

IBM Daeja ViewONE
Version 4.1.3

*Installing, Configuring, and Using IBM
Daeja ViewONE*



IBM Daeja ViewONE
Version 4.1.3

*Installing, Configuring, and Using IBM
Daeja ViewONE*



Note

Before using this information and the product it supports, read the information in "Notices" on page 395.

This edition applies to Version 4 Release 1 Modification 3 of IBM Daeja ViewONE Standard, IBM Daeja ViewONE Professional, and IBM Daeja ViewONE Virtual (product numbers 5725-Q02, 5725-Q03, and 5725-Q04) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2014, 2015.**

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

Contents

ibm.com and related resources xi

Contacting IBM xi

Installing IBM Daeja ViewONE 1

Installing IBM Daeja ViewONE viewers and modules 1

 Installing IBM Daeja ViewONE interactively 2

 Installing IBM Daeja ViewONE silently 2

Configuring the servlet 4

Deploying the servlet 5

Configuring viewers: Applet or virtual 6

 Deploying and configuring the Java viewer applet 6

 Configuring the virtual viewer 7

 Configuring the virtual viewer servlet 7

 Configuring the virtual viewer by using HTML 8

 Configuring the virtual viewer by using

 JavaScript 9

Configuring the annotations module 11

 Enabling annotations in the viewer 12

 Configuring the annotations server object 12

 Configuring the object as a servlet. 13

 Configuring the object by using URL-encoded

 data 13

 Configuring annotation security 14

 Simple annotation security model 14

 Extended annotation security model 15

Creating an annotation template 17

 Annotation definition markers and common

 properties 18

 EMPTY marker 18

 ANNOTINFO marker 18

 EDIT property 19

 PAGE property 19

 BLANKOUTIMAGE property 20

 COLOR property 20

 CREATEDATE property 21

 CREATEDID property 21

 CUSTOMPROPERTY property 22

 HYPERLINK property. 22

 HYPERLINKSETTINGS property 25

 LABEL property 26

 LINEWIDTH property. 26

 MODIFIEDDATE property 27

 MODIFIEDID property 27

 PAGESIZE property 28

 PAGEURL property. 28

 VIEW property 28

 Annotation types and their properties 29

 ARROW annotation type 29

 CUSTOM annotation type 30

 FREEHAND annotation type 31

 HIGHLIGHT annotation type 32

 HIGHLIGHTPOLYGON annotation type 33

 NOTE annotation type. 35

 OPENPOLYGON annotation type 36

 OVAL annotation type. 37

 POLYGON annotation type 38

 RECTANGLE annotation type 40

 REDACT annotation type. 41

 REDACTPOLYGON annotation type 42

 STAMP annotation type 43

 TEXT annotation type 45

 Annotations and flow layout format documents 47

Configuring the streaming module 47

 Document streaming and prefetching. 48

 Configuring the servlet for document streaming 49

 Configuring the client viewer for document

 streaming 49

Configuring the redaction module. 50

 Configuring the servlet for redactions 50

 Configuring the client viewer for redactions . . 51

Upgrading IBM Daeja ViewONE 53

Uninstalling IBM Daeja ViewONE 55

Configuring IBM Daeja ViewONE 57

Configuring the server. 57

 Logging 58

 logFile parameter 58

 logFileSize parameter 58

 logFileCount parameter 59

 Cache control. 59

 cacheResources parameter 59

 cacheResourcesMaxItems parameter 60

 cacheResourcesMaxSize parameter. 60

 cacheRoot parameter 60

 cacheEmptyOnRestart parameter 61

 cacheCheckTime parameter 61

 cacheSizeLimit parameter 61

 cacheAgeLimit parameter. 62

 openDocumentWaitTime parameter 62

 correctPageWaitTime parameter 63

 correctDocumentWaitTime parameter 63

 anyDocumentWaitTime parameter 64

 Server sessions 64

 docURLPrototype parameter 64

 Special tokens 65

 docIdPrototype parameter 66

 sessionAuthenticationTimeout parameter . . . 66

 sessionCookieList parameter 67

 signInResource parameter 68

 serverSupportsHeadRequestMethod parameter 68

 modifiedDocumentMinimumCheckTime parameter 68

 modifiedDocumentCheckMethod parameter . . . 69

 documentExpiryTime parameter 69

 poolInitialEntryCount parameter. 70

 poolMaxEntryCount parameter 70

Configuring the viewer by using HTML parameters 70

 Checking your parameters 71

 Advanced Cache Manager parameters 72

ACMDownloadPrompt parameter	72	enableSaveButtonAfterModuleVersionChange	parameter	95
ACMDownloadPromptTrigger parameter.	72	migrateAnnotationsNotification	parameter	95
ACMPreInstall parameter.	73	migrateAnnotationsOnModuleVersionChange	parameter	95
ACMRedirect parameter	73	officeEmailInstalled parameter	96	
ACMRedirectTarget parameter	74	useAnnotationDefinedModule parameter	96	
ACMReInstall parameter	74	Notes	97	
ACMRemove parameter	75	annotationNoteRectangular parameter	97	
ACMUpdate parameter	75	annotationNoteSize parameter	98	
cachePath parameter	76	annotationNoteTextWrapping parameter	98	
useWindowsTempPath parameter	76	annotationSaveNoteId parameter	98	
Annotations options	76	showNoteIds parameter	99	
Angles and rulers	76	showNoteTooltips parameter.	99	
angleUnits parameter	77	Printing	99	
rulerScale parameter	77	annotationTemplate parameter	99	
rulerUnits parameter	77	Redaction	100	
unitDecimalPlaces parameter	78	redactAnnotationStamp<N> parameter	100	
Availability	78	redactAnnotationStampProperties<N>	parameter	101
annotate parameter.	78	reEnableTextAnnotClipping parameter	102	
annotateEdit parameter	78	transparentRedactionColor parameter	103	
annotationFile parameter	79	Resolution	103	
Color	79	tiff1dpiAllowed parameter.	103	
annotationDefaultFillColor<N>	parameter	Saving.	104	
parameter	79	annotationAutoPrompt parameter.	104	
annotationDefaultLineColor<N>	parameter	annotationAutoSave parameter	105	
parameter	80	annotationGzip parameter	105	
annotationHighlightColor parameter.	81	annotationLocalSave parameter	106	
annotationLimitedColors parameter	82	annotationPostPrefix parameter	106	
annotationLineColor parameter	82	annotationSavePost parameter	107	
annotationNoteColor parameter	82	annotationSavePostParamNames parameter	107	
annotationRedactColor parameter.	83	annotationSaveServlet parameter	108	
Cursor	83	Security	108	
annotationsStickyAcrossPages parameter	83	annotationDefaults parameter	109	
annotationsSticky parameter	84	annotationEditPasswordModify parameter	110	
Dates	84	annotationEditPasswordSecurity	parameter	110
annotationDateStyle parameter	84	annotationEditPasswordText parameter	111	
annotationDateTimeOutputLocale	parameter	annotationJavascriptExtensions	parameter	111
parameter	85	annotationSecurityModel parameter.	112	
Deletion	85	userAdmin parameter	112	
annDeleteAllDocConfirm parameter	86	userID parameter	112	
annDeleteAllPageConfirm parameter	86	Selection	113	
Encoding	86	annotationDbClick parameter	113	
annotationEncoding parameter	86	defaultAnnotationSelection parameter	113	
htmlEncoding parameter	87	Stamps	114	
Error messages	87	annotationStamp<N> parameter	114	
annotationSuppressEmptyMessages	parameter	annotationStampProperties<N> parameter	115	
parameter	87	matchStampResolution parameter.	116	
Hyperlinks	88	stampAnnotationKeepAspect parameter	117	
annotationHyperlinkWeb parameter	88	stampAnnotationResize parameter	117	
annotationTarget<N> parameter	88	Style	117	
Kodak WANG file format	89	annotationFreehandLimit parameter.	117	
imageSavePost parameter.	89	defaultLineWidth parameter	118	
imageSaveServlet parameter.	90	Text and rotation	118	
localAnnotationReadMode parameter	91	allowAnnotationRotationOnTags	parameter	118
localAnnotationWriteMode parameter.	91	parameter	118	
saveButtonToServer parameter	92			
unsupportedWangError parameter	92			
wangTextBorder parameter	93			
webAnnotationReadMode parameter.	93			
webAnnotationWriteMode parameter	94			
Module version and migration	94			
disableAnnotInfo parameter.	94			

allowResizeAnnotationRotatedDocument		thumbsTextModeAllowed parameter	143
parameter	119	thumbsText parameter	144
allowResizeRotatedAnnotation parameter	119	Font Substitution options	144
allowTextRubberband parameter	119	allowDocumentFontSubstitutions parameter	144
annotationAllowControlCodes parameter	120	enableFontSubstitutionWarning parameter	145
annotationAutoWrap parameter	120	Graphics	145
annotationClearTextList<N> parameter	120	activeButtons parameter	145
annotationClearTextOnSelection		activeColor parameter	146
parameter	121	animateTransforms parameter	146
annotationDynamicTextCreation		backColor parameter	146
parameter	121	bar<N>button<N> parameter	147
annotationFont parameter	122	border parameter	148
annotationInputKeyMapping<N>		customResource<N> parameter	149
parameter	122	extraRetrievalMessages parameter	149
annotationSubstitution<N> parameter	123	highlightColor parameter	149
annotationTabLength parameter	124	imageDecodeTextDelay parameter	150
annotationTemplateValuesFile parameter	124	imageToolBarCollapsed parameter	150
annotationTextAlignment parameter	125	retrievalTextDelay parameter	150
annotationTextLimit parameter	125	showHotKeys parameter	151
annotationTextLimitReachedMessage		splitBarAllowed parameter	151
parameter	126	statusBar parameter	151
defaultFontHeight parameter	126	theme parameter	152
Text color	126	toolbarCollapseDelay parameter	152
annotationTextColor parameter	126	toolbarExpandAllowed parameter	153
annotationTextFillColor parameter	127	toolbarExpandDelay parameter	153
Toolbars, buttons, and menus	127	toolTipsBackground parameter	153
annDeleteToolBarButton parameter	127	toolTipsForeground parameter	154
annotationAllowHideAll parameter	128	tooltipsMaxLength parameter	154
annotationEditButton parameter	128	tooltips parameter	154
annotationHideButtons parameter	129	viewToolBarCollapsed parameter	155
annotationHideContextButtons<N>		Hyperlinks	155
parameter	130	hyperlink<N> parameter	155
annotationHideContextButtonsIds<N>		hyperlinkDbClick parameter	155
parameter	132	hyperlink parameter	156
annotationHideContextButtonsIds		openOnClick parameter	156
parameter	133	openOnClickKey parameter	157
annotationHideContextButtons parameter	134	openOnClickTarget parameter	157
annotationToolBarCollapsed parameter	135	Image options	157
hideAnnotationToolBar parameter	136	adjustBars parameter	158
prefOverrideAnnotationToolBar		adjustControls1 parameter	158
parameter	136	adjustControls4 parameter	158
Tooltips	136	adjustControls8 parameter	159
alwaysShowCustomAnnotationToolTip		adjustControls24 parameter	159
parameter	136	adjustOptions parameter	159
customAnnotationToolTip parameter	137	allowHeaderTags parameter	160
Document Thumbnails and Index Lists	138	allowHeaderTagsOnPrint (rotation) parameter	160
defaultThumbsDisplayMode parameter	138	allowResolutionTags parameter	160
documentIndex<N> parameter	138	areaZoom parameter	161
documentIndexList parameter	139	areaZoomSticky parameter	161
docThumb<N> parameter	139	autoEnhanceBlack parameter	162
docThumbsEnabled parameter	140	brightness parameter	162
docThumbsImageEnabled parameter	140	contrast parameter	162
docThumbLabel<N> parameter	140	defaultTransformDoc (previously	
docThumbLabelColor<N> parameter	141	defaultRotateDoc) parameter	163
docThumbsTextEnabled parameter	141	enableSharpEnhanceModes parameter	163
hideThumbnailCloseIcon parameter	141	enhanceButton parameter	163
index<N> parameter	142	enhanceColor parameter	164
indexList parameter	142	enhanceMenu parameter	164
prefOverrideThumbsDisplayMode parameter	142	enhanceMode parameter	164
thumbsImageEnabled parameter	143	enhance parameter	165
thumbsTextEnabled parameter	143	flipOptions parameter	165

flip parameter	165	newWindowKey parameter	186
imageBackColor parameter	166	saveDocumentWithoutAnnotations parameter	186
imageButtons parameter	166	tiffSaveColor parameter	186
imageForeColor parameter	167	tiffSaveVersion6Jpeg parameter	187
imageKeys parameter	167	Microsoft Office module options	187
imageMenus parameter	167	allowExternalEmailImages parameter	187
infoOptions parameter	168	allowSaveEmailAttachments parameter	188
infoOptionsShowSizeCm parameter	168	blockEmailModule parameter	188
infoOptionsShowSize parameter	168	emailHeaderInfo parameter	188
invertButtons parameter	169	excelGridlines parameter	189
invertKeys parameter	169	excelViewMode parameter	189
invertMenus parameter	169	listEmailAttachmentsInHeader parameter	189
invert parameter	170	showExternalEmailImages parameter	190
luminance parameter	170	showPlainTextForRTFEmail parameter	190
magOptions parameter	170	Documents and images	190
pageRotationN parameter	170	baseFilename parameter	191
persistViewProcessingProperties parameter	171	description parameter	191
photometric parameter	171	doc<N> parameter	191
rotation parameter	172	fileBase parameter	193
scale parameter	172	fileList parameter	193
thirdPartyFileCache parameter	173	fileDir parameter	194
visibleImageButtons parameter	173	filename parameter	194
visibleImageKeys parameter	173	httpRedirectAllowSSL parameter	195
visibleImageMenus parameter	174	httpRedirectCountLimit parameter	195
zoom100Options parameter	174	page<N> parameter	196
zoom100 parameter	174	pageCount parameter	197
JavaScript options	175	pageNumber parameter	197
annotationJavascriptExtensions parameter	175	prefetchPages parameter	197
javascriptExtensions parameter	175	prefetchThumbs parameter	198
eventHandler parameter	176	processHttpRedirects parameter	199
eventHandlerJson parameter	176	redirectNotFound parameter	199
eventInterest parameter	176	redirectTarget parameter	199
serverEventHandler parameter	177	resourceToUrlEncoding parameter	200
Labels	178	routeDocs parameter	200
displayPageLabels parameter	178	routeHTML parameter	201
docRef<N> parameter	178	routeHTMLTarget parameter	201
pageLabel<N> parameter	178	routeXML parameter	202
pageLabelColor<N> parameter	179	routeXMLTarget parameter	202
pageLabelNullifier parameter	179	routeTarget parameter	203
printLabelBorder parameter	179	suppressMenusDuringOpen parameter	204
printLabels parameter	180	thumb<N> parameter	204
thumbLabel<N> parameter	180	timeOut parameter	205
thumbLabelColor<N> parameter	180	PDF options	205
Local file options	181	alwaysShowPDFAnnotations parameter	205
defaultSaveFilename parameter	181	autoLimitPDFResolutionValue parameter	205
fileButtonClose parameter	181	defaultAutoLimitPDFResolution parameter	206
fileButtonOpen parameter	181	defaultPDFPixelDepth parameter	206
fileButtonSave parameter	182	defaultPDFResolution parameter	207
fileButtons parameter	182	findResultColor parameter	207
fileKeyClose parameter	182	findResultHighlightColor parameter	207
fileKeyOpen parameter	183	findResultHistoryListSize parameter	207
fileKeySave parameter	183	forceJavaPdf parameter	208
fileKeys parameter	183	Print options	208
fileMenuClose parameter	183	allowPrintHeader parameter	208
fileMenuOpen parameter	184	defaultPrintDoc parameter	208
fileMenuSave parameter	184	defaultPrintingNotesText parameter	209
fileMenus parameter	184	defaultPrintNoteAnnotation parameter	209
fixedSavePath parameter	184	disablePrintCopies parameter	210
java2FileDialogs parameter	185	forcePrintsPerPage parameter	210
makeSaveNameSafe parameter	185	imageTooLargePrintHeader parameter	210
newWindowButtons parameter	186	java2PrintSetupDialog parameter	211

