA and IMS-Direct supports PSB authorization. For DBCTL/ODBA requests, Data Virtualization Manager issues a SAF call to verify the Data Virtualization Manager connected user-id has READ access to the PSB prior to scheduling

the PSB in IMS. For IMS-Direct requests, this SAF call is issued even though a PSB is not scheduled in IMS, so PSB level security is consistent regardless of the IMS data access used by the server. To activate this security checking, set the **IMSPSBSECURITY** parameter to yes in the Data Virtualization Manager server's IN00 file.

Creating a data map from SQL

Data Virtualization Manager server allows you to create a data map using SQL functions.

Procedure

1. From the Data Virtualization Manager server - Primary Option Menu, select **IMS** and press Enter.
2. From the Server IMS Control Facility menu, select **IMS Data Mapping** and press Enter.
3. Select **Generate a View of an IMS/ DB DBD and Segment** from the menu and press Enter.
4. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you are creating.
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If this name is not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Case Sensitive:** If the Start Field or End Field are case-sensitive, set this value to Y (Yes) to preserve the case.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set, if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.
 - **Convert Var to True:** Set this value to Y (Yes) to convert VAR fields to TRUE VAR fields. TRUE VAR fields have a 2-byte data length field preceding the data.
 - **Flatten Arrays:** Determines how arrays are processed. Valid values depend on the data source:
 - Flatten arrays into a single fixed table at runtime (Y)
 - Return arrays into separate tables at runtime (N)
 - Flatten arrays now (C)
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem in the server started task.
 - **DBD Name:** The name of the DBD for which you are creating a view.
 - **Segment Name:** The name of the Segment, from the specified DBD, for which you are creating a view.
 - **PSB Name:** The name of the PSB to use to access the specified segment.
Note: If you leave this field BLANK, the product automatically selects a PSB name based on the SQL query.
 - **PCB Name:** The name of the PCB, from the specified PSB, to use to access the specified segment.
Note: If you leave this field BLANK, the product automatically selects a PCB name based on the SQL query.
5. Press Enter.
5. If either the DBD Name or Segment Name field is BLANK, the system displays a panel allowing you to choose a name from a selection list. Select a DBD Name or Segment Name from the Selection List.
6. Press Enter. If the operation is successful, the Create Successful message appears on the panel.

Using the method for SQL access to IMS DB

The IBM Data Virtualization Manager for z/OS Interface for IMS DB provides SQL access. Use the Data Mapping Facility to define maps. Maps are defined once, and then updated/replaced, if needed.

You can extract a map from a source by using either of the following methods:

- [“Using the AVZMFPAR member” on page 433](#)
- [“Using the DMF parser”](#)

Using the AVZMFPAR member

Use the AVZMFPAR member that is located in your *hlq.SAVZCNTL* data set as a sample JCL to virtualize DBC, PSB, and COBOL maps.

For more information about the parameters in the AVZMFPAR member, see Using batch JCL jobs section.

Using the DMF parser

The Data Virtualization Manager server console allows you to extract data mapping information from DBD or PSB sources. Additionally, this facility enables the addition of segment detail definitions to the extracted DBD maps.

About this task

Procedure

1. From the Primary Option Menu, select **IMS** and press Enter.
2. From the **IMS Control Facility** menu, select **IMS Data Mapping** and press Enter.
3. Select **Mapping Defaults** from the menu and press Enter.
4. Make sure the Parser Version option is set to N (New).
5. Extract by using a DBD source:
 - a. Return to the **IMS Mapping Option** panel, and select the **Extract using DBD Source** option. Press Enter.
 - b. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you are creating.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem in the server started task. The map name is the DBD or PSB name. This name also is used as the member name for the map in the mapping data set, if possible.
 - c. Press Enter. If the extract completes with no errors, the **Create Successful** message appears on the panel.
6. Extract the data map by using the PSB source.
 - a. Return to the **IMS Mapping Options** panel and select **Extract Using PSB Source**. Press Enter.
 - b. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you are creating.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem in the server started task. The map name is the DBD or PSB name. This name also is used as the member name for the map in the mapping data set, if possible.
 - c. Press Enter. If the extract completes with no errors, the **Create Successful** message appears on the panel.
7. Optional: Add segment detail definitions to the extracted DBD:

- a. Return to the **IMS Mapping Options** panel and select **Extract COBOL from listing**. Press Enter.
 - b. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you want to create.
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If this name is not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set, if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.
 - **Flatten Arrays:** Determines whether arrays are flattened. Valid values depend on the product:
 - For IBM Data Virtualization Manager for z/OS SQL, you can specify C (COMPATIBLE) or Y (YES).
 - For IBM Data Virtualization Manager for z/OS Streams, you can specify C (COMPATIBLE) only.
 - For IBM Data Virtualization Manager for z/OS SQL 92, you can specify C (COMPATIBLE), Y (YES), or N (NO).

Note: The C (COMPATIBLE) value is provided for backward compatibility with an older mapping architecture. When C is specified, OCCURS fields are flattened in the map and OCCURS DEPENDING ON fields generate an error message.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem in the server started task.
 - c. Press Enter. If the extract completes with no errors, the `Extract Successful` message appears on the panel. Both the map library and Data Virtualization Manager server contain the mapped structure definition.
8. Merge the other maps into the DBD maps to add the segment detail definitions from the COBOL listings to the DBD segments (see [“Merging maps into a DBD map”](#)).
 9. Display the maps to make sure they were all created successfully (see [“Displaying maps”](#)).

Merging maps into a DBD map

Data Virtualization Manager server ISPF panel allows you to merge the data maps into a DBD map.

Procedure

1. From the Primary Option Menu, select **IMS** and press Enter.
2. From the IMS Control Facility Menu, select **IMS Data Mapping** and press Enter.
3. From the panel, select **Merge Other Maps into a DBD map** and press Enter.
4. Enter information in the **DBD Map Merge Utility** panel.
 - Provide the information for the Map Data Set Library, including values for the Project, Group, Type, and Member fields (optional) for the DBD data map. Otherwise, you can use the **Other Map Data Set Name** field to specify another data set for the DBD data map.
 - To disable duplication fields, select the **Disable duplicate fields** parameter.
 - To disable FILLER fields, select the **Disable FILLER fields** parameter.

Press Enter.

 - If you specified a member name, that member is selected and the system displays the Data Map Linkages panel.
 - If you did not specify a member name, the system displays a **Selection List** panel.
5. Select a member:
 - a. From the **Selection List** panel, type one of the following commands in front of the member name:

- B: Browse the member
- E: Edit the member
- S: Select the member

Note: You can process one or multiple members.

b. Type the END command to process the members.

6. From the **Data Map Linkages** panel, in the LINK NAME column, type the names of the data maps that were extracted from the COBOL listing to link with the DBD segments. Press Enter.
7. For each DBD segment that is linked to a data map, the Data Map link established message appears in the MESSAGE column.

Note: To force a mapping update, you must delete or leave the link name blank, and press Enter to process. After you see the Warning: No Linked Data Map defined message, you can rekey the link name and press Enter to pick up the revised map layout. If you performed these steps and are unable to pick up the new definition, you must perform a Map Refresh. You can also set the option Auto Refresh to Y (Yes) on the panel prior to the map extract.

8. Type the END command to process the links. The system returns to the **IMS DBD Map Links** panel. If the linking completes with no errors, the Create Successful message appears on the panel.
9. Return to the **IMS Mapping Options** panel and select **Map Refresh** from the menu.
 - a. Press Enter for a map refresh to add your map to the map display list. If the refresh completes with no errors, the Refresh Successful message appears in the upper right corner of the panel.

Displaying maps

Displaying all maps is useful to make sure that maps are created correctly.

Procedure

1. Return to the **IMS Mapping Options** panel and select **Display IMS DB DBD Maps** from the menu and press Enter.

The system displays the DBD maps. For more information about the available line commands and column descriptions, see the following sections.

2. Return to the **IMS Mapping Options** panel and select **Display IMS DB PSB Maps** from the menu and press Enter.

For more information about the available line commands and column descriptions see the following sections.

3. Return to the **IMS Mapping Options** panel and select **Display IMS DB COBOL/PLI Extract Maps** from the menu. Press Enter.

The system displays the PSB maps.

These examples show the information that displays for existing data maps. Use the LEFT and RIGHT scroll commands (or PF keys) to shift between them.

• Available Commands

This program supports all four scrolling commands (UP, DOWN, LEFT, RIGHT) and their PF key equivalents or scroll bar equivalents.

It also supports the primary SORT and LOCATE commands and the following line commands:

Line commands	Description
D	Disables the map causing it to be unavailable for use.
E	Enables the map for use.
K	Deletes a map, also making it unavailable for use.

Line commands	Description
P	Prints the associated control block for the selected row.
S	Displays the associated control block for the selected row.
X	Displays map elements for the selected row.

- **Column names**

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column names	Description	Sort name
STRUCTURE NAME	The member names in the map data set.	NAME
TYPE	One of the following types of structure: <ul style="list-style-type: none"> – ADABAS – Input – Output – Screen – LPTBL – Header – USER 	TYPE
STATUS	Status of this map (Enabled, Disabled, or Deleted)	STATUS
LANGUAGE	Language type this map was generated from (for example, Adabas, COBOL, DB2, Natural, VSAM). Determined at the time of the extract. The extracted map is independent of language type.	LANGUAGE
AT	Attachments (OPDWs) present in the map (Yes/No)	AT
MODIFICATION DATE TIME	The date and time the map was modified. Used only for informational purposes.	DATE
USER ID	The user ID of the map creator. Used only for informational purposes.	USERID
CREATION DATASET	The data set that the map was extracted from.	DATASET

VSAM and sequential files

The IBM Data Virtualization Manager for z/OS Interface for VSAM provides seamless, real-time controlled access to VSAM files, CICS-assigned KSDS VSAM files, RRDS VSAM files, and sequential files, including flat files and partitioned data sets (PDSs) as shown in the following table.

Table 62. Access to file type by interface

Interface	VSAM			QSAM
	ESDS	KSDS/IAM	RRDS	
VSAM (read-only)	YES	YES	YES	YES
VSAM CICS (read/write)	YES	YES	YES	NO
VSAM RLS (read/write)	YES	YES	YES	YES

Using the IBM Data Virtualization Manager for z/OS Interface for VSAM, any ODBC- or JDBC-enabled application can use standard ODBC or JDBC facilities to make SQL requests to VSAM and sequential files and return a result set. No host programming is required.

The VSAM/Sequential Interface Facilities option on the Primary Option Menu provides access to the Server VSAM/Sequential Data Mapping Facility features.

Table 63. Server VSAM/Sequential Data Mapping Facility

Option	Description
Map Defaults	Set the defaults for the mapping facility
Extract VSAM	Extract from a VSAM file
Extract Seq	Extract from a Sequential file
Map Display	Display all map information
Map Copy	Copy maps
Map Refresh	Refresh maps
VSAM File Control	Displays the status of VSAM files used in the system

Using the Data Mapping Facility (DMF)

Use the Data Mapping Facility (DMF) to create data maps for VSAM and sequential file access.

Creating data maps for VSAM file access

You can extract a map from a VSAM file. These instructions also apply if you are extracting a VSAM file through CICS by selecting **CICS / CICS Data Mapping / Extract VSAM** from the Primary Option Menu.

You can extract a VSAM data map by using either of the methods:

- Using the AVZMF PAR member
- Using the DMF parser

Using the AVZMF PAR member

To extract VSAM maps in batch and to extract VSAM maps with alternate indexes, run the AVZMF PAR member that is located in your *hlq.SAVZCNTL* data set as sample JCL. A compiled listing is required to perform the extract. Also, a COBOL listing with OPT(FULL) cannot be processed to produce a map. Keywords for this process define the same elements that you specify on the ISPF panels.

Note: Perform a Map Refresh before updating the display with the IBM Data Virtualization Manager for z/OS display map command.

For more information about the VSAM parameters that are located in the AVZMFPAR member, see Using batch JCL jobs section.

Using the DMF parser

The DMF parser is used to extract a VSAM data map using the DMF parser.

About this task

Perform the following steps to extract a VSAM data map:

Procedure

1. From the Primary Option Menu, select **VSAM/Sequential** and press Enter.
2. From the **VSAM/Seq Data Mapping Facility** panel, select **Map Defaults** and press Enter.
3. Make sure the Parser Version is set to N (New).
4. Return to the VSAM/Seq Data Mapping Facility Display.
5. Select **Extract VSAM** from this menu and press Enter.
6. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map being created.
 - **Start Field:** The field name where the map starts building.
 - **End Field:** The field name where the map stops building. If not specified, the first field that is at the same level as the Start Field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set if possible.
 - **Use Offset Zero:** If the Start Field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.
 - **Flatten Arrays:** Determines whether arrays are flattened. Valid values depend on the product:
 - For IBM Data Virtualization Manager for z/OS SQL, specify C (COMPATIBLE) or Y (YES).
 - For IBM Data Virtualization Manager for z/OS Streams, specify C (COMPATIBLE) only.
 - For IBM Data Virtualization Manager for z/OS SQL 92 Engine, specify C (COMPATIBLE), Y (YES), or N (NO).

Note: The C (COMPATIBLE) value is provided for backward compatibility with an older mapping architecture. When C is specified, OCCURS fields are flattened in the map and OCCURS DEPENDING ON fields generate an error message.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem.
7. Press Enter. The system displays the VSAM Extract panel.
8. Provide the following information:
 - **For read-only VSAM files allocated to the Data Virtualization Manager server address space:** In the VSAM DSN field, type the VSAM data set name (DSN) for the data that you want to access. The DSN is dynamically allocated during the execution of the query.

Note: To create the sample VSAM file, use the sample *hlq.SAVZCNTL(DEFSTAFF)*.
 - **For READ/WRITE VSAM files via CICS:**
 - The FCT for this VSAM cluster.
 - The CICS connection name, as defined in the Data Virtualization Manager server Initialization EXEC.

- The mirror transaction name or the transaction ID, as defined in CICS.
 - The name of the Post-Read Exit and Pre-Write Exit routines if you are using the exit processing feature.
 - Type Y or N to indicate whether to use alternate indexes for this file. For this example, specify Y (Yes).
9. Press Enter. If you selected Y for alternate indexes, the VSAM Extract panel appears.
 10. Specify the name of up to eight alternate indexes and press Enter. If the extract completes with no errors, the Extract Successful message appears on the panel.
 11. Select **Map Refresh** from the **VSAM/Seq Data Mapping Facility** menu to refresh the data maps.

Using alternate indexes for a VSAM cluster

The IBM Data Virtualization Manager for z/OS Interface for VSAM supports VSAM alternate indexes by defining a data map that contains the following items:

About this task

- For read-only VSAM files allocated to the Data Virtualization Manager server address space, the data map is the path name in the base VSAM cluster.
- For read/write access to VSAM files by using CICS, the data map is the base cluster ID and an alternate index path ID as known to CICS.

The DMF allows for the same or different views in a VSAM file by changing the map name.