multiPrintNum parameter	211	displayColorProgressively parameter	238
multiPrintGap parameter	212	docCompare parameter	238
multiPrintSingleHeader parameter	212	displayDocButtons parameter	238
printAcceleratorForgetSettings parameter	213	displayMonoProgressively parameter	239
printAccelerator parameter	213	draggingEnabled parameter.	239
printAcceleratorReverseMono parameter	213	emptyClipboardOnClose parameter	239
printAnnotations parameter	214	emptyOnClose parameter	240
printAutoRotate parameter.	214	enableDocButtons parameter	240
printButtons parameter	215	externalMagnifier parameter	241
printComplete parameter	215	flickScrollEnabled parameter	241
printDialog parameter	215	focusBorder parameter	241
printDocumentAllowed parameter.	216	helpTarget parameter	241
printer parameter.	216	helpURL parameter.	242
printHeader parameter	217	htmlListShortcutEnabled parameter.	242
printingColorHeader parameter	217	httpReadTimeout parameter.	243
printingFontBold parameter	218	initialFocus parameter.	243
printingFontName parameter	218	keepAlive parameter	243
printingFontSelectionEnabled parameter	219	keepAliveTime parameter	244
printingFontSize parameter	219	keepScroll parameter	244
printingNotesFontName parameter	220	magnifierFollowMouse parameter.	245
printingNotesFontSize parameter	220	magnifierPersistent parameter	245
printingNotesStandardFont parameter	220	menuNewWindow parameter	246
printingNotesTextpageTitle parameter	221	mouseRightButtonEnabled parameter.	246
printingNotesTextTitle parameter	221	newWindow parameter	246
printingNotesTitleFontSize parameter	222	nwViewMode parameter	246
printJobName parameter.	222	obfuscate parameter	247
printKeys parameter	222	pageButtons parameter	248
printMenus parameter	223	pageKeys parameter	248
printMonoAllowed parameter	223	pageMenus parameter	248
printMono parameter	223	preferenceMenus parameter.	249
printOriginalSizeAllowed parameter	224	saveExtension parameter	249
printOriginalSizeOverride parameter	224	scrollbars parameter	249
printRangeAllowed parameter	224	selectKeys parameter	250
printSelectedAllowed parameter.	225	selectMenus parameter	250
printStickyMenus parameter	225	serverEventInterest parameter	250
printVisibleAllowed parameter	225	serverFileMonitor parameter	251
printWithUserSettings parameter	226	synchronizeThumbs parameter	251
reusePrinter parameter.	226	systemInfoShortcutEnabled parameter	251
separator<N> parameter	227	thumbDbClick parameter	252
separatorList parameter	229	thumbDbClickSelect parameter (Deprecated)	252
Text file options	230	thumbsHeight parameter.	253
allowTextFilterDisplayAnyFile parameter	230	thumbsWidth parameter	253
ignoreTextFilterMimeTypes parameter	231	useBrowserAlertOnShutdown parameter	253
textEncoding parameter.	231	useErrorFiles parameter	254
textFilterLocalFileExtensions parameter	231	viewButtonsDetail parameter	254
textFilterMimeTypes parameter	232	viewButtons parameter	254
textFilterPageBreaks parameter.	232	viewKeys parameter	254
textFilterPageHeight parameter.	233	viewMenus parameter	255
textFilterPageWidth parameter	233	viewMode parameter	255
textFilterWrapping parameter	234	zoom, xScroll, and yScroll parameters.	256
Viewing options	234	ViewONE Document Streaming Server Module	
abortSaveOnShutdown parameter	234	options	257
allButtons parameter	234	cookieAllowedDomainList parameter.	257
allKeys parameter.	235	documentCookies parameter.	257
allMenus parameter	235	streamerEnabled parameter.	257
autoScroll parameter	235	streamerURL parameter	258
cacheMenus parameter	236	ViewONE Permanent Redaction Server Module	
clipOptions parameter	236	options	258
defaultLinkViewer parameter	236	annotationBurnURL parameter	258
defaultZoomDoc parameter	237	annotationDefaults parameter	259
disableShutdownDialogs parameter	237	burnIncrementalAnnotations parameter	260

burnUseGZip parameter	260	setAnnotationsSemiTransparent method	290
redactionHttpReadTimeout parameter	260	setAnnotationStampText method	291
redactionHttpReadTimeout parameter	260	setDefaultSelectAnnotation method	291
showAnnotationBurnPrompt parameter	261	setDelimiter method	292
ViewONE Universal Viewing Module options	261	setRedactionIsSemiTransparent method	292
obfuscateUV parameter	261	setRulerScale method	293
uvTextDirectionMethod parameter	262	setRulerUnits method	293
IBM Daeja ViewONE color scheme text.	262	setStickyAnnotations method	293
IBM Daeja ViewONE Default Stamps	263	showAnnotationToolBar method	294
Supported fonts	263	startAnnotation method	294
WANG Compliant Annotation Types	263	startAnnotationWithProperties method	295
Configuring the viewer by using JavaScript API		startModifyAnnotations method	297
methods	264	Document hyperlinks.	298
Adding selected items to the clipboard	265	clearHyperlink method	298
clearSelections method	265	setHyperlink method	298
copyPageToClipboard method	265	Documents and images	298
getSelection method	266	closeDocument method	299
selectPage method	266	firstDoc method	299
startCopyAreaToClipboard method	266	getDocIndex method	299
Annotations	267	getDocReference method	300
Specify an Annotations File.	267	getNumDocs method	301
isShowAnnotations method.	267	getNumPages method	301
setAnnotationFile method.	267	getPage method	301
setShowAnnotations method	268	initializeDocumentArray,	
Open and save annotations.	268	openDocumentArray, and setDocumentArray	
reloadAnnotations method.	268	method group	302
saveAnnotations method	269	initializePageAndThumbsArray,	
setAnnotationFile method.	269	openPageArray, setPageArray, and	
setAnnotationPostPrefix method	270	setThumbsArray method group	302
setAnnotationSave method.	271	initializePageArray, openPageArray, and	
setAnnotationSavePost method	271	setPageArray method group	303
setAnnotationSaveServlet method	272	lastDoc method	304
Edit and find annotations	273	nextDoc method	304
addAnnotation method	273	nextPage method	305
addAnnotationStamp method	274	openDirectory method	305
addAnnotationStampWithProperties		openDoc method	306
method	274	openFile method	306
burnAnnotations method	276	openList(listFile, page) method	307
clearAnnotationStamps method	276	prevDoc method	307
deleteAllAnnotations method	276	previousPage method	308
deleteAnnotation method	277	reloadList method	308
endAnnotation method	278	setBackgroundImageEnabled method.	308
endModifyAnnotations method	278	setDocumentCookies method	309
getActiveAnnotation method	278	setDocumentIndexListFile method	309
getAnnotation method	279	getDescription and setDescription method	
getAnnotationLabels method	279	group	310
getAnnotationOnPage method	280	setPage method	310
getDelimiter method	281	showExternalEmailImages method	310
getNumAnnotations method.	281	isMultipageTif method	311
insertAnnotationStamp method	282	saveAsFormat method.	311
isAnnotationsSemiTransparent method	283	save method.	311
isAnnotationToolBar method	284	savePage method	312
isAnnotationsUpdated method	284	saveSelected method	312
modifyAnnotation method	284	setDefaultSaveFilename(filename) method	313
parseProperty method	285	setImageSaveURL method	313
removeAnnotationStamp method	285	Document text search	314
setAnnotateEdit method	286	cancelFind method	314
setAnnotationHideButtons method	286	clearFindResults method	314
setAnnotationHideContextButtons method	287	find method	314
setAnnotationHideContextButtonsIds		nextFindResult method	315
method	289	prevFindResult method	315

Event handlers and event handling	315	isPageKeys method	345
Events IDs and descriptions	316	isPageMenus method	345
Obtain additional information from an event	323	isPreferenceMenus method	346
Image options	323	isPrintKeys method	346
applyRotationArray,		isPrintMenus method	346
initializeRotationArray, and		isSelectKeys method	346
setRotationArray method group	323	isSelectMenus method	347
endUserDragMode method	324	isViewKeys method	347
getBrightness method	324	isViewMenus method	347
getContrast method	325	setAllKeys method	347
getEnhanceMode method	325	setAllMenus method	348
getFlip method	325	setCacheMenus(true/false) method	348
getImageHeight method	326	setFileKeys method	349
getImageWidth method	326	setFileMenus method	349
getLuminance method	326	setImageKeys method	350
getRotation method	326	setImageMenus method	350
getScale method	327	setPageKeys method	351
getStates and setStates method group	327	setPageMenus method	351
getXResolution method	328	setPreferenceMenus method	351
getYResolution method	328	setPrintKeys method	352
getXScroll method	328	setPrintMenus method	352
getYScroll method	328	setSelectKeys method	353
getZoom method	329	setSelectMenus method	353
grabImage method	329	setViewKeys method	354
invert method	332	setViewMenus method	354
isDraggingEnabled method	332	Print options	354
isEnhance method	332	isPrintDialog method	354
isInverted method	333	printDocument method	355
isUserDragMode method	333	printPage method	355
resetBrightness method	333	printRange method	355
resetContrast method	334	printSelected method	356
resetLuminance method	334	printTransformed method	356
rotateClockwise method	334	printVisible method	357
rotateCounterclockwise method	335	setPrintAutoRotate method	357
rotate180 method	335	setPrintCopies method	358
setBrightness method	335	setPrintDialog method	358
setContrast method	336	setPrinter method	359
setDraggingEnabled method	336	setPrintHeader method	359
setEnhanceMode method	336	setPrintJobName method	360
setEnhance method	337	Streamer methods	361
setFlip method	337	setStreamerEnabled method	361
setInverted method	338	setStreamerURL method	361
setLuminance method	338	Timeout and the user idle control	361
setRotation method	339	isTimedOut method	361
setScale method	339	getTimeLeft method	362
setXYScroll method	340	getTimeout method	362
setZoomAndXYScroll method	340	setServerEventHandler method	363
setZoom method	341	setEventHandlerResponse method	363
startUserDragMode method	341	setTimeout method	363
zoomArea method	342	stopTimeout method	364
zoomIn method	343	wakeUp() method	364
zoomOut method	343	Toolbars and buttons	365
zoom100 method	343	isAdjustToolVisible method	365
Menus and keyboard shortcuts	343	isAllButtons method	365
isAllKeys method	343	isDocButtons method	366
isAllMenus() method	344	isFileButtons method	366
isCacheMenus method	344	setImageButtons method	366
isFileKeys method	344	invertButtons method	366
isFileMenus method	344	isNewWindowButtons method	367
isImageKeys method	345	isPageButtons method	367
isImageMenus method	345	isPrintButtons method	367

isScrollbars method	367
isStatusBar method	368
isViewButtons method	368
isZoom100Button method	368
setAdjustToolVisible method	368
setAllButtons method	369
setAppletEnabled method	369
setDocButtons method	370
setFileButtons method	370
setImageButtons method	370
setImageToolBarCollapsed method	371
setInvertButtons method	371
setNewWindowButtons method	372
setPageButtons method	372
setPrintButtons method	372
setScrollbars method	373
setStatusbar method	373
setViewButtons method	374
setViewToolBarCollapsed method	374
setZoom100Button method	374
toggleAdjustTool method	375
Viewing options	375
getMagFactor method	375
getView method	375
getViewerWindow method	376
getViewerWindowInstanceId method	376
initializeLabels, setLabel, useLabels, and clearLabels method group	377
isAreaZoom method	378
isMagnifier method	378
isNewWindowVisible method	378
setAreaZoom method	378
setImageBackColor method	379
setImageForeColor method	379
setMagBounds method	380
setMagFactor method	380
setMagnifierInternal method	380

setMagnifier method	381
setNewWindowVisible method	381
setView method	381
showImageBackColorDialog method	382
showImageForeColorDialog method	382
toggleAreaZoom method	382
toggleMagnifier method	383

Annotating and redacting documents 385

Editing annotations	385
Tips for using the viewer	386

Troubleshooting and support 389

Getting fixes from Fix Central	391
Contacting IBM Software Support	391

Notices 395

Glossary 399

A	399
D	399
E	399
F	399
I	399
K	400
L	400
M	400
N	400
P	400
R	400
S	400
T	400
V	400
W	400
Z	401

ibm.com and related resources

Product support and documentation are available from ibm.com[®].

Support and assistance

From ibm.com, click **Support & downloads** and select the type of support that you need. From the Support Portal, you can search for product information, download fixes, open service requests, and access other tools and resources.

IBM[®] Knowledge Center

See your online product information in IBM Knowledge Center at <http://www.ibm.com/support/knowledgecenter/SSTPHR/welcome>.