Procedure

1. From the Primary Option Menu, select **VSAM/Sequential** and press Enter.
2. From the VSAM/Seq Data Mapping Facility panel, select **Extract VSAM** and press Enter.
3. Provide the following information:
 - **Listing Library:** Type the information for the listing, including values for the Project, Group, Type, and Member fields. Alternatively, you can use the **Other Partitioned Data Set Containing Listing** field to specify the data set.
 - **Map Library:** Type the information for the map output data set, including values for the Project, Group, Type, and Member fields. Alternatively, you can use the **Other Partitioned Data Set To Contain Map** field to specify another data set for the map output.
4. Provide the following information in the Listing Search Criteria fields:
 - **Start Search Field:** This is used to search the listing data set for the starting point of the language-dependent data declaration. The search criteria must be unique enough to find the specific declaration to be mapped. For best results, use the fully qualified name of the declaration as it appears in the listing.
 - **End Search Field:** If this field is blank, extraction starts with the level number of the line found and continues until an equal or higher level is processed. If you enter a value in this field, extraction continues until the ending search string is found in the listing.
 - **Offset Zero:** (Y/N) Indicates whether to set the Start Search Field offset to zero, even if it is not a group level or the first definition in a group.
5. Press Enter. The system displays the VSAM Extract panel.
6. Indicate whether to use alternate indexes for this file. Specify Y (Yes) to allow the use of alternate indexes on this file.
7. Indicate whether to treat this file as an IAM file. The default is N (No); however, if the file is an IAM file, it is still treated as an IAM file.
8. Press Enter. The system displays the following panel.
9. Provide the following information:

- For read-only VSAM files allocated to the Data Virtualization Manager server address space, the path name of the VSAM alternative index. You can add up to 10 alternative index names.
- For read/write access to VSAM files by using CICS, the path name of the VSAM alternative index and the CICS FCT name. You can add up to 8 alternative index names.

Defining multiple VSAM logical records in the same file

If you are using the IBM Data Virtualization Manager for z/OS Interface for VSAM support of multiple logical records in the same file, you must define different views in the VSAM file. You create different maps that contain a different view for each of the logical records.

The following examples show two logical records from two different views in the same VSAM file. One view contains demographic information, and a second view contains account information. The RECORD_TYPE column specifies the view that contains the record.

Normally, a COBOL application that reads this data reads the record's content by using a record type (or view) indicator and then uses the redefinition of the record layout. If the COBOL program uses a redefine of the data area, the data map that is extracted contains the redefined columns. The application checks the content of RECORD_TYPE and uses the appropriate columns to view the data.

An alternative to this approach is to define the views in two separate data mapping definitions. Both data maps refer to the same file, but each has a different table name to distinguish its view in the VSAM data set. Using the preceding example, the data map DEMOGRAF can contain definitions for ACCOUNT_NUMBER, RECORD_TYPE, NAME, and ADDRESS. The data map ACCOUNT can contain ACCOUNT_NUMBER, RECORD_TYPE, and ACCOUNT_BALANCE. The application can issue the following queries to obtain all rows (records) in each view:

```
SELECT * FROM DEMOGRAF WHERE RECORD_TYPE = 1
SELECT * FROM ACCOUNT WHERE RECORD_TYPE = 2
```

To alternate the views, the application can run the following statements, where the &VALUE information is substituted from the previous query ACCOUNT_NUMBER column:

```
SELECT * FROM DEMOGRAPH WHERE RECORD_TYPE = 1
SELECT * FROM ACCOUNT WHERE ACCOUNT_NUMBER = "&VALUE" AND RECORD_TYPE = "2"
```

Creating data maps for sequential file access

You need to define a sequential file before you can access sequential files. You can define and extract this map by using either of the following methods:

- Using the AVZMF PAR member
- Using the DMF parser

Using the AVZMF PAR member

The Data Virtualization Manager server ISPF panel allows you to create data maps for sequential file access using AVZMF PAR.

To extract sequential maps in batch and to extract sequential maps with alternate indexes, run the AVZMF PAR member that is located in your *hlq*.SAVZCNTL data set as sample JCL. A COBOL listing with OPT(FULL) cannot be processed to produce a map. Keywords for this process define the same elements that you specify on the ISPF panels.

Note: You must perform a Map Refresh before it shows in the IBM Data Virtualization Manager for z/OS display map command.

For more information about the sequential parameters that are located in the AVZMF PAR member, see Using batch JCL jobs section.

Using the DMF parser

The Data Virtualization Manager server ISPF panel allows you to create data maps for sequential file access using DMF parser.

About this task

To extract a sequential data map using the DMF parser, follow these steps.

Procedure

1. From the Primary Option Menu, select **VSAM/Sequential** and press Enter.
2. From the VSAM/Seq Data Mapping Facility panel, select **Extract Seq** and press Enter.
3. Provide the following information:
 - **Source Library Name:** The data set name and member name that contain the source code for the map you want to create.
 - **Start Field:** The field name that is used to start building the map.
 - **End Field:** The field name that is used to stop building the map. If not specified, the first field that is at the same level as the Start field stops the build process.
 - **Map Name:** The name of the map in the DMF. This name also is used as the member name for the map in the mapping data set if possible.
 - **Use Offset Zero:** If the Start field is not an '01' level, start the offset at zero; otherwise, the offset starts at the offset of the field in the structure.
 - **Flatten Arrays:** Determines whether arrays are flattened. Valid values depend on the product:
 - For Data Virtualization Manager server SQL, specify C (COMPATIBLE) or Y (YES).
 - For Data Virtualization Manager server Streams, specify C (COMPATIBLE) only.
 - For Data Virtualization Manager server SQL 92 Engine, specify C (COMPATIBLE), Y (YES), or N (NO).

Note: The C (COMPATIBLE) value is provided for backwards compatibility with an older mapping architecture. When C is specified, OCCURS fields are flattened in the map and OCCURS DEPENDING ON fields generate an error message.
 - **Map Data Set Name:** The data set name where the map is stored. The default is the first data set in the AVZMAPP DD statement for the subsystem in the server started task.
4. Press Enter. The system displays the Sequential Extract panel.
5. Provide the following information:
 - **For flat files:** The data set name in the **Enter DSN** field.
 - **For PDSs:** The data set name in the **Enter DSN** field. In addition, if you want to create a data map that includes columns for viewing or searching the data set name and/or PDS member name, provide the following information:
 - To view the data set name, in the DSN Column Name field, type a column name that represents the data set name information.
 - To view the PDS member name, in the Member Column Name field, type a column name that represents the member name information.
 - If you want to search by the data set name or PDS member name columns, specify Y (Yes) to indicate that the columns are allowed to be used in search criteria.

Note: If you do not specify the appropriate information to search by data set name or member name, a query returns information for all PDS members of all of data sets, without any indication of the corresponding member name or data set name.
6. Press Enter. If the extract completes with no errors, the Extract Successful message appears on the panel.
7. Return to the VSAM/Seq Data Mapping Facility and select **Map Refresh** to refresh the data maps.

Query syntax

The following syntax shows the query for each type of data file:

- **VSAM data (read-only)**

```
select (5) * from vsam1
```

- **VSAM for CICS data (read/write)**

```
select (5) * from filea
```

- **Sequential files**

```
select * from flatfile
```

Using a CALL statement to obtain map metadata

The IBM Data Virtualization Manager for z/OS Interface for VSAM and Sequential Files allows users to view metadata information for VSAM or sequential file data maps on the client with a simple CALL statement. The syntax of the call is:

```
CALL DVS_MAP('DESCRIBE','mapname')
```

where *mapname* is the name of the map.

This call returns a result set with a single column named FORMAT. The FORMAT column contains details on the fields of the map. The FORMAT column types and their SQL equivalents are shown in the following table.

FORMAT column types	SQL equivalent
CHARACTER	SQL_CHARACTER
NUMERIC	SQL_NUMERIC
DECIMAL	SQL_DECIMAL
INTEGER	SQL_INTEGER
SMALLINT	SQL_SMALLINT
FLOAT	SQL_FLOAT
DOUBLE	SQL_DOUBLE
DATE	SQL_DATE
TIME	SQL_TIME
TIMESTAMP	SQL_TIMESTAMP
VARCHAR	SQL_VARCHAR
LONGVARCHAR	SQL_LONGVARCHAR
BINARY	SQL_BINARY
VARBINARY	SQL_VARBINARY
LONGVARBINARY	SQL_LONGVARBINARY
UNICODE	SQL_UNICODE
UNICODE_VARCHAR	SQL_UNICODE_VARCHAR
UNICODE_LONGVARCHAR	SQL_UNICODE_LONGVARCHAR

Using the Data Mapping Facility

You can use the Data Mapping Facility to set default maps to display, copy, or refresh data maps, to view individual items in a data map, to generate RPC skeletons, and to create source library definitions.

The Data Mapping option on the Primary Option Menu provides access to the Data Mapping Facility features.

Option	Description
Map Defaults	Set the defaults for the mapping facility
Map Display	Display all map information
Map Copy	Copy maps
Map Refresh	Refresh maps
VSAM File Control	Displays the status of VSAM files used in the system
Source Library Management	View or create source library definitions

Setting default values for data maps

You can use the IBM Data Virtualization Manager for z/OS ISPF panel to set the default values for the data maps.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Map Defaults** and press Enter.
3. Type Y (Yes) or N (No) for Auto Refresh. Press Enter.

Y means that the storage data maps are automatically refreshed after changes.

N requires a manual refresh by using the Map Refresh option.

Auto Refresh can incur significant overhead if you have several changes to make and you exit after each change. Either make all changes before exiting, or turn off Auto Refresh and use the Map Refresh option when you are finished.

If you set this value to Y, you do not need to perform a Map Refresh before the HTML generation. If you set it to N, you must perform a Map Refresh before and after the HTML generation.

Results

The Profile Saved message appears, indicating that the data set name is saved in the user profile pool.

Displaying data maps

You can view the data maps in detail using the IBM Data Virtualization Manager for z/OS ISPF panel.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Map Display** and press Enter.

The **Schema Listing** panel displays.

3. Type **X** next to the schema to view the **Data Mapping Block** panel for the schema.

The **Data Mapping Block** panel displays.

4. Use the available line commands to perform the appropriate functions. The following commands are available:

- P – Prints map
- S – Shows map
- D – Disables map
- E – Enables map
- K – Deletes map
- X – Displays map

Type the command name and press Enter.

Results

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column name	Description	Sort name
STRUCTURE NAME	The member names in the map data set.	NAME
TYPE	TYPE <ul style="list-style-type: none"> • ADABAS • Input • Output • Screen • LPTBL • Header • USER 	TYPE
STATUS	The status of the map (Enabled, Disabled, or Deleted).	STATUS
MR	Map Reduce (Yes/No)	MR
LANGUAGE	The language of the extracted map. This value is determined at the time of the extract.	LANGUAGE
AT	Attachments (OPDWs) present in the map (Yes/No)	AT
MODIFICATION DATE TIME	The date and time the map was last modified.	DATE
USERID	The user ID of the map creator. Used only for informational purposes.	USERID
NOTE	Comments	

Viewing individual data elements

The data element page displays information about the data map for a specific program.

About this task

To display the contents of a data map, use the following instructions.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Map Display** and press Enter.
The **Schema Listing** panel displays.
3. Type X next to the schema to view the **Data Mapping Block** panel for the schema.
The **Data Mapping Block** panel displays.
4. Type X next to the structure to view individual data elements of that structure. Press Enter.
The system displays the Data Elements for the structure.
5. Use the available line commands to perform the appropriate functions. Available commands:
 - P – Prints map
 - S – Shows map
 - D – Disables map
 - E – Enables map
 - C – Changes mapType the command name and press Enter.

Results

The following table describes each column name on the ISPF panels and provides a sort name (if available).

Column Name	Value	Description
FIELD NAME	1-50 characters	The name of the field.
COLUMN NAME	1-18 characters	The name of the column heading. During map extract, column names were created using the field names and translating any dash characters to underscores. The map editor can be used to make column names more meaningful for users.
STATUS	<ul style="list-style-type: none">• Enabled• Disabled	The status of the map.
LEVEL	1-nnn	The level in relation to other elements. This is maintained for informational purposes only.
LENGTH	1-65635	The length of the data element.
FORMAT	<ul style="list-style-type: none">• Char• Bin• Packed• Decimal• Date, Time• Group	The format of the data element.

Column Name	Value	Description
OFFSET	1-65635	An offset is maintained as the relative position 0 (zero) displacement from the beginning of the structure.
PRECISION	1-65635	The element precision.
SCALE	1-65635	The element scale.
LINKED STRUCTURE	1-8 characters	The related structure name.
LINKED COLUMN	1-32 characters	The related structure column name.
FILL CHAR	1 character	The default fill character.
FILL DATA	1-200 characters	The default fill data.
ORIGINAL STATEMENT	1-80 characters	The originating statement from which the elements were extracted. For items that were entered using the editor, these are not available.

Copying data maps

Data maps may be copied, or copied then edited to create new maps.

About this task

Use the following instructions to copy data maps.

Note: We recommend using the batch migration jobs for moving maps from one environment to another.

Procedure

- From the Primary Option Menu, select **Data Mapping** and press Enter.
- From the **Data Mapping Facility** menu, select **Map Copy** and press Enter.
The **Move/Copy Utility** panel displays.
- Use the available line commands to perform the appropriate functions.
 - C – Copy
 - CP – Copy and Print
 - M – Move
 - MP – Move and Print

Type the command name and press Enter.
- Project, Group, and Type are used for source code management. In the From ISPF Library fields, specify the following information:
 - Project
 - Group
 - Type
 - Member (if the data set is partitioned). You can perform the following actions:
 - To move, copy, or promote a single member, type the member name.
 - To move, copy, or promote all members, type * (an asterisk).
 - To request a member selection list, leave the member name blank or specify a pattern

Alternatively, for other partitioned or sequential data sets, you can specify the From Other Partitioned or Sequential Data Set field. Type the data set name and volume serial (volume serial number).

Note: If you do not enter a correct password for a data set that requires one, the system prompts you in standard TSO (line) mode. On TSO/TCAM systems, it may be necessary to press the CLEAR key before you respond to the password prompt. If you enter the password incorrectly or encounter other problems, you may be prompted again to enter the password until you reach a system limit of attempts.

Press Enter.

Refreshing data maps

Refreshing a data map may be required when changes to the underlying data structure occur. When you refresh a map, the Data Mapping Facility checks the library for modifications, and then refreshes in-core map tables from the library.

About this task

Use the following instructions to refresh a data map.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Map Refresh** and press Enter.

If the refresh is completed with no errors, the Refresh Successful message appears on the Server Mapping Facility options menu.

Generating RPC skeletons

This option generates RPC programs from an extracted data map by generating the SQLBINDCOL statements in a new PDS member by using the skeleton program that is provided in the same partitioned data set. The skeleton program contains all the language and application-specific code that is required to perform the RPC task. Substitute your information for the required keywords, and write the new specified member.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Generate an RPC skeleton** and press Enter.

The **RPC Generation Facility** panel displays.

3. Specify the data set information for the following fields:

- Map library
- RPC library
- Skeleton library

Press Enter to generate.