PDF publications

See the following PDF publications for your product at https://www.ibm.com/support/entry/portal/documentation_expanded_list/enterprise_content_management/case_manager?productContext=-1017567207.

Contacting IBM

For general inquiries, call 800-IBM-4YOU (800-426-4968). To contact IBM customer service in the United States or Canada, call 1-800-IBM-SERV (1-800-426-7378).

For more information about how to contact IBM, including TTY service, see the Contact IBM website at <http://www.ibm.com/contact/us/>.

Installing IBM Daeja ViewONE

Before you install IBM Daeja™ ViewONE, you must install all of the prerequisite software.

Before you begin

IBM Daeja ViewONE requires a web application server. Ensure that you are using a supported version of a web application server, such as WebSphere® Application Server.

Related information:

System requirements

Installing IBM Daeja ViewONE viewers and modules

The installation program automatically installs the built-in modules for your edition. In addition, you use the installation program to install add-on modules for which you have entitlements. You can install the IBM Daeja ViewONE software on the server by running the installation program interactively or by running a silent installation.

About this task

The IBM Daeja ViewONE installation program is a multiple-product installation program that you use for any product edition:

- IBM Daeja ViewONE Professional
- IBM Daeja ViewONE Virtual
- IBM Daeja ViewONE Standard

Procedure

To install IBM Daeja ViewONE:

1. Log in to Fix Central.
2. From the **Product Group** list, select **Enterprise Content Manager**.
3. From the **Select from Enterprise Content Manager** list, select the IBM Daeja ViewONE edition that you want to install. You must have a license for this edition.
4. From the **Installed Version** list, select the version that is installed on your system. Select **All** if your version is not listed.
5. From the **Platform** list, select the operating system on which you are installing the software and then click **Continue**.
6. Select **Browse for fixes** and click **Continue**.
7. On the Select fixes page, select **Version 4.1.3** and click **Continue** to download the software.
8. Extract the downloaded file into a temporary directory on your system.
9. Run the installation program either interactively or silently.

Installing IBM Daeja ViewONE interactively

You can install IBM Daeja ViewONE on the server by running the installation program. During installation, you are prompted to select the editions and modules for which you are entitled.

About this task

The interactive installation program uses a graphical user interface (GUI). If your operating system does not support GUIs, you must install IBM Daeja ViewONE silently.

Procedure

To install IBM Daeja ViewONE interactively:

1. Navigate to temporary directory in which you extracted the IBM Daeja ViewONE software package.
2. Run the installation program:

Operating system	Command
AIX®	./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin
Linux	./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin
Solaris	./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin
Windows	DAEJA_VIEWONE_4.1.3_WIN_ML.exe

3. Complete the IBM Daeja ViewONE installation program.
4. When the installation completes, check for errors in the IBM Daeja ViewONE installation log.

The installation log file, `DaejaViewONE_install_4.1.3.0.txt`, is in the root installation directory. For example, the default location for the log file on a Windows server is `C:\Program Files\IBM\Daeja ViewONE`.

Installing IBM Daeja ViewONE silently

You can install IBM Daeja ViewONE on the server by running a silent installation. When you run a silent installation, the installation program does not display any installation windows or messages. Instead, any messages and errors are written to the installation log file.

About this task

A silent installation uses a response file that contains the installation parameters that you specify. The software package includes two sample response files. You must modify one of these files to specify your entitlements. The files contain comments that describe the modifications that you can make.

The file that you modify depends on your contract type and operating system:

- If IBM Daeja ViewONE was purchased from IBM or your company already migrated to an IBM license, then your contract type is *Passport Advantage*®. If you are installing on a Microsoft Windows system, use the `win_IBMDaejaViewONE_PA.properties` file. If you are installing on an AIX, Linux, or Solaris system, use the `unix_IBMDaejaViewONE_PA.properties` file.
- If you have a legacy version of IBM Daeja ViewONE that was purchased directly from Daeja Systems Ltd., then your contract type is *Flexible Contract Type*. If you

are installing on a Microsoft Windows system, use the win_IBMDaejaViewONE_FCT.properties file. If you are installing on an AIX, Linux, or Solaris system, use the unix_IBMDaejaViewONE_FCT.properties file.

Procedure

To install IBM Daeja ViewONE silently:

1. Navigate to temporary directory in which you extracted the IBM Daeja ViewONE software package.
2. If your contract type is Passport Advantage, edit the win_IBMDaejaViewONE_PA.properties response file or unix_IBMDaejaViewONE_PA.properties response file:
 - a. Under Accept the License Agreement, set the value to TRUE:
LICENSE_ACCEPTED=TRUE
 - b. Under Select Edition, set the value to 1 for the edition that you want to install:

Value	Edition
PRO	IBM Daeja ViewONE Professional
VIR	IBM Daeja ViewONE Virtual
PRO-VIR	IBM Daeja ViewONE Professional and IBM Daeja ViewONE Virtual
STD	IBM Daeja ViewONE Standard

For example, to install IBM Daeja ViewONE Professional, set the values as follows:

```
PRO=1
VIR=0
PRO-VIR=0
STD=0
```

- c. If you are installing to a development or test environment, set PRODUCTION to 0 under Select the Installation environment:
PRODUCTION=0
 - d. If you purchased ViewONE Office Module that is bundled for IBM Daeja ViewONE Professional and IBM Daeja ViewONE Virtual, set MS_OFFICE_BUNDLE to 1 under Select Bundles:
MS_OFFICE_BUNDLE=1
 - e. Under Select Entitlements, specify the modules that you want to install. For example, to install IBM Daeja ViewONE Virtual with the ViewONE Office Module, set the value as follows:
CHOSEN_INSTALL_FEATURE_LIST=VirtualModuleForMicrosoftOffice
3. If your contract type is Flexible Contract Type, edit the DAEJA_VIEWONE_FCT_silent_install.txt response file:
 - a. Under Accept the License Agreement, set the value to TRUE:
LICENSE_ACCEPTED=TRUE
 - b. Under Select Edition, set the value to 1 for the edition that you want to install:

Value	Edition
PRO	IBM Daeja ViewONE Professional
STD	IBM Daeja ViewONE Standard

For example, to install IBM Daeja ViewONE Professional, set the values as follows:

```
PRO=1
STD=0
```

- c. Under Select Entitlements, specify the modules that you want to install. You must always include the base product in this value. For example, to install IBM Daeja ViewONE Professional with the ViewONE Permanent Redaction Server Module, set the value as follows:

```
CHOSEN_INSTALL_FEATURE_LIST=ProfessionalPermanentRedactionServerModule
```

4. Save your changes to the response file.
5. At a command line, change to the directory that contains the installation program and run the following command:

Operating system	Command
AIX	<code>./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin -f response_file</code>
Linux	<code>./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin -f response_file</code>
Solaris	<code>./DAEJA_VIEWONE_4.1.3_UNIX_ML.bin -f response_file</code>
Windows	<code>DAEJA_VIEWONE_4.1.3_WIN_ML.exe -f response_file</code>

response_file is the fully qualified path of the edited response file.

6. When the installation completes, check for errors in the IBM Daeja ViewONE installation log.

The installation log file, `DaejaViewONE_install_4.1.3.0.txt`, is in the root installation directory. For example, the default location for the log file on a Windows server is `C:\Program Files\IBM\Daeja ViewONE`.

Configuring the servlet

You must edit the `web.xml` file and the `servlet-name.properties` files to configure the viewer servlet.

About this task

The viewer servlet makes up the core server component for the IBM Daeja ViewONE software suite. For IBM Daeja ViewONE Professional, the servlet is used for the server-side ViewONE Document Streaming Server Module and ViewONE Permanent Redaction Server Module. For IBM Daeja ViewONE Virtual, the servlet provides the entry point that is required for the viewer to work.

To configure the servlet, you need the following files:

web.xml

This XML file defines the location of the core servlet class files (`com.ibm.dv.server.Platform`).

If you are adding IBM Daeja ViewONE to an existing web application, you might already have a `web.xml` file. If you are creating a new web application, you might need to create the `web.xml` file.

servlet_name.properties

This properties file has the same name as the servlet that you are using, such as `viewone.properties`. You use this file to configure extended debugging output to a log file.

Procedure

To edit the configuration files:

1. Copy all the files and directories from the *daeja_install_dir/server/WEB-INF* directory into the */WEB-INF* directory of your web application. These files include the JAR files in the */lib* directory and the *lic-server.v1* or *lic-server-virtual.v1* configuration file.
2. Open the *WEB-INF/web.xml* file in a text editor and make sure that it contains the following lines.

```
<web-app>
...
  <servlet>
    <servlet-name>viewone</servlet-name>
    <servlet-class>com.ibm.dv.server.Platform</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>viewone</servlet-name>
    <url-pattern>/v1files/*</url-pattern>
  </servlet-mapping>
...
</web-app>
```

You can replace *viewone* with the name of your servlet. This servlet is called when a request is made to */v1files/**.

Important: You must include the */** component in the URL-pattern mapping to enable the servlet to load the viewer files and resources that are required for the servlet to function correctly.

3. Open the *WEB-INF/servlet_name.properties* file in a text editor and make sure that it contains the following lines to configure logging.

```
logfile=WEB-INF/viewone.log
```

The properties file name must match the servlet name that you defined in *WEB-INF/web.xml*. For example, name the file *viewone.properties*.

If you do not already have a *WEB-INF/servlet_name.properties* file, you must create the file.

You can replace *viewone* with the name of your servlet.

To send log information to the standard web application server log file, use the following line to configure logging:

```
logfile=<stdout>
```

What to do next

You can edit these files later to customize the viewer to meet your requirements.

Related information:

 [Configuring the server](#)

You can set optional properties to configure and customize the server elements of IBM Daeja ViewONE.

Deploying the servlet

To deploy the core servlet, you must use a Java™ web application server of your choosing. The procedure for setting up the servlet depends on the web application server that you are deploying to.

Procedure

To deploy the application:

1. Follow the directions from your web application server documentation to deploy the web application to the web application server.
2. Verify that you can access the application by entering the URL to the server into the browser. For example, enter the following URL: `http:myserver:8080/viewone_applet.html`

Related information:

[☞ WebSphere Application Server Network Deployment, Version 7.0 documentation](#)

See this document for information about deploying the servlet to WebSphere Application Server Network Deployment, Version 7.0

[☞ WebSphere Application Server Network Deployment, Version 8.0 documentation](#)

See this document for information about deploying the servlet to WebSphere Application Server Network Deployment, Version 8.0

[☞ WebSphere Application Server Network Deployment, Version 8.5.5 documentation](#)

See this document for information about deploying the servlet to WebSphere Application Server Network Deployment, Version 8.5.5

Configuring viewers: Applet or virtual

IBM Daeja ViewONE provides a Java applet viewer and a virtual viewer.

About this task

The Java applet viewer is deployed for IBM Daeja ViewONE Professional and IBM Daeja ViewONE Standard. This applet renders the images on the client instead of on the server. Users must be able to download and run the applet to view files.

The IBM Daeja ViewONE Virtual viewer is designed for systems that do not support the Java plug-in. All the work is carried out on the server so that the viewer, associated images, and other relevant data are streamed to the client device.

The IBM Daeja ViewONE Virtual viewer is compatible with most mobile devices, such as tablets and phones.

Deploying and configuring the Java viewer applet

The applet viewer is deployed as a Java applet. This applet viewer requires the use of a web server of your choosing to deliver the viewer files to the client system by using a standard HTTP request.

Procedure

1. Copy all the `/client` files, including the `lic.v1` license file, into a directory on your web application server. For example, copy the files to the `/v1files` directory.

2. If your web server requires you to register MIME types, add the .v1 and .ant MIME types in order for the browser to download files to the client. For details about how to configure the MIME types, refer to your web server documentation.
3. If you use authentication credentials that are passed as cookies to the server repository, set the **HttpOnly** parameter to false.
4. Edit the HTML page that loads the viewer applet to add the <APPLET> section as follows.

```
<APPLET CODEBASE = "."  
ARCHIVE = "viewone.jar"  
CODE = "com.ibm.dv.client.Viewer.class"  
NAME = "ViewONE"  
WIDTH = "100%"  
HEIGHT = "97%"  
HSPACE = "0"  
VSPACE = "0"  
ALIGN = "middle">  
</APPLET>
```

The codebase value must point to the location of the /v1files directory relative to the location of the HTML page. In this example, a dot (.) indicates the current folder.

If the viewer fails to run and you see messages that refer to `java.lang.ClassNotFoundException`, use the full URL for the location of the codebase files instead of a relative path. Refer to your web server logs to determine the issue with delivering the viewer files to the client system.

What to do next

After you deploy the viewer, you can use the HTML parameters and JavaScript methods to customize the viewer.

Related information:

 [Configuring the viewer](#)

You can configure the viewer by using HTML parameters and JavaScript API methods.

Configuring the virtual viewer

IBM Daeja ViewONE Virtual viewer is deployed as a Java servlet.

About this task

When you configure and deploy the servlet, the core of IBM Daeja ViewONE Virtual is also included. However, you must modify the web page that loads the viewer. You can use HTML5 or later or JavaScript to configure the viewer.

Important: Ensure that your code that runs the viewer is running from the same location where the server is installed or else the viewer does not run. Because of the restrictions of cross domain origin policies, it is not possible to run the HTML code from one host while also accessing the JavaScript files from another host.

Configuring the virtual viewer servlet

You can edit the `servlet-name.properties` file to configure the virtual viewer servlet. This file is shared with the core IBM Daeja ViewONE servlet.

About this task

For example, you can add parameters to enable debug output to a specified log file, to control the servlet caching area, or to determine how documents are retrieved.

Important: Because the *servlet-name.properties* file is shared with other modules, you must precede the property entry with *tile* to indicate the virtual viewer. For example, to specify debugging level, use *tile.tracenet=true*.

Procedure

To edit the configuration file:

1. Open the `WEB-INF/servlet_name.properties` file in a text editor and make sure that it contains the following lines to configure logging.

```
tile logfile=WEB-INF/viewone.log
tile traceNet=true
```

The properties file name must match the servlet name that you defined in `WEB-INF/web.xml`. For example, use the `viewone.properties` file.

You can replace *viewone* with the name of your servlet.

2. Optional: Add more parameters as need to customize the servlet.

Related information:

 [Configuring the server](#)

You can set optional properties to configure and customize the server elements of IBM Daeja ViewONE.

Configuring the virtual viewer by using HTML

You configure the IBM Daeja ViewONE Virtual viewer by using HTML.