Example Cobol Program

The following program is a sample of the skeleton program that gets generated. Note the commands that begin with the '@' character.

```
CBL APOST
010010 IDENTIFICATION DIVISION.
010020 PROGRAM-ID. DFSSAM02.
010080 ENVIRONMENT DIVISION.
010090 CONFIGURATION SECTION.
010100 SOURCE-COMPUTER. IBM-370.
010110 OBJECT-COMPUTER. IBM-370.
010120 DATA DIVISION.
```

```

010130 WORKING-STORAGE SECTION.
COPY SBCPHD.
77 SDF-RETURN-CODE PIC S9(05) VALUE 0.
77 STATEMENT-HANDLE USAGE IS POINTER.
77 SQL-PRECISION PIC S9(5) COMP VALUE 0.
77 SQL-SCALE PIC S9(5) COMP VALUE 0.
77 SQL-COLUMN-LEN PIC S9(5) COMP VALUE 1.
77 SQL-COLUMN-NAME-LEN PIC S9(5) COMP.
77 SQL-COLUMN-NUMBER PIC S9(5) COMP.
77 SQL-COLUMN-NAME PIC X(30).
77 SQL-COLUMN-TYPE PIC S9(5) COMP.
77 ERROR-MESSAGE-AREA PIC X(256) VALUE IS SPACES.
77 TRACE-MESSAGE-AREA PIC X(256) VALUE IS SPACES.
77 STRING-PTR PIC S9(5) COMP VALUE IS 1.
77 CONNECTION-HANDLE USAGE IS POINTER.
77 ENVIRONMENT-HANDLE USAGE IS POINTER.
77 ERROR-MSG-LENGTH-AREA PIC S9(5) COMP VALUE 0.
77 NATIVE-ERROR-CODE-AREA PIC S9(5) COMP VALUE 0.
77 SQLSTATE-DATA-AREA PIC X(6) VALUE IS SPACES.
@DATABUFFER
060110 LINKAGE SECTION.
080010 PROCEDURE DIVISION.
080020 INIT.
@SQLBINDCOL BEGIN
MOVE @LENGTH TO SQL-COLUMN-LEN.
MOVE @COLUMN_NAME_LENGTH TO SQL-COLUMN-NAME-LEN.
MOVE @COLUMN_NAME TO SQL-COLUMN-NAME.
MOVE @TYPE TO SQL-COLUMN-TYPE.
MOVE @SEQ TO SQL-COLUMN-NUMBER.
MOVE @PRECISION TO SQL-PRECISION.
MOVE @SCALE TO SQL-SCALE.
CALL 'SDCPBC' USING STATEMENT-HANDLE
SQL-COLUMN-NUMBER
SQL-C-DEFAULT
SQL-COLUMN-TYPE
SQL-PRECISION
SQL-SCALE
SQL-NO-NULLS
@FIELD_NAME
SQL-COLUMN-LEN
SQL-COLUMN-NAME
SQL-COLUMN-NAME-LEN.
MOVE RETURN-CODE TO SDF-RETURN-CODE.
IF SQL-INVALID-HANDLE OR SQL-ERROR OR SQL-NO-DATA-FOUND
PERFORM 0000-ERROR-ROUTINE
END-IF.
@SQLBINDCOL END
CALL 'SDCPH' USING STATEMENT-HANDLE SQL-THROW-DONE.
MOVE RETURN-CODE TO SDF-RETURN-CODE.
IF SQL-INVALID-HANDLE OR SQL-ERROR OR SQL-NO-DATA-FOUND
PERFORM 0000-ERROR-ROUTINE THRU 0000-ERROR-EXIT
END-IF.
080140 EXIT-RTN.
080160 GOBACK.
0000-ERROR-ROUTINE.
MOVE 256 TO SQL-PRECISION.
IF SQL-INVALID-HANDLE GO TO 0000-ERROR-EXIT.
*****
* IF AN ERROR OCCURS CALL THE SQLERROR ROUTINE
*****
CALL 'SDCPSE' USING ENVIRONMENT-HANDLE CONNECTION-HANDLE
STATEMENT-HANDLE SQLSTATE-DATA-AREA
NATIVE-ERROR-CODE-AREA
ERROR-MESSAGE-AREA
SQL-COLUMN-LEN ERROR-MSG-LENGTH-AREA.
MOVE RETURN-CODE TO WS-ODBCAPI-RETURN-CODE.
IF SQL-SUCCESS OR SQL-SUCCESS-WITH-INFO
PERFORM 0000-ERROR-DISPLAY-ROUTINE THRU
0000-ERROR-DISPLAY-EXIT.
0000-ERROR-EXIT.
0000-ERROR-DISPLAY-ROUTINE.
*****
* SEND THE ERROR MESSAGE TO THE CLIENT USING SQLRETURNSTATUS
*****
STRING 'HOST ERROR MESSAGE - ' ERROR-MESSAGE-AREA
DELIMITED BY SIZE INTO TRACE-MESSAGE-AREA WITH
POINTER STRING-PTR
END-STRING.
CALL 'SDCPRS' USING CONNECTION-HANDLE TRACE-MESSAGE-AREA
SQL-NTS NATIVE-ERROR-CODE-AREA.
0000-ERROR-DISPLAY-EXIT.

```

Program Explanation

- The following statement causes the facility to substitute the originally extracted statements in the program at the location where the statement is found:

```
@DATABUFFER
```

- The following statements declare the beginning and ending of the SQLBINDCOL substitution. All of the statements between the begin and end are replicated for the number of ENABLED fields in the map data.

```
@SQLBINDCOL BEGIN  
@SQLBINDCOL END
```

- The following keywords may be contained between the SQLBINDCOL BEGIN and SQLBINDCOL END statements. These keywords are substituted with the correct values for each **ENABLED** field in the data map.

```
@LENGTH - the length of the field element  
@COLUMN_NAME_LENGTH - the length of the column name.  
@COLUMN_NAME - the column name used to identify the field  
@TYPE - SQL data type of column data. All DB2 SQL data types are supported  
except for graphic (DBCS) data.  
@SEQ - a sequentially assigned number for this column  
@PRECISION - the precision of the field  
@SCALE - the scale of the field  
@FIELD_NAME - the field name itself as defined in the @DATABUFFER -
```

Considerations

The skeleton can contain as many or as few statements as needed. It does not have to be a complete program, and you do not have to use all keywords.

For example, a skeleton member that contains the following statements generates a list of **ENABLED** field names as defined in the data map:

```
@SQLBINDCOL BEGIN  
@FIELD_NAME  
@SQLBINDCOL END
```

Initializing catalogs

You can create data maps that represent standard IBM Data Virtualization Manager for z/OS catalog tables. These tables are used when the call wrapper is eliminated in the ODBC and JDBC drivers. They are also used to generate the metadata information that determines access to the data.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Initialize catalog** and press Enter.
The **Catalog Extract** panel displays.
3. Specify the map data set information:

- Project
- Group
- Type

Alternatively, you can use the **Other Map Data Set Name** field to specify the map data set.

Press Enter to perform the catalog extract. The **Catalog Entries Defined** message is displayed on the panel.

4. Use the END command (or press F3) to return to the **Data Mapping Facility** panel.

5. Select the **Map Refresh** option. Press Enter.

The system displays the maps. In the LANGUAGE column, those maps with the value 'CATALOG' in the language value represent IBM Data Virtualization Manager for z/OS catalog tables.

Note: In the STRUCTURE NAME column, the following entries contain the value 'CATALOG' in the LANGUAGE column:

- COLUMNS
- FOREIGNK
- PRIMARYK
- SPECIALC
- STATISTI
- TABLES

Specifying catalog names on metadata calls

If you are using WebMethods, specify catalog names on metadata calls for ADABAS, IMS/SQL, VSAM, and/or CICS VSAM.

About this task

To specify a catalog name on metadata calls, perform the following:

Procedure

1. Add the parameters you want to use to the Data Virtualization Manager configuration member, AVZSIN00, using the MODIFY PARM command:

```
MODIFY PARM NAME(CATADABAS) VALUE(NULL)
MODIFY PARM NAME(CATSQLIMS) VALUE(NULL)
MODIFY PARM NAME(CATVSAM) VALUE(NULL)
MODIFY PARM NAME(CATVSAMCICS) VALUE(NULL)
MODIFY PARM NAME(POPULATECATNAME) VALUE(YES)
```

Change the value of the POPULATECATNAME parameter from NO (the default) to YES.

2. Refresh the catalog maps TABLES and COLUMNS3. You can do this using one of the following methods:
 - Use the Initialize Catalog option from the **Data Mapping Facility** panel.
 - Copy the maps from the distribution map library.

Creating source library maps

You can create new source library maps.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Server Data Mapping Facility** panel, select **Source Library Management** and press Enter.
3. From the **Data Mapping Facility** menu, select **Create Source Library Map** and press Enter.
The **Source Library Management** panel displays.
4. Enter information for the Source Library Definitions.
 - Type the name of the list of Source Libraries.
 - Enter Y or N to specify whether to Replace an existing definition.
 - Enter information to specify the Data Set Source Library or Natural Source Library. Data Set supports all libraries except Natural.

Press Enter.

The system displays the **Source Library** panel and shows the new source library map.

Results

The following table describes each column name on the ISPF panels.

Column name	Description
NAME	The source library name.
DESCRIPTION	A description of the source library.
REPLACE	Replace an existing definition. Yes or No.
DATA SET NAME	The name of the PDS/Sequential file that contains the source code.
NATURAL LIBRARY	The name of the Natural library.
ADABAS DBID	The Adabas database ID.
ADABAS FILE	The FUSER or FDIC file number.
SERVER TYPE	The generic ACI server to run query: <ul style="list-style-type: none">• B — BATCH• C — CICS

Displaying source library maps

You can view current source library maps.

Procedure

1. From the Primary Option Menu, select **Data Mapping** and press Enter.
2. From the **Data Mapping Facility** menu, select **Display Source Library Map** and press Enter.
The **Source Library** panel displays.
3. Use the available line commands to perform the appropriate functions. Available commands:
 - P — Prints map
 - S — Shows map
 - D — Disables map
 - E — Enables map

Type the command name and press Enter.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*
 - *z/OS TSO/E Primer*
 - *z/OS TSO/E User's Guide*

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Product legal notices

This information was developed for products and services offered in the U.S.A.

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.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give 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 Director of Licensing
IBM Corporation
North Castle Drive

Armonk, NY 10504-1785
U.S.A.

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 information 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.

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.

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.html>.

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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 <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

- accessibility
 - overview [453](#)
- accessing data [390](#)
- accessing IDF from DB2I (SPUFI) [226](#)
- accessor environment element), *See* ACEE
- ACEE
 - deletion [233](#)
 - retention [233](#)
- ACF2
 - defining resources [248](#)
- ACI server map
 - defining [391](#)
 - displaying information [399](#)
 - extracting information [397](#)
- ACIMAPREDUCETRACKS [152](#)
- Adabas
 - configuring [101](#)
 - Data Virtualization Manager configuration [102](#)
- ADABAS
 - specifying catalog names on metadata calls [450](#)
- all authorization events [33](#)
- analytics
 - configure [157](#)
- ANSI SQL data access [179](#)
- APPC/MVS
 - interval summary [347](#)
 - records [346](#)
- authorization events
 - all events [33](#)
 - AVZ command [50](#)
 - communication link [40](#)
 - control block [36](#)
 - database [37](#)
 - global variable events [38](#)
 - IMSLTERM [39](#)
 - log off events [41](#)
 - log on events [43](#)
 - MQ [48](#)
 - parameter events [49](#)
 - RPC [49](#)
 - SEF command [51](#)
 - token events [53](#)
 - TSO command [54](#)
 - user events [55](#)
- AVZ command authorization events [50](#)
- AVZ.SERVICES table [353](#)
- AVZ.STREAMS [355](#)
- AVZECURE API function [91](#)
- AVZINFO API function [89](#)
- AVZSUBMIT API function [94](#)
- AVZVALUE API function [83](#)

B

- batch

- batch (*continued*)
 - copying maps [389](#)
 - creating maps [389](#)
 - extracting maps [389](#)
- batch extract member [389](#)
- batchmember
 - batch extract [389](#)
- Big SQL
 - configure [125](#)
- block fetch [280](#)

C

- CA IDMS
 - configuring [162](#)
 - modifying server configuration member [142](#), [163](#)
 - verify access [164](#)
- CA Top Secret
 - defining resources [248](#)
- call attachment facility (CAF) [205](#)
- catalog names for metadata calls [450](#)
- CICS
 - configure [156](#)
 - security [253](#)
 - support for distributed transactions [209](#)
- CICS failover
 - enabling [280](#)
- CICS VSAM
 - specifying catalog names on metadata calls [450](#)
- classification rules [273](#)
- client response time [364](#)
- command events [57](#)
- communication link authorization events [40](#)
- configuration
 - support for distributed CICS and IMS transactions [209](#)
 - support for distributed Db2 transactions [206](#)
 - support for distributed Db2 transactions with Microsoft Transaction Server [207](#)
- configuring
 - Adabas [101](#)
 - ADDI [165](#)
 - CA IDMS [162](#)
 - data sources [101](#)
 - Data Virtualization Manager Service Provider on zcEE [191](#), [196](#)
 - IBM MQ [141](#)
 - Innovation Access Method (IAM) [284](#)
 - mixed case password for RACF [233](#)
 - RAA [171](#)
 - relational database management systems (RDBMS) [103](#)
 - security access to Adabas data [103](#)
 - server advanced security [243](#)
 - z/OS Connect [191](#)
- Configuring
 - DB2 for Distributed Relational Database Architecture (DRDA) [111](#)

- Configuring (*continued*)
 - Db2 for Resource Recovery Services attachment facility (RRSAF) [113](#)
 - Resource Recovery Services attachment facility (RRSAF) [111](#)
 - server started task JCL [105](#)
- configuring AT-TLS [238](#)
- configuring CA IDMS
 - modifying server configuration member [142](#), [163](#)
- configuring Data Virtualization Manager server for IDF [220](#)
- configuring IMS
 - modifying server configuration member [137](#)
 - server started task JCL [134](#)
- configuring NoSQL
 - modifying the Data Virtualization Manager configuration member [181](#)
- configuring System Management Facility (SMF) files [158](#)
- Configuring VSAM Control Dataset [146](#)
- connections
 - limiting [15](#)
 - queuing [15](#)
 - rejecting [14](#)
- control block authorization events [36](#)
- controlling information access
 - TRACEDATA resource [251](#)
- cookie policy [455](#), [457](#)
- copying maps [389](#)
- CPU time limits [257](#)
- creating a db2 RESTful service using IDF [224](#)
- creating maps [389](#)

D

- dashDB
 - configure [126](#)
- data
 - accessing [385](#)
 - virtualization [385](#)
- data map
 - generating RPC programs [447](#)
- Data Mapping Facility
 - displaying data maps [443](#)
- data sources
 - CA IDMS [162](#)
 - configuring [101](#)
 - configuring Adabas [101](#)
 - configuring relational database management systems (RDBMS) [103](#)
 - IBM MQ [141](#)
 - sequential (using ADDI) [165](#)
 - sequential (using RAA) [171](#)
 - VSAM (using ADDI) [165](#)
 - VSAM (using RAA) [171](#)
- Data Virtualization Manager Service Provider on zcEE [191](#)
- database authorization events [37](#)
- Db2
 - configure the started task [105](#)
- Db2 data access [115](#)
- Db2 Direct [115](#)
- db2 mainframe applications
 - IDF [219](#)
- DB2 RESTful services provisioning [224](#)
- DB2 SQL failures [348](#)
- Db2 unload data set, configuring access [114](#)