Procedure

1. Edit the HTML page that displays the virtual viewer to add the following lines.

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript"
    src="/viewone/vlfiles/viewone.js">
  </script>
</head>
<body>
  <object
    class="com.ibm.dv.client.Viewer"
    id="viewone"
    width="600"
    height="600">
    <!-- Optional - place additional HTML parameters here -->
  </object>
</body>
</html>
```

Important: The line that begins `src=` must point to the IBM Daeja ViewONE Virtual viewer servlet that provides the viewer application code. By default, the path for the servlet is `"/viewone/vlfiles/viewone.js`. Be sure to use the correct path for your environment.

2. Verify that you can access the viewer by entering the URL to the HTML page in a browser. For example, enter the following URL: `http://myserver:8080/viewone/viewone_virtual.html`

Tip: If the viewer fails to run, specify the full URL to the location of the servlet rather than a relative path for the src= entry.

What to do next

You can edit the HTML file later to customize the viewer to meet your requirements.

Configuring the virtual viewer by using JavaScript

You can use JavaScript to start the IBM Daeja ViewONE Virtual viewer. In this approach, the initial page that the user sees displays a list of documents to be viewed, but not a viewer. By using this approach, you can support different viewers for different documents or you can display multiple viewers on the same page to view multiple documents simultaneously. You can also provide quick viewing of individual documents in a tabbed interface.

About this task

With the HTML approach, the viewer detects the presence of the object tag within the HTML page and starts the viewer automatically. However, if you are injecting the viewer with JavaScript, the viewer does not automatically detect the object tag. After the injection to start the viewer, you must call the following function with the ID viewone:

```
com.ibm.dv.client.Viewer.init('viewone');
```

If you do not call this function, the viewer does not run, and you have an empty object container on the screen and no viewer.

The following example illustrates the correct way to call this function:

```
function appendViewer1()
{
    var viewerCode = '<object class="com.ibm.dv.client.Viewer"
        + id="viewone1" width="900" height="600">' +
        + '<param name="filename"
        + value=http://myserver/sample.tif?&path=docs/>'
        + '</object>';

    document.getElementById('container').innerHTML = viewerCode;

    com.ibm.dv.client.Viewer.init('viewone');
}
```

You can remove a viewer from a page by calling the following function:

```
com.ibm.dv.client.Viewer.destroy('viewone');
```

The following example demonstrates how to call the viewer to display four different documents: JPG, TIFF, JPG, and DOCX.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type"
content="text/html; charset=UTF-8">

<title>ViewONE Virtual</title>

<style type="text/css">
html, body{
    height: 100%;
    position: relative;
```

```

}
.viewers{
    height: 100%;
}
.documentList{
    height: 100%;
    float: left;
    border-right: 2px solid gray;
    padding-right: 2em;
}
.viewerContainer {
    position: relative;
    clear: none;
    float: left;
    width: 400px;
    height: 500px;
}
</style>

<script type="text/javascript" src="v1files/viewone.js"></script>

<script type="text/javascript">

function addViewer(docURL, docId)
{
    //alert ("add viewer "+docId);
    var viewerId = "viewone"+docId;
    var viewerObj = document.createElement("object");
    viewerObj.setAttribute("id", viewerId);
    viewerObj.setAttribute("class", "com.ibm.dv.client.viewer");
    viewerObj.setAttribute("width", "400px");
    viewerObj.setAttribute("height", "500px");
    addParameter(viewerObj, "filename", docURL);
    var viewerDiv = document.createElement("div");
    viewerDiv.appendChild(viewerObj);
    viewerDiv.setAttribute("id", "div_"+docId);
    viewerDiv.setAttribute("class", "viewerContainer");
    document.getElementById("viewerContainer").appendChild(viewerDiv);
    window.com.ibm.dv.client.Viewer.init(viewerId);
    setTimeout(setViewerStyle(viewerId), 50);
    return false;
}

function setViewerStyle(id){
    document.getElementById(id).style.cssFloat = "left";
}

function addParameter(target, pName, pVal)
{
    var paramObj = document.createElement("param");
    paramObj.setAttribute("name", pName);
    paramObj.setAttribute("value", pVal);
    target.appendChild(paramObj);
}

function removeViewer(docId)
{
    var viewerId = "viewone"+docId;
    window.com.ibm.dv.client.Viewer.destroy(viewerId);
    var viewerDiv = document.getElementById("div_"+docId);
    viewerDiv.parentNode.removeChild(viewerDiv);
    window[viewerId] = null;
    delete window[viewerId];
    return false;
}

```



```

</script>
</head>

<body>
  <div class="documentList">
    <h3>Documents</h3>
    <p>
      Photo 1 <a href="#"
        onclick="addViewer('http://server/app/docs/photo1.jpg',
          + 'doc1');">view</a>&nbsp;&nbsp;&nbsp;
      <a href="#" onclick="removeViewer('doc1')">close</a>
    </p>
    <p>
      Image 1 <a href="#"
        onclick="addViewer('http://server/app/docs/image.tif',
          + 'doc2');">view</a>&nbsp;&nbsp;&nbsp;
      <a href="#" onclick="removeViewer('doc2')">close</a>
    </p>
    <p>
      Photo 2 <a href="#"
        onclick="addViewer('http://server/app/docs/photo2.jpg',
          + 'doc3');">view</a>&nbsp;&nbsp;&nbsp;
      <a href="#" onclick="removeViewer('doc3')">close</a>
    </p>
    <p>
      Document 1 <a href="#"
        onclick="addViewer('http://server/app/docs/document.docx',
          + 'doc4');">view</a>&nbsp;&nbsp;&nbsp;
      <a href="#" onclick="removeViewer('doc4')">close</a>
    </p>
  </div>
  <div id="viewerContainer" class="viewers"></div>
</body>
</html>

```

Procedure

1. Edit the HTML page that displays the virtual viewer as shown in the examples.
2. Verify that you can access the viewer by entering the URL to the HTML page in a browser. For example, enter the following URL: `http://myserver:8080/viewone/viewone_virtual.html`

What to do next

You can edit the HTML file later to customize the viewer to meet your requirements.

Configuring the annotations module

The ViewONE Annotations Module provides text and image annotations that users can add to documents.

Before you begin

You must configure the viewer and verify that you can view images in this viewer. You can use either the IBM Daeja ViewONE applet viewer or the IBM Daeja ViewONE Virtual viewer.

About this task

If you are not using an existing Enterprise Content Manager system in which the annotations interface is already written and integrated, you must create the server

component. This component provides the interface between the viewer and the repository where image and annotation data is stored.

Related tasks:

“Installing IBM Daeja ViewONE viewers and modules” on page 1

The installation program automatically installs the built-in modules for your edition. In addition, you use the installation program to install add-on modules for which you have entitlements. You can install the IBM Daeja ViewONE software on the server by running the installation program interactively or by running a silent installation.

Enabling annotations in the viewer

After you deploy a valid `lic.v1` configuration file that includes the ViewONE Annotations Module, you can enable annotations in the viewer.

Procedure

To enable annotations:

1. Open the viewer by entering the URL to the HTML page in a browser. For example, enter the following URL: `http://myserver:8080/viewone/viewone_virtual.html`
2. Click the IBM Daeja ViewONE logo in the viewer and verify that the ViewONE Annotations Module is included in the list of extensions.
3. Edit the viewer HTML page to add the following parameters.

Option	Description
Enable annotations	<code><param name="annotate" value="true"></code>
Add and edit annotations	<code><param name="annotateEdit" value="true"></code>

4. Optional: Configure more annotation options by adding parameters to the HTML file.

Related information:

 Annotations options

You can use the annotations parameters to determine annotation options such as availability, color, deletion, error message display, text, redaction, and save options.

 `annotate` parameter

This parameter determines whether all annotation features are available in the viewer.

 `annotateEdit` parameter

This parameter determines whether the Annotations toolbar is available to users.

 Annotations

You can use the annotation methods determine whether annotations are displayed and to configure annotations settings.

Configuring the annotations server object

A server object is required to send and receive annotations in IBM Daeja ViewONE. When the viewer starts an annotation save operation, this object is called to receive the annotation data that a user created or edited.

About this task

The server object is the interface between the viewer and the repository. This object must be able to receive and parse an annotations text stream. You can create the object as CGI, ASP, EXE, or a servlet. You can adapt how the object handles the annotation definitions to meet your specific environment. For example, the definitions might be written to a database, stored as text files next to their respective images, or passed on to another object.

To control users' access to annotation data, use a simple or complex security model with the appropriate code changes to the server object that handles the annotation data.

Configuring the object as a servlet

This approach uses the HTTP Post protocol to deliver the annotation data in a single stream of data.

About this task

After you create the server object, you use the **annotationSaveServlet** parameter to specify the location of the server object.

You can add parameters such as user ID or document ID to the query string part of the URL to provide user or document-specific annotations. The server object must process these parameters.

At save time, the annotations data is URL encoded and streamed to the servlet.

Related information:

 **annotationSaveServlet** parameter

This parameter determines the location of the server-side object that handles the saving of annotations. You can use this parameter to set up the interface between the viewer and the back-end repository for saving and retrieving annotation data.

Configuring the object by using URL-encoded data

This approach uses the HTTP Post protocol with the URL-encoded content type to deliver the annotation data in chunked data segments. Therefore, this approach is inherently more complex than the **annotationSaveServlet** approach.

About this task

You use the **annotationSavePost** parameter to specify the location of the POST object. The value of this parameter must be a URL that is relative to the codebase or is an absolute value.

You can add parameters such as user ID or document ID to the POST data by using the **annotationPostPrefix** parameter. IBM Daeja ViewONE tags on its own annotations data.

At save time, the annotations data is URL-encoded and put into a series of segments. Each segment can contain up to 64 KB of encoded data. The number of segments that are generated depends on the amount of annotations data to be sent.

The following parameters are then added to the URL that is specified by the **annotationSavePost** parameter.

Parameter	Description
size	The size in bytes of the URL encoded annotations data that is being sent.
numdata	The total number of data segments that are to be sent.
data<N>	The URL-encoded annotations data, where<N> is the sequential identifier for the segment.

IBM Daeja ViewONE then calls the server object and passes the data with a content-type of `application/x-www-form-urlencoded`.

On the server, IBM Daeja ViewONE retrieves the annotations data from the URL by concatenating the **data<N>** segments. URL-encoded data is normally handled by the web server, so this approach is not an issue. You can check the data that is retrieved by comparing the value of the size parameter that is passed in the URL to the size of the data that is received.

Each time that IBM Daeja ViewONE calls the object to save an annotation, the object sends back a confirmation. This confirmation takes the form `<OK>` for a successful save operation or `FAILED` for a failed save operation.

Related information:

 `annotationSavePost` parameter

This parameter determines the location of the server-side object that handles the saving of annotations.

Configuring annotation security

You can implement simple security for the ViewONE Annotations Module by using the **EDIT** and **VIEW** annotation properties. Alternatively, you can use HTML parameters to provide extended security.

About this task

Ensure that your annotation server-side components can handle the additional security data with the logged on user ID so that the permissions are handled correctly.

Related reference:

“EDIT property” on page 19

The EDIT property provides a Boolean flag that indicates whether the annotation can be edited. This property is required for each annotation type.

“VIEW property” on page 28

The VIEW property specifies an integer value for the print and view properties of an annotation.

Simple annotation security model:

You can implement basic security for the ViewONE Annotations Module by using the EDIT and VIEW annotation properties to define whether a user can edit, view, or print a particular annotation. You can also use the **annotateEdit** HTML parameter to disable the edit option for all annotations.

The annotations file is an ASCII text file with the optional `.ant` extension that lists the properties for all annotations in an image. You can implement annotation

simple security through your servlet object that controls the content of the annotation file when it is served out to users.

By default, the EDIT annotation property is not included in the annotations file, and anyone can edit the annotations for that image. When a user creates and saves an annotation by using the user interface, the annotations file contains details about the annotation that do not include the EDIT property. That user has edit privileges for that annotation. When another user retrieves the annotation file without the EDIT property included, that user can edit that annotation.

However, the server object that serves up the annotation file can insert the EDIT property for any annotation. Setting this property to 0 (zero) prevents the user from editing that particular annotation. Setting this property to 1 allows the user to edit the annotation. You can set the property selectively for each annotation.

The same principle applies for the VIEW annotation property. By default, the VIEW annotation property is not included in the annotations file, and all users can view and print the annotation. Setting this property to 1 allows viewing of the annotation but not printing. Setting this property to 2 allows printing but not viewing. The default value is 3, allow viewing and printing the annotation.

To disable the edit option for all annotations, you can also use the **annotateEdit** HTML parameter in the HTML file that loads the viewer. This parameter is a global option that prevents the specified user from editing or modifying any annotation.

Related reference:

“EDIT property” on page 19

The EDIT property provides a Boolean flag that indicates whether the annotation can be edited. This property is required for each annotation type.

“VIEW property” on page 28

The VIEW property specifies an integer value for the print and view properties of an annotation.

Related information:

 [annotateEdit parameter](#)

This parameter determines whether the Annotations toolbar is available to users.

Extended annotation security model:

You can use HTML parameters to provide extended security for the ViewONE Annotations Module. For example, you can give users control over the privileges that are assigned to annotations they create. In addition, you can provide separate privileges for modifying, deleting, and reading annotations.

The extended option relies on you using the **userId** HTML parameter and then adding the **annotationSecurityModel** HTML parameter with a value of "2" as shown in the following example:

```
<param name="userId" value="user1">  
<param name="annotationSecurityModel" value="2">
```

This code enables, as default, extended security options for each new annotation that is created by a user.

When the user saves annotations, extra properties are written to the annotations file, as follows:

```
SECURITYMODEL = 2
READ = 1
MODIFY = 1
EXECUTE = 1
PRINT = 1
DELETE = 1
PASSWORDMODIFY = *****
PASSWORDSECURITY = *****
OWNER = Whoever
MODIFYSECURITY = 1
```

There are separate privileges for read, modify, execute, print, delete, and modify-security. Each of these privileges can be changed by either the user or the server object that serves the file back to the user.

By default, the privileges are all set to 1 (enabled). The MODIFYSECURITY property indicates whether the user can edit these privileges by using the user interface. If MODIFYSECURITY is enabled, a padlock context button is displayed when the user selects the annotation. If the user click this button, a security dialog box is displayed for the user to change any of the security properties.