- DBCTL
 - configure server configuration [136](#)
- DISABLEATTACH parameter [132](#)
- distributed CICS and IMS transactions [209](#)
- distributed databases
 - configure [119](#)
- distributed Db2 transactions
 - Microsoft Transaction Server [207](#)
- Distributed Relational Database Architecture (DRDA)
 - configuring access [105](#), [120](#)
- distributed transactions
 - configuring for CICS/TS and IMS [208](#)
 - configuring for Db2 [205](#)
- DSCLIENAUXTSTGCUTOFF parameter [261](#)

E

- enabling support [234](#)
- End of Session records [352](#)
- enterprise auditing
 - enabling [243](#)
- enterprise transactions
 - CICS/TS [208](#)
 - Db2 [205](#)
 - IMS [208](#)
- ERLY Server [145](#)
- Error log records [347](#)
- Event Facility
 - setting connection limits [15](#)
- events
 - authorization [33](#)
 - command [57](#)
 - configuring rules for [31](#), [33](#)
 - exception [59](#)
 - global variable [73](#)
 - remote procedure call [74](#)
 - SQL [75](#)
 - time-of-day [76](#)
 - virtual table [77](#)
- exception events [59](#)
- extended IDs
 - host side support [242](#)
- external CPU time limits [257](#)

F

- FIND command [298](#)

G

- generic IDs
 - host side support [242](#)
- Getting started [1](#)
- global variable authorization events [38](#)
- global variable events [73](#)

H

- host commands
 - DISPLAY [97](#)
- host side support [242](#)

I

- IBM Application Discovery and Delivery Intelligence
 - authentication [170](#)
 - configuration [166](#), [169](#), [170](#)
 - configuring [165](#)
 - rules [170](#)
- IBM Db2 for z/OS
 - configure access to data [104](#)
- IBM MQ
 - configuration [142](#)
 - configuring [141](#)
 - rules [142](#)
- IBM Rational Asset Analyzer
 - authentication [176](#)
 - configuration [172](#), [173](#), [175](#), [176](#)
 - configuring [171](#)
 - rules [176](#)
- IBM z/OS Connect Enterprise Edition server
 - instance
 - creating [193](#), [194](#)
- IDF
 - Creating a db2 RESTful service [224](#)
 - db2 mainframe applications [219](#)
 - DB2 RESTful services provisioning [224](#)
 - known restrictions and limitations [219](#)
 - populating db2 communications database [221](#)
 - populating the SYSIBM.IPNAMES table [222](#)
 - populating the SYSIBM.LOCATIONS table [221](#)
 - populating the SYSIBM.USERNAMES table [223](#)
 - using IDF for Mainframe Applications to implement DB2 RESTful services [224](#)
- IDs
 - extended [241](#)
 - generic [241](#)
- IMS
 - APPC environment requirements [208](#)
 - configuring server started task JCL [134](#)
 - OTMA environment requirements [208](#)
 - SQL access to database [425](#)
 - support for distributed transactions [209](#)
- IMS catalog access [134](#)
- IMS database
 - configure access to data [134](#)
- IMS Direct
 - modifying server configuration member [137](#)
- IMS/SQL
 - specifying catalog names on metadata calls [450](#)
- IMSLTERM authorization events [39](#)
- Innovation Access Method (IAM)
 - configuring [233](#), [284](#)
- Instrumentation Server
 - installing [367](#)
 - sysplex [368](#)
- Integrated DRDA Facility (IDF) [212](#), [216](#), [219](#)
- interface
 - ACI [390](#)
 - Adabas [412](#)
 - CICS/TS [413](#)
 - DB2 [423](#)
 - Sequential [437](#)
 - VSAM [437](#)
- Interface for ACI [390](#)
- Interface for Adabas [412](#)

- Interface for CICS/TS [413](#)
- Interface for DB2 [423](#)
- Interface for VSAM and Sequential [437](#)
- internal CPU time limits [257](#)
- Interval records [340](#)
- ISPF load modules
 - optionally restrict [249](#)

J

- JCL
 - configuring [101](#), [142](#), [163](#)
 - started task [101](#), [142](#), [163](#)

L

- legal notices
 - cookie policy [455](#), [457](#)
 - notices [455](#)
 - programming interface information [455](#)
 - trademarks [455](#), [456](#)
- link security [254](#)
- Linux, UNIX, and Windows databases
 - configure [127](#)
- load balancing
 - enabling for CICS/TS [278](#)
 - enabling for group director [277](#)
 - enabling for Services [278](#)
- LOCATE command [295](#)
- log entries
 - filtering [295](#)
 - finding character strings in the log [298](#)
 - labeling [298](#)
 - locating [295](#)
 - printing [299](#)
 - profiles for filtering [295](#)
 - viewing [290](#)
- log off authorization events [41](#)
- log on authorization events [43](#)
- logging
 - enable [336](#)
- logging tables
 - APPC/MVS interval summary [347](#)
 - AVZ.ERRORLOG [348](#)
 - AVZ.INTERVALS [341](#)
 - AVZ.SESSIONS [338](#)
 - AVZ.SQLSOURCE [343](#)
 - AVZ.STORAGE [345](#)
- logging to DB2 tables [335](#)
- login security [253](#)
- logoff [232](#)
- logon [232](#)

M

- managing user connections [10](#)
- managing users [10](#)
- Map Migration utility [383](#)
- mapping
 - batch [385](#)
 - IBM Data Virtualization Manager studio [385](#)
 - ISPF [385](#)
- MapReduce

- MapReduce (*continued*)
 - Innovation Access Method (IAM) [284](#)
 - metadata repository [285](#)
 - Virtual Parallel Data [282](#)
- maps
 - copying [446](#)
 - creating source library maps [450](#)
 - displaying source library maps [451](#)
 - initializing catalogs [449](#)
 - refreshing [447](#)
 - setting default [443](#)
 - viewing individual data elements [444](#)
- metadata repository
 - creating [285](#)
- Microsoft SQL Server
 - configure [128](#)
- migrating maps [383](#)
- mode that Data Virtualization Manager configuration is running [294](#)
- modifying parameter [5](#)
- modifying the client auxiliary storage cut-off parameter [261](#)
- modifying the data buffer for VSAM files [144](#)
- modifying the index buffer for VSAM files [144](#)
- MongoDB [181](#)
- MongoDB interface examples
 - C# [184](#)
 - Eclipse BIRT [185](#)
 - JavaScript [183](#)
 - NoSQL [182](#)
- MongoDB query language [186](#)
- monitoring
 - client response time [364](#)
- monitoring events [363](#)
- MQ resource authorization events [48](#)
- MRO bind time security [254](#)
- MULTACC rule [152](#)
- multiple servers
 - running [211](#)

N

- native Db2 subsystem, access and display [132](#)
- netaccess [238](#), [239](#)
- network access control
 - enabling [239](#)
- NoSQL solution [181](#)
- notices [455](#)

O

- operation level records [352](#)
- Oracle Database Provider for DRDA
 - configure [130](#)

P

- parameter authorization events [49](#)
- parameter information [3](#)
- PassTickets [232](#)
- peer server
 - virtual tables [216](#)
- peer servers
 - configuring [212](#)

- peer-to-peer communication [212](#), [219](#)
- performance features [257](#)
- populating db2 communications database for IDF [221](#)
- populating the SYSIBM.IPNAMES table [222](#)
- populating the SYSIBM.LOCATIONS table [221](#)
- populating the SYSIBM.USERNAMES table [223](#)
- Primary Option Menu [9](#)
- printing log entries [299](#)
- profiles for filtering log entries [295](#)
- programming interface information [455](#)
- protected resources [243](#)

Q

- QMF DRDA Server
 - configure [131](#)
- query acceleration rule [113](#)