The owner of an annotation is the user that created the annotation and is identified by the owner property that is written to the annotations file. If this owner property matches the **userId** HTML parameter, then that user always has access to this security dialog.

Alternatively, you can specify the HTML parameter **userAdmin**. If you set this parameter to true, then irrespective of owner and user ID values, the current user is considered an administrator who can edit annotation security for any annotation.

To prevent other users, excluding administrators, from modifying an annotation, a user can clear the **modify** option in the Annotation Security dialog box. For example, other users cannot select the annotation or move it. If the user selects the **modify** option and clears the **delete** option, then other users can modify the annotation but they cannot delete it. If the user clears the **read** option, the other users are not able to see the annotation. If user clears the **print** option, then users are not able to print the annotation.

The EXECUTE option controls the action for annotation hyperlinks. For example, if an annotation has a hyperlink, but the EXECUTE option is disabled, then other users are not able to use that hyperlink.

If a document has existing annotations that were created by using the simple annotation security model, the extended options can be enabled on those annotations. The server object that is serving the annotations file must add the "SECURITYMODEL = 2" line to each annotation at retrieval time.

If the **annotationEditPasswordModify** parameter is set to true, then the user who creates the annotation can set a password that must be entered before a user can modify an annotation. If the **annotationEditPasswordSecurity** parameter is set to true, then it is possible to set a password that must be entered before a user can modify annotation security.

The passwords are saved in the annotations file by using the properties PASSWORDMODIFY and PASSWORDSECURITY. The passwords are private-key 32 bit encrypted so they are not viewable or editable except through IBM Daeja ViewONE. If an OWNER or an ADMIN user forgets a password that is assigned to

an annotation, that user can clear and reenter a password. All other users cannot, unless they enter the correct modify annotation security password.

Passwords are not displayed anywhere. Asterisks (*) are displayed in the dialog boxes and in the annotations file.

Related information:

 Security

You can use the annotation security parameters to determine the default security settings for specified annotations. You can specify whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security. The parameters also configure the administrator that can edit security settings and password options in the security dialog box.

 annotationEditPasswordModify parameter

When this parameter is set to true, a further option is presented in the security dialog box when you use extended annotation security.

 annotationEditPasswordSecurity parameter

When this parameter is set to true, a further option is presented in the security dialog box when you use extended annotation security.

 annotationSecurityModel parameter

This parameter determines which model is used for annotation security.

 userAdmin parameter

When this parameter is set to true, the current user is specified as being an administrator, with the ability to edit security settings for any annotation.

 userID parameter

This parameter specifies the value to identify a user for user-specific areas of IBM Daeja ViewONE.

Creating an annotation template

Annotation data is stored in a separate file from the document. You can create or edit an annotation definition file to use as a template to apply annotations consistently to your documents.

About this task

For example, you might add a corporate logo stamp to the bottom of every document. Or you might apply redactions to a standard form.

The annotations file is an ASCII text file with the optional .ant extension. Annotations are defined through a set of mandatory and optional properties, and each property must be defined on a separate line.

Typically, the file is not case-sensitive except for where text properties are defined. Blank lines are ignored.

The start of an annotation definition is denoted by an opening bracket. This bracket is followed by a type identifier that indicates the kind of annotation that is being defined. In the following example, the type identifier is ARROW. The identifier is ended by a right square bracket. A carriage-return, a linefeed, or a carriage-return and linefeed then indicate the end of the line. The end of an annotation definition is denoted by the start of another annotation definition or by the end of the file.

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
```

Procedure

To create the annotations definition file:

1. Use a text editor to create a new text file with the `.ant` extension or open an existing annotation definition file.

Tip: You can begin by drawing the annotations in the viewer, and then modify the annotation definition file.

2. Add a definition section for each annotation and edit the properties as needed.
3. Save the file to the document repository.
4. To display the annotations definition file in a document, set the `annotationFile` parameter to point to the location of the `.ant` file.

Annotation definition markers and common properties

Each annotation definition consists of an annotation type, markers, and the related annotation properties. The common properties are used by most annotation types.

EMPTY marker:

The EMPTY marker indicates that no annotations are present in the document when annotations are saved. In this situation, IBM Daeja ViewONE returns a file with just [EMPTY] inside.

When an annotations definition file is loaded, it must contain either annotation definitions or the [EMPTY] marker. If it does not, then an error message is displayed and the document is not shown.

ANNOTINFO marker:

The ANNOTINFO marker indicates that annotations were saved in V4.0.2 or later of the viewer. These annotations do not load in earlier versions of the viewer that do not recognize the ANNOTINFO marker. If you want to maintain compatibility with an earlier version, for example, for evaluating a new version of the viewer, you can use the `disableAnnotInfo` parameter.

The ANNOTINFO marker identifies the module and the version of the module that was used to view the document when the annotation data was saved. This marker enables the viewer to be configured to use the same module and version with which the annotation data was added.

Annotations can move and even to leave the page if they are viewed with different modules or with different versions of the same module. For example, this issue can occur in the following situations:

- The annotation data was created with a version earlier than 4.0.2 and does not have the ANNOTINFO marker. Therefore, the viewer cannot tell which module was used to annotate the documents originally. This lack can mean that annotation data moves if users use a different module to that used to annotate

the document originally. For example, this problem can occur if the user previously viewed and annotated emails by using the ViewONE Universal Viewing Module but now uses the ViewONE Office Module to view these same documents.

- Enhancements or bug fixes to the modules might cause changes to the way that documents are laid out. For example, a fix to an issue with a table in a border on a Word document, a single page Word document is now a two pages. Thus, annotations might move to a different page.

To address these issues, version 4.0.2 and later of the viewer include a number of parameters that provide tight control of the viewer's behavior when it encounters various scenarios. These parameters affect only Microsoft Office and email documents that have annotations. Other documents always load in the appropriate module that is detected by the viewer.

Related information:

 `disableAnnotInfo` parameter

This parameter determines whether version checking is disabled.

EDIT property:

The EDIT property provides a Boolean flag that indicates whether the annotation can be edited. This property is required for each annotation type.

This property uses the following format:

`EDIT = binary_value`

When *binary_value* is set to 0, the annotation cannot be edited.

When *binary_value* is set to 1, the annotation can be edited.

Example

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
```

PAGE property:

The PAGE property specifies a page for the annotation. This property is required for each annotation type.

This property uses the following format:

`PAGE = n`

If the value specified is -1, then the annotation is displayed across all pages in the document. To display the annotation across all pages, IBM Daeja ViewONE produces a copy of the annotation for each page of the document. If the annotations are then saved, these copies are returned as individual annotations in the definitions file (along with their appropriate page number) and are treated as independent annotations thereafter.

If the value specified is -2, then the behavior is the same except that the annotations are not treated as independent annotations. A change to the annotation on one of the pages is reflected across all the pages automatically.

The LABEL properties for these new annotations are a copy of the original LABEL property with a page number appended. This feature makes these annotations unsuitable for use with hyperlinks, as their names change automatically upon creation.

Example

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
```

BLANKOUTIMAGE property:

The BLANKOUTIMAGE property specifies a Boolean value that indicates whether the page that contains the annotation is to be blanked out rather than displayed.

This property uses the following format:

```
BLANKOUTIMAGE = 1
```

This property provides a security option that allows an image to be blocked from view but for its annotations to still be visible.

For example, if a user cannot view a particular page of a document then a simple text annotation might be set up on that page that says Sorry, viewing denied. The BLANKOUTIMAGE property for this annotation is set to 1. The user is not able to edit or modify this annotation, even if the EDIT property is set to 1.

Include the BLANKOUTIMAGE property only when the value is set to 1.

COLOR property:

The COLOR property specifies the color of the annotation in either comma-delimited RGB values or the IBM Daeja ViewONE color scheme text.

This property uses the following format:

```
COLOR = color
```

The default value is 255, 0, 0 (red). You can change the default color by using the **annotationsLineColor** parameter or the **annotationsTextColor** parameter.

Example

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
```

Related reference:

➤ annotationLineColor parameter

This parameter determines the default color for lines.

➤ annotationTextColor parameter

This parameter specifies the default color for annotation text.

Related information:

➤ IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

CREATEDATE property:

The CREATEDATE property specifies the date that the annotation was created.

This property uses the following format:

CREATEDATE = *dd mm yyyy, hh:mm:ss* [, *loc±hh:mm*]

The optional *loc±hh:mm* is a local time zone stamp. The *loc* variable stands for location, and is a three letter identifier for the time zone. The *hh:mm* variable specifies the amount that the local time was offset from that location's time zone. The "+" or "-" then specify whether that time is ahead or behind the time zone.

Example 1

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
CREATEDATE = 12 Jun 2001, 18:27:05
```

Example 2

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
CREATEDATE = 12 Jun 2001, 18:27:05, GMT+01:00
```

CREATEDID property:

The CREATEDID property specifies the user ID value that was in use when the annotation was created.

This property uses the following format:

CREATEDID = *user ID*

If the user ID is not set at the time the annotation is saved, the property is not returned in the definitions file.

Related information:

➤ userID parameter

This parameter specifies the value to identify a user for user-specific areas of IBM

Daeja ViewONE.

CUSTOMPROPERTY property:

The CUSTOMPROPERTY property specifies custom text that is to be kept with the annotation.

This property uses the following format:

CUSTOMPROPERTY = *any text*

This text is not interpreted by IBM Daeja ViewONE. However, if the CUSTOMPROPERTY property is defined, IBM Daeja ViewONE retrieves the text when the annotations file is read and writes the text out again at save time. The property is written back at save time only if the property came from a valid annotation definition

Just as the CUSTOM annotation type allows information to be associated with an annotation definition file, the CUSTOMPROPERTY property allows information to be associated with individual annotations.

Related reference:

“CUSTOM annotation type” on page 30

This annotation type defines a custom annotation type. The custom annotation type is not really an annotation, but is a way to add extra information to an annotations definition file.

HYPERLINK property:

The HYPERLINK property defines actions to be taken with pages, annotations, JavaScript, and web pages.

This property uses the following format:

HYPERLINK = *<hyperlink type><hyperlink property>[<hyperlink property>]*

This property defines the action that is to be taken in the following situations:

- The user clicks an annotation.
- The user double-clicks the annotation and the **annotationDb1Click** HTML parameter is used.
- The annotations toolbar is open in IBM Daeja ViewONE.

Important: You must include the angle brackets (< >) in the HYPERLINK property as shown.

The following table lists the four types of hyperlinks that you can define:

Hyperlink type	Description
page	An internal hyperlink that takes the user to a specific page in the document.
annotation	An internal hyperlink that takes the user to a specific annotation in the document.
javascript	A hyperlink that calls the JavaScript event handler.
web	An external hyperlink that opens another web page.

Each of these hyperlink types has parameters that you specify.

The easiest way to set up annotation hyperlinks is to use the IBM Daeja ViewONE user interface.

Before you define hyperlinks in a definitions file, test the hyperlinks in the IBM Daeja ViewONE user interface to see how they work. For example, both a JavaScript hyperlink and a web hyperlink can be used to open a web page. However, you might find that a JavaScript hyperlink works better for some tasks because the JavaScript hyperlink gives flexibility over how the window looks.

JavaScript hyperlinks can also be used for any other task, such as opening another document or starting a custom job. However, JavaScript hyperlinks require the use of the IBM Daeja ViewONE JavaScript event-handler. Therefore, these hyperlinks are more complex to set up initially.

Page hyperlinks

This property uses the following format:

```
HYPERLINK = <page><page number>
```

If the page number is not in the document's page range, the hyperlink does not work. Instead, the user remains on the current page.

Annotation hyperlinks

This property uses the following format:

```
HYPERLINK = <annotation><annotation label>
```

The *annotation label* variable specifies the label of the annotation that the hyperlink links to. If a link is made to a label name that is shared by two or more annotations in a document, IBM Daeja ViewONE links to the first of those annotations.

By default, IBM Daeja ViewONE scrolls the page to as close to the upper left of the linked annotation as possible. IBM Daeja ViewONE uses the current scale preference such as fit to height or fit to width. However, if window settings such as a scroll position or rotation angle are associated with the linked annotation, IBM Daeja ViewONE uses these settings instead. For more information, see the HYPERLINKSETTINGS property.

JavaScript hyperlinks

This property uses the following format:

```
HYPERLINK = <javascript><any text>
```

When a user clicks the annotation for which a JavaScript hyperlink is defined, IBM Daeja ViewONE calls the designated JavaScript event handler.

The *any text* variable is used as the event handler's text property and can be interpreted in any way. For example, it might be interpreted as a document identifier, a web page, or a comment. The event identifier is 23.

A JavaScript hyperlink is useful for designing implementation-specific hyperlink behavior. For example, after the JavaScript event handler receives the hyperlink's

text string, the event handler might display a dialog box with help information. Alternatively, the event handler might use IBM Daeja ViewONE JavaScript calls to open a new document.

Web page hyperlinks

This property uses the following format:

```
HYPERLINK = <web><web page url><target>
```

The *web page url* variable specifies the URL of the web page that is to open when the user clicks the annotation. Use the standard format for entering this URL, for example, `<http://www.mysite.com/page.html>` The web page URL can be relative to the IBM Daeja ViewONE codebase parameter or it can be absolute. If you use the **annotationHyperlinkWeb** HTML parameter, a relative URL is relative to the base address as defined by that parameter.

The *target* variable specifies the window in which the new web page is displayed. For example, setting the *target* variable to `<_self>` causes the browser to open the new page in the web page that currently displays IBM Daeja ViewONE. Setting the *target* variable to `<_blank>` opens a new window that contains the new web page.

The *target* variable can be one of the four default values that are assigned by IBM Daeja ViewONE. The following table lists these default values for web hyperlinks. For each target, the table lists the default text for the menu label.

Target	Menu label text
<code><_self></code>	<code><Same window></code>
<code><_blank></code>	<code><New window></code>
<code><_parent></code>	<code><Parent frame></code>
<code><_top></code>	<code><Top frame></code>

Alternatively, you can specify a custom target for the *target* variable by using the **annotationTarget** HTML parameter.

Adding annotation tooltips to help with hyperlinks

When you create any annotation, you can use the **Help text and hyperlink** option to add a tooltip that provides explanatory text when a user hovers the mouse pointer over the annotation. For example, if you define a hyperlink annotation to jump to another page, you can include a tooltip to tell users what happens when they click the annotation. The tooltip might say "Click here to see page 5" or "Click here to see the index."

Related reference:

"HYPERLINKSETTINGS property" on page 25

The HYPERLINKSETTINGS property defines window settings for an annotation. These settings are used when this annotation is accessed from another annotation by using the HYPERLINK property.

Related information:

 [userID parameter](#)

This parameter specifies the value to identify a user for user-specific areas of IBM Daeja ViewONE.

➡ annotationDbClick parameter

This parameter determines the mouse-click operation that is used to activate annotations.

➡ annotationHyperlinkWeb parameter

This parameter determines the default base location for web hyperlinks. If you define this parameter, relative hyperlinks are then relative to the specified URL.

➡ annotationTarget<N> parameter

This parameter specifies a list of optional targets for web hyperlinks to use in place of IBM Daeja ViewONE default targets.

HYPERLINKSETTINGS property:

The HYPERLINKSETTINGS property defines window settings for an annotation. These settings are used when this annotation is accessed from another annotation by using the HYPERLINK property.

This property uses the following format:

```
HYPERLINKSETTINGS = <view#scale#flip#rotation  
#invert#zoom#scrollx#scrolly  
#brightness#contrast#luminance>
```

The following table explains the variables in this value:

Variable	Value
<i>view</i>	Currently ignored but reflects the view mode used (use a value of 1)
<i>scale</i>	0: Fit to width 1: Fit to height 2: Best fit 3: Zoom
<i>flip</i>	0: No flip 1: Flip horizontally 2: Flip vertically 3: Flip both ways
<i>rotation</i>	One of the following values to indicate the angle in degrees: 0, 90, 180, 270
<i>invert</i>	0: Colors not inverted 1: Colors inverted
<i>zoom</i>	Zoom factor, for example, 2.0
<i>scrollX</i>	Percentage of image to scroll in the x-axis
<i>scrollY</i>	Percentage of image to scroll in the y-axis
<i>brightness</i>	0 - 512, where 0 is dark, 512 is light, and 256 is the middle value
<i>contrast</i>	0 - 512, where 256 is the middle value
<i>luminance</i>	0 - 512, where 256 is the middle value

The format of this property allows a wide range of window settings to be defined. If you are unsure what value to use for this property, you can get a value as follows:

1. In the viewer, select the annotation that you want to link to.
2. Zoom in, scroll, rotate, and so on, so that the page appears in the way you want it to be viewed.
3. Right-click the annotation to select it in edit mode, and then click the **H** hyperlink button in the context toolbar.
4. Click **Grab window settings** and save the annotation. Note the annotations label so that you know which annotation to link to when you define the hyperlink from another annotation.
5. Review the annotations definition file that IBM Daeja ViewONE created. This file might be on the server if you use the **annotationSave** HTML parameter or in the ANT file if the image was loaded from a local file.
6. Locate the annotation and review the HYPERLINKS value that is included for the window settings that you used. The format for this property is as follows:

Example


```
HYPERLINKSETTINGS = <1#3#0#90#1#2.0#20#14#256#256#512>
```

Related reference:

"HYPERLINK property" on page 22

The HYPERLINK property defines actions to be taken with pages, annotations, JavaScript, and web pages.

Related information:

 docThumb<N> parameter

This parameter determines whether a separate thumbnail file is specified for each document. Having separate files for document thumbnails can be advantageous because the smaller files are quicker to download and view.

LABEL property:

The LABEL property specifies an identifier for the annotation. Carriage-returns and line feeds are not permitted within the value.

This property uses the following format:

```
LABEL = any text
```

The default value is the annotation type, for example, "Arrow" or "Oval." Leading and trailing white space is removed when the server opens or closes the annotation definition file.

Example

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
LABEL = Arrow
```

LINEWIDTH property:

The LINEWIDTH property specifies the width of the annotation's lines in image pixels.

This property uses the following format:

```
LINEWIDTH = n
```

The default value is 0.

Example

```
[ARROW]  
X1 = 535  
Y1 = 277  
X2 = 89  
Y2 = 138  
PAGE = 1  
EDIT = 1  
COLOR = 255, 0, 0  
LINEWIDTH = 4
```

MODIFIEDDATE property:

The MODIFIEDDATE property specifies the date that the annotation was last modified.

This property uses the following format:

```
MODIFIEDDATE = dd mm yyyy, hh:mm:ss [, loc±hh:mm]
```

The optional *loc±hh:mm* is a local time zone stamp. The *loc* variable stands for location, and is a three letter identifier for the time zone. The *hh:mm* variable specifies the amount that the local time was offset from that location's time zone. The "+" or "-" then specify whether that time is ahead or behind the time zone.

Example 1

```
[ARROW]  
X1 = 535  
Y1 = 277  
X2 = 89  
Y2 = 138  
PAGE = 1  
EDIT = 1  
COLOR = 255, 0, 0  
MODIFIEDDATE = 12 Jun 2001, 18:27:05
```

Example 2

```
[ARROW]  
X1 = 535  
Y1 = 277  
X2 = 89  
Y2 = 138  
PAGE = 1  
EDIT = 1  
COLOR = 255, 0, 0  
MODIFIEDDATE = 12 Jun 2001, 18:27:05, GMT+01:00
```

MODIFIEDID property:

The MODIFIEDID property specifies the user ID value that was in use when the annotation was last modified.

This property uses the following format:

```
MODIFIEDID = user ID
```

If the user ID is not set at the time the annotation is saved, the property is not returned in the definitions file.

Related information:

 [userID parameter](#)

This parameter specifies the value to identify a user for user-specific areas of IBM Daeja ViewONE.

PAGESIZE property:

The PAGESIZE property specifies the main page's width and height that is associated with the annotation.

This property uses the following format:

PAGESIZE = *width, height*

This property is required only if annotations are displayed in thumbnails where thumbnails are specified as separate files by using the **docThumb<N>** HTML parameter. In such cases, the width and height values must be those values for the original main image and not the scaled thumbnail image. IBM Daeja ViewONE is then able to rescale any annotations that might be present on that page to fit the thumbnail image.

PAGEURL property:

The PAGEURL property shows the URL that IBM Daeja ViewONE used to retrieve the page on which the annotation was placed.

This property uses the following format:

PAGEURL = *pageURL*

The property is written out by IBM Daeja ViewONE, but is ignored upon reading. The property allows server processes that handle the annotations definition file to see with which image file each annotation and each page is associated.

VIEW property:

The VIEW property specifies an integer value for the print and view properties of an annotation.

This property uses the following format:

VIEW = *n*

The possible values are as follows:

Value	Description
1	Viewing only (no printing)
2	Printing only (no viewing or editing by the user)
3	Printing and viewing

The default value is 3.

IBM Daeja ViewONE does not return the property in the definition file if the value is 3.

Example

```
[ARROW]
X1 = 535
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
COLOR = 255, 0, 0
VIEW = 1
```

Annotation types and their properties

IBM Daeja ViewONE provides various types of annotations. Each annotation type has required properties and optional properties.

The optional properties that are described in Annotation definition common properties apply to all annotation types except where noted. For example, the COLOR property applies to all annotation types.

You can change the default values for some properties by using HTML parameters or JavaScript methods that are described in the relevant reference topics.

Related reference:

“Annotation definition markers and common properties” on page 18

Each annotation definition consists of an annotation type, markers, and the related annotation properties. The common properties are used by most annotation types.

ARROW annotation type:

This annotation type defines an annotation that is an arrow. You must define two points for the arrow. The arrowhead is drawn at the first point.

Definition

```
[ARROW]
```

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the ARROW annotation type:

Property	Description
X1 = <i>n</i>	The coordinate of the arrow's first point on the X axis of the image. This point is an image pixel value.
Y1 = <i>n</i>	The coordinate of the arrow's first point on the Y axis of the image. This point is an image pixel value.
X2 = <i>n</i>	The coordinate of the arrow's second point on the X axis of the image. This point is an image pixel value.
Y2 = <i>n</i>	The coordinate of the arrow's second point on the Y axis of the image. This point is an image pixel value.

Optional properties

You can use any of the common properties in the definition of the ARROW annotation type.

The minimum value of the LINEWIDTH property is 1.

In addition to the common properties, you can include the following property in the definition of the ARROW annotation type:

Property	Description
ARROWHEADSIZE = <i>n</i>	The size of the arrowhead. The value must be at least 1 and has no fixed upper value. The default value is 1.

Example:

```
[ARROW]
X1 = 537
Y1 = 277
X2 = 89
Y2 = 138
PAGE = 1
EDIT = 1
ARROWHEADSIZE = 1
```

CUSTOM annotation type:

This annotation type defines a custom annotation type. The custom annotation type is not really an annotation, but is a way to add extra information to an annotations definition file.

The CUSTOM annotation type cannot be added from the user interface, but can be generated through a server process that adds instances to a definitions file. An instance of the CUSTOM annotation type is not interpreted by IBM Daeja ViewONE. Instead, IBM Daeja ViewONE recognizes the entry when the annotations file is read and then writes out the entry at save time. The entry might not be in the same place when it is written out again.

The CUSTOM type allows server processes to track custom information between definition time and save time without requiring IBM Daeja ViewONE to interpret the information.

Definition

```
[CUSTOM]
```

Required properties

You cannot include any of the required properties in the definition of the CUSTOM annotation type.

Optional properties

You cannot include any of the optional properties in the definition of the CUSTOM annotation type.

Example:

[CUSTOM]

This text shows that you can put anything into a CUSTOM annotation.

Text 1

Text 2

Text 3

I was generated by server "nautilus1" @ 18:31 GMT on 27/03/02

FREEHAND annotation type:

This annotation type defines an annotation that is a freehand line. You must include a minimum of three points in the definition.

Definition

[FREEHAND]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the FREEHAND annotation type:

Property	Value
X1 = <i>n</i>	The coordinate of the line's first point on the X axis of the image. This point is an image pixel value.
Y1 = <i>n</i>	The coordinate of the line's first point on the Y axis of the image. This point is an image pixel value.
X2 = <i>n</i>	The coordinate of the line's second point on the X axis of the image. This point is an image pixel value.
Y2 = <i>n</i>	The coordinate of the line's second point on the Y axis of the image. This point is an image pixel value.
X3 = <i>n</i>	The coordinate of the line's third point on the X axis of the image. This point is an image pixel value.
Y3 = <i>n</i>	The coordinate of the line's third point on the Y axis of the image. This point is an image pixel value.

Optional properties

You can use any of the common properties in the definition of the FREEHAND annotation type.

The minimum value of the LINEWIDTH property is 1.

In addition to the common properties, you can include the following properties in the definition of the FREEHAND annotation type:

Property	Value
$X_n = value$	The X coordinate of point n on the line.
$Y_n = value$	The Y coordinate of point n on the line.

Example:

```
[FREEHAND]
X1 = 408
Y1 = 940
X2 = 408
Y2 = 940
X3 = 412
Y3 = 936
X4 = 412
Y4 = 927
X5 = 425
Y5 = 907
PAGE = 1
EDIT = 1
```

HIGHLIGHT annotation type:

This annotation type defines a highlight annotation that is a rectangle.

Definition

```
[HIGHLIGHT]
```

Mandatory properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the HIGHLIGHT annotation type:

Property	Value
$X = n$	The coordinate of the rectangle's upper left corner on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the rectangle's upper left corner on the Y axis of the image. The coordinate is an image pixel value.
$WIDTH = n$	The width of the rectangle in image pixels.
$HEIGHT = n$	The height of the rectangle in image pixels.

Optional properties

You can use any of the common properties in the definition of the HIGHLIGHT annotation type.

The minimum value of the LINEWIDTH property is 0. The LINEWIDTH property is ignored if it is not set to 0.

In addition to the common properties, you can include the following properties in the definition of the HIGHLIGHT annotation type:

Property	Value
FILLCOLOR = <i>color</i>	The highlight color. You can specify the color by using either comma-delimited RGB values or one of the IBM Daeja ViewONE color scheme values. If no color is defined, the highlight is invisible.
TRANSPARENT = 0	Boolean for whether the highlight is transparent. False=0, True=1. The default value is 0.
ASPECTRATIO=1:1	When set to 1:1, forces the highlight to be a square

Example:

```
[HIGHLIGHT]
X = 674
Y = 61
WIDTH = 310
HEIGHT = 176
PAGE = 1
EDIT = 1
LINEWIDTH = 0
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:



IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

HIGHLIGHTPOLYGON annotation type:

This annotation type defines a highlight annotation that is a polygon. You must define a minimum of three points for this type.

Definition

[HIGHLIGHTPOLYGON]

Mandatory properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the HIGHLIGHTPOLYGON annotation type:

Property	Value
X1 = <i>n</i>	The coordinate of the polygon's first point on the X axis of the image. The coordinate is an image pixel value
Y1 = <i>n</i>	The coordinate of the polygon's first point on the Y axis of the image. The coordinate is an image pixel value
X2 = <i>n</i>	The coordinate of the polygon's second point on the X axis of the image. The coordinate is an image pixel value

Property	Value
Y2 = <i>n</i>	The coordinate of the polygon's second point on the Y axis of the image. The coordinate is an image pixel value
X3 = <i>n</i>	The coordinate of the polygon's third point on the X axis of the image. The coordinate is an image pixel value
Y3 = <i>n</i>	The coordinate of the polygon's third point on the Y axis of the image. The coordinate is an image pixel value

Optional properties

You can use any of the common properties in the definition of the HIGHLIGHTPOLYGON annotation type.

The minimum value of the LINEWIDTH property is 0, which removes the border around the annotation.

In addition to the standard optional properties, you can include the following properties in the definition of the HIGHLIGHTPOLYGON annotation type:

Property	Value
X <i>n</i> = <i>value</i>	The X coordinate of point <i>n</i> in the polygon.
Y <i>n</i> = <i>value</i>	The Y coordinate of point <i>n</i> in the polygon.
FILLCOLOR = <i>color</i>	The highlight color. You can specify the color by using either comma-delimited RGB values or one of the IBM Daeja ViewONE color scheme values. If no color is defined, the highlight is invisible.
TRANSPARENT = 0	A Boolean value that indicates whether the highlight is transparent. False=0, True=1. The default value is 0.