R

- RACF
 - defining resources [247](#)
 - PassTickets [250](#)
- record buffer [286](#)
- Record buffer [286](#)
- Record: SQL Source [343](#)
- Recoverable Resource Management Services (RRMS) [205](#)
- relational database management systems (RDBMS)
 - configuring [103](#)
- remote peer servers
 - configuring [212](#)
- remote procedure call (RPC) events [74](#)
- Remote users panel [10](#)
- report class definition
 - modifying [272](#)
- Resource Recovery Services (RRS) [208](#)
- Resource Recovery Services attachment facility (RRSAF) [205](#)
- RFIND command [298](#)
- RPC authorization events [49](#)
- RPC programs
 - generating [447](#)
 - skeleton for generating [447](#)
- rules
 - API functions for [83](#)
 - automatic limits [31](#)
 - AVZECURE API function [91](#)
 - AVZINFO API function [89](#)
 - AVZSUBMIT API function [94](#)
 - AVZVALUE API function [83](#)
 - configuring [31](#), [32](#)
 - parts of rules [31](#)
 - types of events [33](#)
 - variables [32](#)

S

- screen readers and magnifiers [453](#)
- searching the server log [298](#)
- Secure Sockets Layer [234](#)
- security
 - CICS [253](#)
 - configure [157](#)
 - defining for RPCs [250](#)

- security (*continued*)
 - resource [252](#)
 - virtual table SAF security [255](#)
- Security
 - configure [104](#)
- security jobs
 - optional [248](#)
- Security Optimization Management (SOM)
 - enabling advanced support [231](#)
 - enabling basic support [229](#)
 - using PasTickets [232](#)
- SEF command authorization events [51](#)
- sending comments to IBM [xvii](#)
- sequential files
 - configuring access [152](#)
- server configuration member
 - configuring for CA IDMS [142](#), [163](#)
 - configuring for IMS Direct [137](#)
- server event facility (SEF)
 - configure ADDI [170](#)
 - configure Big SQL [125](#)
 - configure dashDB [126](#)
 - configure Db2 unload data set access [114](#)
 - configure IBM MQ [142](#)
 - configure Linux, UNIX, and Windows databases [127](#)
 - configure Microsoft SQL Server [128](#)
 - configure Oracle Database Provider for DRDA [130](#)
 - configure QMF DRDA Server [131](#)
 - configure query acceleration [113](#)
 - configure RAA [176](#)
 - read ahead tracks [152](#)
- server log
 - SQL [294](#)
- server parameters
 - DISABLEATTACH [132](#)
 - SQLENGVSAMDATABUFF [144](#)
 - SQLENGVSAMINDEXBUFF [144](#)
- server started task JCL
 - configuring for IMS [134](#)
- Server Trace log [289](#)
- Server Trace panel columns [291](#)
- servers
 - configuring [211](#)
 - multiple [211](#)
- service class definition
 - modifying [271](#)
- Service Provider on zcEE
 - configuring [203](#)
- Services records [349](#), [351](#)
- Session records [337](#)
- SET CURRENT QUERY ACCELERATION [113](#)
- SMF
 - enabling [302](#)
 - SMFNUMBER [302](#)
- SMF logging [302](#)
- SMF Record Subtype 02 [309](#)
- SMF Record Subtype 03 [310](#)
- SMF Record Subtype 04 [312](#)
- SMF Record Subtype 05 [313](#)
- SMF Record Subtype 06 [316](#)
- SMF Record Subtype 09 [318](#)
- SMF Record Subtype 10 [319](#)
- SMF Record Subtype 11 [320](#)
- SMF Record Subtype 13 [322](#)
- SMF Record Subtype 14 [323](#)
- SMF Record Subtype 16: Record Broker Statistics [325](#)
- SMF Record Subtype 17 [327](#)
- SMF Record Subtype 18: Interval Usage Recording Options [331](#)
- SMF Record Subtype 18: Services Records [329](#)
- SMF Record Subtype 20: Transcation SMF Record [333](#)
- SMF Records
 - SMF Record Subtype 01 [306](#)
- SQL
 - CALL statement resource usage [343](#)
 - dynamic resource usage [343](#)
- SQL connection
 - log records [341](#)
- SQL entries in the server log
 - displaying [294](#)
- SQL events [75](#)
- SQL failures [348](#)
- SQL user
 - log records [338](#)
- SQLENGVSAMDATABUFF parameter [144](#)
- SQLENGVSAMINDEXBUFF parameter [144](#)
- SSL, *See* Secure Sockets Layer
- started task parameters
 - queueing connections [15](#)
 - rejecting connections [14](#)
- Storage records [345](#)
- storage use
 - by Data Virtualization Manager server [345](#)
- Stream records [355](#)
- streams
 - monitoring [365](#)
- subsystem and classification rules
 - viewing [272](#)
- SYSLOG
 - configuring [161](#)
- sysplex [368](#)
- SYSPLEX [145](#)
- System Management Facility (SMF) files
 - configuring [158](#)
- system resources management
 - detecting when sessions fail [262](#)
 - enabling block fetch [281](#)
 - enabling program execution duration time limit mechanism [260](#)
 - enabling time limits [259](#)

T

- time-of-day (TOD) events [76](#)
- token authorization events [53](#)
- trace
 - archive [299](#)
 - enabling [365](#)
- Trace Browse archive [299](#)
- Trace Browse facility
 - global view with Instrumentation Server [366](#)
- TRACEDATA [251](#)
- tracing
 - reducing the amount [366](#)
- trademarks [455](#), [456](#)
- transaction coordination [205](#), [208](#)
- transactions
 - monitoring [363](#)

TSO command authorization events [54](#)
two-phase commit [205](#), [208](#)

U

unload data set, configuring access [114](#)
user authorization events [55](#)
user connections
 limiting the number [14](#)
 terminating [14](#)
user security [254](#)
using IDF for Mainframe Applications to implement DB2 RESTful services [224](#)
using IDF in DB2 application programs [227](#)

V

variables in rules [32](#)
virtual connections
 configuring [15](#)
Virtual Parallel Data
 configuring [283](#)
virtual table (VTB) events [77](#)
virtual table SAF security [255](#)
virtual tables
 peer server [216](#)
VSAM
 access to data [153](#)
 access to VSAM data [144](#)
 configure [153](#)
 configure access [144](#)
 configure CICS [156](#)
 configure security [157](#)
 specifying catalog names on metadata calls [450](#)
VSAM File Control [145](#)

W

Web Services Directory level records [352](#)
WebMethods [450](#)
work area [286](#)
Work area [286](#)
Workload Manager (WLM)
 activating service policy [274](#)
 Administration Tool [270](#)
 class rules [273](#)
 configuring [264](#)
 definitions [270](#)
 enclaves [263](#)
 Health Reporting [275](#)
 modifying a report class definition [272](#)
 modifying service class definition [271](#)
 modifying the workload [270](#)
 providing definitions [265](#), [269](#)
 using classifications [273](#)
 verifying classification [274](#)
 viewing subsystem and classification rules [272](#)

Z

z Systems Data Compression (zEDC)
 enabling [217](#)
z/OS Connect

z/OS Connect (*continued*)
 advanced configuration [196](#)
 configuring Data Virtualization Manager server [196](#)
 configuring Data Virtualization Manager server parameters [200](#)
 configuring Data Virtualization Manager Service Provider on zcEE [191](#)
 configuring z/OS Connect [191](#)
 configuring Data Virtualization Manager Service Provider on zcEE [196](#)
 DEFINE ZCPATH [196](#)
 DV configuration parameters [196](#)
 modifying the Data Virtualization Manager configuration member [194](#)
 Service Provider on zcEE [203](#)
 starting server [195](#)
z/OS Connect solution [191](#)
z/OS resource usage information [341](#)
z/OS security environment [242](#)
zFS files
 configuring access [157](#)



SC27-9303-02