Example:

```
[HIGHLIGHTPOLYGON]
X1 = 1246
Y1 = 61
X2 = 1581
Y2 = 220
X3 = 1356
Y3 = 371
X4 = 1144
Y4 = 200
PAGE = 1
EDIT = 1
LINEWIDTH = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

NOTE annotation type:

This annotation type defines an annotation that is a note. The definition includes the text for this note.

Definition

[NOTE]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the NOTE annotation type:

Property	Value
$X = n$	The coordinate of the rectangle's upper left corner on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the rectangle's upper left corner on the Y axis of the image. The coordinate is an image pixel value.
$WIDTH = n$	The width of the rectangle in image pixels.
$HEIGHT = n$	The height of the rectangle in image pixels.
$TEXT = any\ text$	The text that is to be included in the note. Indicate new lines by using the <N> identifier. Leading and trailing white space is removed when the annotation is saved or loaded.

Optional properties

You can use any of the common properties except the following properties in the definition of the NOTE annotation type:

- LINEWIDTH
- HYPERLINK

In addition to the common properties, you can include the following properties in the definition of the NOTE annotation type:

Property	Value
$FILLCOLOR = color$	The background color of the note. You can specify the value by using either comma-delimited RGB values or the IBM Daeja ViewONE color scheme value. The default values are 255, 255, 153 (pale yellow).
$TRANSPARENT = 0$	A Boolean value that indicates whether the note is transparent. False=0, True=1. The default value is 0.
$RECTANGULAR = 0$	A Boolean value that indicates whether the note style is rectangular (that is, without leaf turn up). False=0, True=1. The default is 0.

Example:

```
[NOTE]
X = 81
Y = 323
WIDTH = 50
HEIGHT = 65
TEXT = This is line 1<N>This is line 2
PAGE = 1
EDIT = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
RECTANGULAR = 0
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

OPENPOLYGON annotation type:

This annotation type defines an annotation that is an open-ended polygon. You must define a minimum of three points for the polygon.

Definition

[OPENPOLYGON]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the OPENPOLYGON annotation type:

Property	Value
X1 = <i>n</i>	The coordinate of the polygon's first point on the X axis. The coordinate is an image pixel value.
Y1 = <i>n</i>	The coordinate of the polygon's first point on the Y axis. The coordinate is an image pixel value.
X2 = <i>n</i>	The coordinate of the polygon's second point on the X axis. The coordinate is an image pixel value.
Y2 = <i>n</i>	The coordinate of the polygon's second point on the Y axis. The coordinate is an image pixel value.
X3 = <i>n</i>	The coordinate of the polygon's third point on the X axis. The coordinate is an image pixel value.
Y3 = <i>n</i>	The coordinate of the polygon's third point on the Y axis. The coordinate is an image pixel value.

Optional properties

You can use any of the common properties in the definition of the OPENPOLYGON annotation type.

The minimum value of the LINEWIDTH property is 1.

In addition to the common properties, you can include the following properties in the definition of the OPENPOLYGON annotation type:

Property	Value
$X_n = value$	The X coordinate of point n in the polygon.
$Y_n = value$	The Y coordinate of point n in the polygon.

Example:

```
[OPENPOLYGON]
X1 = 1242
Y1 = 670
X2 = 1246
Y2 = 878
X3 = 927
Y3 = 882
X4 = 1017
Y4 = 658
PAGE = 1
EDIT = 1
```

OVAL annotation type:

This annotation type defines an annotation that is an oval.

Definition

```
[OVAL]
```

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the OVAL annotation type:

Property	Value
$X = n$	The coordinate of the oval's center on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the oval's center on the Y axis of the image. The coordinate is an image pixel value.
$WIDTH = n$	The oval's horizontal radius in image pixels.
$HEIGHT = n$	The oval's vertical radius in image pixels.

Optional properties

You can use any of the common properties in the definition of the OVAL annotation type.

The minimum value of the LINEWIDTH property is 0.

In addition to the common properties, you can include the following properties in the definition of the OVAL annotation type:

Property	Value
FILLCOLOR = <i>color</i>	The fill color of the annotation. You can specify the color by using either comma-delimited RGB values or one of the IBM Daeja ViewONE color scheme values. If no color is defined, the highlight is invisible.
TRANSPARENT = 0	A Boolean that indicates whether the oval is transparent. False=0, True=1. The default value is 1.
ASPECTRATIO=1:1	A property that is set to 1:1 to force the annotation to be a circle. Restriction: This property applies only to V3 of IBM Daeja ViewONE.

Example:

```
[OVAL]
X = 257
Y = 752
WIDTH = 143
HEIGHT = 130
PAGE = 1
EDIT = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

POLYGON annotation type:

This annotation type defines an annotation that is a closed polygon; that is, all sides of the polygon are complete. You must define a minimum of three points for the polygon.

Definition

[POLYGON]

Mandatory properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the POLYGON annotation type:

Property	Value
X1 = <i>n</i>	The coordinate of the polygon's first point on the X axis of the image. The coordinate is an image pixel value.
Y1 = <i>n</i>	The coordinate of the polygon's first point on the Y axis of the image. The coordinate is an image pixel value.
X2 = <i>n</i>	The coordinate of the polygon's second point on the X axis of the image. The coordinate is an image pixel value.

Property	Value
Y2 = <i>n</i>	The coordinate of the polygon's second point on the Y axis of the image. The coordinate is an image pixel value.
X3 = <i>n</i>	The coordinate of the polygon's third point on the X axis of the image. The coordinate is an image pixel value.
Y3 = <i>n</i>	The coordinate of the polygon's third point on the Y axis of the image. The coordinate is an image pixel value.

Optional properties

You can use any of the common properties in the definition of the POLYGON annotation type.

The minimum value of the LINEWIDTH property is 0, which removes the border around the annotation.

In addition to the common properties, you can include the following properties in the definition of the POLYGON annotation type:

Property	Value
X <i>n</i> = <i>value</i>	The X coordinate of point <i>n</i> in the polygon.
Y <i>n</i> = <i>value</i>	The Y coordinate of point <i>n</i> in the polygon.
FILLCOLOR = <i>color</i>	The background color of the polygon. You can specify the value by using either comma-delimited RGB values or the IBM Daeja ViewONE color scheme value. The default values are 255, 255, 153 (pale yellow).
TRANSPARENT = 0	A Boolean value that indicates whether the polygon is transparent. False=0, True=1. The default value is 1.

Example:

```
[POLYGON]
X1 = 1528
Y1 = 457
X2 = 1634
Y2 = 694
X3 = 1442
Y3 = 809
X4 = 1360
Y4 = 621
PAGE = 1
EDIT = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

RECTANGLE annotation type:

This annotation type defines an annotation that is a rectangle.

Definition

[RECTANGLE]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the RECTANGLE annotation type:

Property	Value
$X = n$	The coordinate of the rectangle's upper left corner on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the rectangle's upper left corner on the Y axis of the image. The coordinate is an image pixel value.
$WIDTH = n$	The width of the rectangle in image pixels.
$HEIGHT = n$	The height of the rectangle in image pixels.

Optional properties

You can use any of the common properties in the definition of the RECTANGLE annotation type.

The minimum value of the LINEWIDTH property is 0.

In addition to the common properties, you can include the following properties in the definition of the RECTANGLE annotation type:

Property	Value
$FILLCOLOR = color$	The fill color of the annotation. You can specify the color by using either comma-delimited RGB values or one of the IBM Daeja ViewONE color scheme values. If no color is defined, the highlight is invisible.
$TRANSPARENT = 0$	A Boolean value that indicates whether the rectangle is transparent. False=0, True=1. The default value is 1.
$ASPECTRATIO=1:1$	A property that is set to 1:1 to force the annotation to be a square. Restriction: This property applies only to V3 of IBM Daeja ViewONE.

Example:

```
[RECTANGLE]
X = 690
Y = 269
WIDTH = 282
HEIGHT = 209
```

PAGE = 1
EDIT = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

REDACT annotation type:

This annotation type defines an annotation that is a rectangular redaction, that is, a filled rectangle.

Definition

[REDACT]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the REDACT annotation type:

Property	Value
$X = n$	The coordinate of the rectangle's upper left corner on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the rectangle's upper left corner on the Y axis of the image. The coordinate is an image pixel value.
$WIDTH = n$	The width of the rectangle in image pixels.
$HEIGHT = n$	The height of the rectangle in image pixels.

Optional properties

You can use any of the common properties in the definition of the REDACT annotation type.

The minimum value of the LINEWIDTH property is 0, which removes the border around the annotation.

In addition to the common properties, you can include the following properties in the definition of the REDACT annotation type:

Property	Value
$FILLCOLOR = color$	The fill color of the annotation. You can specify the color by using either comma-delimited RGB values or one of the IBM Daeja ViewONE color scheme values. If no color is defined, the highlight is invisible.
$TRANSPARENT = 0$	A Boolean value that indicates whether the redaction is transparent. False=0, True=1. The default value is 0.

Property	Value
ASPECTRATIO=1:1	A property that is set to 1:1 to force the annotation to be a square. Restriction: This property applies only to V3 of IBM Daeja ViewONE.

Example:

```
[REDACT]
X = 1058
Y = 371
WIDTH = 225
HEIGHT = 221
PAGE = 1
EDIT = 1
LINewidth = 0
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:

 [IBM Daeja ViewONE color scheme text](#)

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

REDACTPOLYGON annotation type:

This annotation type defines an annotation that is a redaction polygon, that is, a filled polygon. You must define a minimum of three points for the polygon.

Definition

[REDACTPOLYGON]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the REDACTPOLYGON annotation type:

Property	Value
X1 = <i>n</i>	The coordinate of the polygon's first point on the X axis of the image. The coordinate is an image pixel value.
Y1 = <i>n</i>	The coordinate of the polygon's first point on the Y axis of the image. The coordinate is an image pixel value.
X2 = <i>n</i>	The coordinate of the polygon's second point on the X axis of the image. The coordinate is an image pixel value.
Y2 = <i>n</i>	The coordinate of the polygon's second point on the Y axis of the image. The coordinate is an image pixel value.
X3 = <i>n</i>	The coordinate of the polygon's third point on the X axis of the image. The coordinate is an image pixel value.

Property	Value
Y3 = <i>n</i>	The coordinate of the polygon's third point on the Y axis of the image. The coordinate is an image pixel value.

Optional properties

You can use any of the common properties in the definition of the REDACTPOLYGON annotation type.

The minimum value of the LINEWIDTH property is 0, which removes the border around the annotation.

In addition to the common properties, you can include the following property in the definition of the REDACTPOLYGON annotation type:

Property	Value
Xn = <i>value</i>	The X coordinate of point <i>n</i> in the polygon.
Yn = <i>value</i>	The Y coordinate of point <i>n</i> in the polygon.
FILLCOLOR = <i>color</i>	The background color of the polygon. You can specify the value by using either comma-delimited RGB values or the IBM Daeja ViewONE color scheme value. The default values are 255, 255, 153 (pale yellow).
TRANSPARENT = 0	A Boolean value that indicates whether the polygon is transparent. False=0, True=1. The default value is 1.

Example:

```
[REDACTPOLYGON]
X1 = 702
Y1 = 519
X2 = 948
Y2 = 621
X3 = 845
Y3 = 792
X4 = 645
Y4 = 756
PAGE = 1
EDIT = 1
LINEWIDTH = 1
FILLCOLOR = 255, 255, 255
TRANSPARENT = 0
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

STAMP annotation type:

This annotation type defines an annotation that is an image overlay. This image overlay can be in any image format that is supported by IBM Daeja ViewONE.

Definition

[STAMP]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the STAMP annotation type:

Property	Value
$X = n$	The coordinate of the stamp's upper left corner on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the stamp's upper left corner on the Y axis of the image. The coordinate is an image pixel value.
RESOURCE = image:image file path<color>	<p>The absolute or relative path to the image file for the stamp. You must include the prefix <code>image:</code> before the path. You can include spaces in the path.</p> <p>Optionally, you can specify the <i>color</i> parameter to provide a transparent background color for the image.</p> <p>You must enclose the <i>color</i> value in angle brackets (< >). The value can be either comma-delimited RGB values or the IBM Daeja ViewONE color scheme value. The default values are 255, 255, 153 (pale yellow).</p>

Optional properties

You can use any of the common properties in the definition of the STAMP annotation type.

The standard optional properties COLOR and LINEWIDTH are not used with the STAMP annotation type.

In addition to the common properties, you can include the following properties in the definition of the STAMP annotation type:

Property	Value
SCALE = n	The scale multiplier. 100% = 1.0, 200% = 2.0, and so on. The default value is 1.0.
ROTATION = n	The rotation of the stamp in degrees. Valid ROTATION values are 0, 90, 180, 270, 360. The default is 0.
FONTHEIGHT = n	The font height to use for the stamp text in image pixels. The minimum value is 1.

Example:

```
[STAMP]
X = 134
Y = 1017
```

```

SCALE = 1.0
RESOURCE = image:mytif.tif<255, 255, 255>
ROTATION = 180
PAGE = 1
EDIT = 1
LINEWIDTH = 0

```

TEXT annotation type:

This annotation type defines an annotation that is a text overlay.

Important: IBM Daeja ViewONE does not use the user's font configuration to display a TEXT annotation. Instead, TEXT annotations are displayed by using predefined fonts that are based on Arial in different sizes. The predefined fonts are loaded from a font resource file that is installed with IBM Daeja ViewONE.

The ROTATION and TEXTROTATION properties both support rotation of text annotations. When you are integrating the viewer into an existing system, choose the type of rotation that is appropriate for that system. The TEXTROTATION property is the recommended property.

Definition

[TEXT]

Required properties

In addition to the required EDIT and PAGE properties, you must include the following properties in the definition of the TEXT annotation type:

Property	Value
$X = n$	The coordinate of the text overlay's first point on the X axis of the image. The coordinate is an image pixel value.
$Y = n$	The coordinate of the text overlay's first point on the Y axis of the image. The coordinate is an image pixel value.
TEXT = <i>any text</i>	The text that is displayed in the overlay. Indicate a new line by using the <N> identifier. Leading and trailing white space is removed when the annotation is saved or loaded.
FONTHEIGHT = n	The font height to use for the text in image pixels. The minimum value is 1.

Optional properties

You can use any of the common properties in the definition of the TEXT annotation type.

The LINEWIDTH property is not available for this annotation type.

In addition to the common properties, you can include the following properties in the definition of the TEXT annotation type:

Property	Value
FILLCOLOR = <i>color</i>	<p>The background color of the overlay. You can specify the value by using either comma-delimited RGB values or the IBM Daeja ViewONE color scheme value. The default values are 255, 255, 153 (pale yellow).</p> <p>This property is applied only if the TRANSPARENT property is set to 0. If this property is set to 1 and the PAGE specified is a color image, the overlay is given a visible fill color with the values 255, 255, 255 (white). Color transparencies are not supported.</p>
ROTATION = <i>n</i>	The rotation of the text clockwise relative to the page. The value is in degrees. Valid values are 0, 90, 180, 270, 360. The default is 0.
TEXTROTATION = <i>n</i>	The rotation of the text counterclockwise relative to the page. The value is in degrees. A valid value is an integer 0 - 359. The default is 0.
TRANSPARENT = 0	A Boolean value that indicates whether the overlay is transparent. False=0, True=1. The default value is 1.
SEMITRANSPARENT = 0	<p>A Boolean value that indicates whether the text is semi-transparent. False=0, True=1. The default value is 0.</p> <p>If this property is set to 1, then the fill color for the overlay is also set to semi-transparent.</p>
STRIKETHROUGH = 0	<p>A Boolean value that indicates whether the text is displayed with a line through the center. The default value is 0.</p> <p>This property applies to the following versions:</p> <ul style="list-style-type: none"> • IBM Daeja ViewONE Standard V3.0.332 or later. • IBM Daeja ViewONE Professional V1.0.332 or later.

Example:

```
[TEXT]
X = 85
Y = 265
TEXT = This is line 1<N>This is line 2
FONTHEIGHT = 34
PAGE = 1
EDIT = 1
FILLCOLOR = 255, 255, 0
ROTATION = 0
TEXTROTATION = 90
TRANSPARENT = 0
SEMITRANSPARENT = 0
STRIKETHROUGH = 1
```

Related information:

 IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

Annotations and flow layout format documents

Beginning with version 4.0.2 of the viewer, you can view emails and Microsoft Office documents by using either the ViewONE Universal Viewing Module or the ViewONE Office Module. These modules format layouts differently. Therefore, if you view a document with one module and then view it again with the other module, you might see a difference in the layout and pagination.

The difference in layout and pagination occurs because formats such as email or Word are not explicitly defined formats such as TIF or PDF. Instead, these documents use a *flow layout* type format in which the layout and pagination depends on the tool that is used to view the document.

The format differences present the further potential for annotation data to move in terms of its location on the document. This movement can represent a serious security risk for customers who are using annotations to securely redact documents.

For example, if you load an email in Microsoft Outlook, the layout of the email is altered when you adjust the size of the window that contains the email. However, this alteration does not occur if you open a PDF document.

Important: Ensure as much as possible that the same module and version are used to view emails and Microsoft Office documents. Otherwise, the annotations that are added to "flow layout" documents might move. In addition, changes to the modules might affect the manner in which a document is laid out. The ANNOTINFO marker goes some way to address this problem by including the version and module that is used to display the document itself in the annotation data. This ability is an important consideration when it comes to applying annotations as secure redactions on these document types.

Related reference:

"ANNOTINFO marker" on page 18

The ANNOTINFO marker indicates that annotations were saved in V4.0.2 or later of the viewer. These annotations do not load in earlier versions of the viewer that do not recognize the ANNOTINFO marker. If you want to maintain compatibility with an earlier version, for example, for evaluating a new version of the viewer, you can use the **disableAnnotInfo** parameter.

Configuring the streaming module

The ViewONE Document Streaming Server Module processes multipage TIFF files, PDF files, Microsoft Word files, and Microsoft PowerPoint files on the server. The module splits each file into individual pages that are then sent to the client when requested. Document streaming can improve performance for downloading large documents.

About this task

Without the ViewONE Document Streaming Server Module, the client typically requests a document directly from a repository by using a URL. The repository then sends the requested document back to the client for viewing.

The ViewONE Document Streaming Server Module works between the IBM Daeja ViewONE Professional client and your source repository. When a user requests a document in viewer client, the module contacts the original repository and requests the source document. The module then streams the document page-by-page back to the client.

Related tasks:

“Installing IBM Daeja ViewONE interactively” on page 2

You can install IBM Daeja ViewONE on the server by running the installation program. During installation, you are prompted to select the editions and modules for which you are entitled.

Document streaming and prefetching

IBM Daeja ViewONE can prefetch pages. Use this feature when pages of a document can be sent to the viewer separately, as when the ViewONE Document Streaming Server Module is used. IBM Daeja ViewONE reads ahead as a user browses a document.

When you consider whether to use the ViewONE Document Streaming Server Module and if so, with or without prefetching, it is important to consider your actual use case. Consider how users are viewing documents to determine whether the ViewONE Document Streaming Server Module is appropriate for your needs and, if it is, how best to configure it.

The following sample use cases illustrate how the ViewONE Document Streaming Server Module works with prefetching.

Scenario A

Users download large documents that have more than 100 pages and then typically view them sequentially. That is, the users start at page 1, which they need to view as quickly as possible, and then move through subsequent pages 2, 3, 4, 5, and so on. This scenario is the ideal scenario in which to best use the ViewONE Document Streaming Server Module with the **preFetch** parameter.

Typically in this scenario, you would set the **preFetch** value to a value such as 5. Then, while users are looking at the first page, the viewer downloads the subsequent pages in the background ready for the user to view instantly. Without **preFetch** turned on, users might experience a slight delay when they move to the next page because the viewer must reconnect to the server to get each page for display.

If users are in a thumbnail view, the viewer must effectively prefetch each page to display in a thumbnail. Therefore, there is little benefit in setting the **preFetch** value to 5, for example, and thumbnail view to thumbs-left. These two parameter settings do the same thing.

Scenario B

Users download large documents, perhaps on average 50-100 pages per document, and then move to different pages randomly. For example, a user scans quickly the contents on page 1, then views pages 50 - 53, and then goes to the document appendix.

In this scenario, setting the **preFetch** value has a detrimental effect because the viewer downloads extra pages that the user never views. It might be preferable not

to use the ViewONE Document Streaming Server Module in this scenario because the module must to make a connection for every page. If the network speed is sufficient, the user can get the whole document in an acceptable time. The viewer then the entire document is kept in memory so that moving between pages is instantaneous.

Related information:

 prefetchPages parameter

This parameter determines the number of pages that are read-ahead by IBM Daeja ViewONE.

Configuring the servlet for document streaming

You use the *servlet-name.properties* file to configure document streaming for the servlet.

About this task

You can configure the *servlet-name.properties* file to control features such as trace parameters that enable debug output to a specified log file and how the servlet controls its own caching area. You can also use this file to configure how documents are retrieved.

Procedure

To configure document streaming for the servlet:

Edit the */WEB-INF/servlet-name.properties* file to add properties as needed for document streaming. Add *streamer.* to the beginning of the property name to indicate that the property applies only to the document streaming module. For example, set the log file name, enable tracing, and set the document expiration interval for document streaming as follows.

```
streamer.logfile=WEB-INF/viewone.log
streamer.traceNet=true
streamer.documentExpiryTime=1d
```

Configuring the client viewer for document streaming

You must make some minor code changes to the HTML page that loads the client viewer to add the ViewONE Document Streaming Server Module.

About this task

You must ensure that the **mayscript** attribute is present in the applet definition so that the client can correctly obtain the browser's cookies. The **mayscript** attribute enables the applet to use JavaScript to communicate with the browser.

If the **mayscript** attribute is omitted, you see errors in the client's Java console. In addition, the ViewONE Document Streaming Server Module cannot stream documents from a document repository that requires a login.

Procedure

To configure document streaming:

1. Edit the HTML page that loads the viewer applet to add the following line to your applet definition.

```
<applet mayscript="true" name="IBM Daeja ViewONE" ...
```

2. Edit the viewer HTML page or JavaScript to add the ViewONE Document Streaming Server Module URL. For example, add the HTML parameter as follows.

```
<param name="StreamerURL" value=http://localhost:8080/viewone/v1files/streamer>
```

For example, add the JavaScript parameter as follows.

```
ViewONE.setStreamerURL("http://myStreamerHost/Streamer");
```

3. Optional: Edit the viewer HTML page or JavaScript to disable streaming or to enable users to enable streaming as needed. By default, **streamerEnabled** parameter is set to true. To disable streaming, add the HTML parameter as follows.

```
<param name="streamerEnabled" value="false" />
```

For example, add the JavaScript parameter as follows.

```
ViewONE.setStreamerEnabled(false);
```

Related information:

 [streamerURL parameter](#)

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

 [streamerEnabled parameter](#)

This parameter determines whether HTTP requests for documents and images are directed to the ViewONE Document Streaming Server Module.

 [setStreamerURL method](#)

Sets the streamer location dynamically to the specified URL.

Configuring the redaction module

With the ViewONE Permanent Redaction Server Module, users can add permanent redaction markups to any IBM Daeja ViewONE document. Redaction markups remove and block out any part of a page, for example to remove potentially sensitive or confidential information.

About this task

The ViewONE Permanent Redaction Server Module is a JavaServer component that requires the ViewONE Annotations Module.

Unless you are using an existing Enterprise Content Manager system where the ViewONE Permanent Redaction Server Module is already configured and integrated, you must write the burner server code. This code interfaces with the repository where the images and annotation data are stored.

Configuring the servlet for redactions

You use the *servlet-name.properties* file to configure document streaming for the servlet.

About this task

You can configure the *servlet-name.properties* file to control features such as trace parameters that enable debug output to a specified log file and how the servlet controls its own caching area. You can also use this file to configure how documents are retrieved.

Procedure

To configure redactions for the servlet:

Edit the `/WEB-INF/servlet-name.properties` file to add properties as needed for document streaming. Add `redaction.` to the beginning of the property name to indicate that the property applies only to the document streaming module. For example, set the log file name, enable tracing, and enable annotation burning for permanent redactions as follows.

```
redaction.logfile=WEB-INF/viewone.log
redaction.traceNet=true
redaction.traceBurn = true
```

Configuring the client viewer for redactions

You must make some minor code changes to link the ViewONE Permanent Redaction Server Module with the IBM Daeja ViewONE Professional viewer.

Before you begin

You must configure the ViewONE Annotations Module before you configure the redaction module.

About this task

You must ensure that the **mayscript** attribute is present in the applet definition so that the client can correctly obtain the browser's cookies. The **mayscript** attribute enables the applet to use JavaScript to communicate with the browser. If the **mayscript** attribute is omitted, you see errors in the client's Java console.

Procedure

To configure the viewer for redactions:

1. Edit the HTML page that loads the viewer applet to add the following line to your applet definition.

```
<applet mayscript="true" name="IBM Daeja ViewONE" ...
```

2. Edit the viewer HTML page or JavaScript to add the ViewONE Permanent Redaction Server Module URL. For example, add the HTML parameter as follows.

```
<PARAM NAME="annotationBurnURL"
value="http://localhost:8080/viewone/v1files/redaction">
```

For example, add the JavaScript parameter as follows.

```
ViewONE.setAnnotationSaveServlet(
    "http://www.mysite.com/servlet/annotationSave.class?
    userID=12&docID=8791"
);
```

3. If your environment requires permanent redactions that alter the underlying document, enable burnable annotations. For example, add the HTML parameter as follows.

```
<PARAM NAME="annotationDefaults" value="redact {burnable=true}>
```

4. If your environment requires permanent redactions that alter the underlying document, the new file must be directed back to the client. For example, add the HTML parameter as follows.

```
<PARAM NAME="receiveRedacted" value="true">
```

5. If your environment requires permanent redactions that alter the underlying document, define the annotation format and encoding. For example, add the HTML parameters as follows.

```
<PARAM NAME="annotationBurnFormat" value="native">  
<PARAM NAME="annotationEncoding" value="utf8">
```

Upgrading IBM Daeja ViewONE

You can upgrade an existing IBM Daeja ViewONE system to version 4.1.3.

Procedure

To upgrade the server and viewer:

1. Back up the files in your installation directory. The installation program will overwrite the existing files.

Server directory

install_path/server/WEB-INF/lib

install_path/server/WEB-INF

Viewer (client) directory

install_path/client/v1files

2. Back up the files in the web application directory.

Server directory

application_path/server/WEB-INF/lib

application_path/server/WEB-INF

Viewer (client) directory

application_path/client/v1files

3. Run the installation program.
4. Copy the server files from the installation directory to the web application directory.
 - a. Copy the files from *install_path/server/WEB-INF/lib* to *application_path/server/WEB-INF/lib*.
 - b. Copy the files from *install_path/server/WEB-INF* to *application_path/server/WEB-INF*.
5. Copy the client files from the installation directory to the web application directory. Copy the files from *install_path/client/v1files* to *application_path/client/v1files*.
6. IBM FileNet® P8 products or IBM Content Navigator only. Copy the *install_path/client/Annot.xsl* file to the *application_path/WEB-INF/xsl* directory or the *install_path/IDM/Web/FNOpenClient/XML* directory.

Uninstalling IBM Daeja ViewONE

You can remove IBM Daeja ViewONE from your environment.

Procedure

To uninstall IBM Daeja ViewONE:

1. To remove the files from the original installation location, run the uninstallation program:
 - On a Windows system, click **Uninstall a program** in the Control Panel window, select **IBM Daeja ViewONE**, and then click **Uninstall**.
 - On an AIX system, a Linux system, or a Solaris system, run the following command from the command line:

```
viewone-installation-directory/_djuninst/dj_uninstaller
```
2. Stop the web application server and remove the files that you copied to that server.
3. If you are reverting to a previous version of IBM Daeja ViewONE, run the installation program for that version and copy the files to the web application server.
4. Start the web application server.

Configuring IBM Daeja ViewONE

You can configure IBM Daeja ViewONE by using properties to configure the server. You can also use HTML parameters and JavaScript API methods to configure the IBM Daeja ViewONE viewer.

Configuring the server

You can set optional properties to configure and customize the server elements of IBM Daeja ViewONE.

About this task

The `servlet_name.properties` parameter configuration file defines the server properties for the following products and modules:

- IBM Daeja ViewONE Virtual
- IBM Daeja ViewONE Professional
- ViewONE Document Streaming Server Module
- ViewONE Permanent Redaction Server Module

The name of the `servlet_name.properties` must be specified in the `/WEB-INF/web.xml` file that is deployed on the application server. This properties file has the same name as the servlet that you are using, such as `viewone.properties`.

All parameters in this file use the following format:

`name=value`

Use the internet URL address format to specify file names and hyperlink addresses, for example `http://mysite/myimage.tif`. If you provide a relative location, for example, `myimage.tif`, the applet assumes a base address that is the same as the applet location (the codebase).

Except for file names and hyperlink addresses, all parameters are case insensitive.

Procedure

To edit the server configuration file:

1. Open the `WEB-INF/servlet_name.properties` file in a text editor and add the desired parameters.
2. If you are using more than one module, you can set properties for a specific module by adding the appropriate prefix to the parameter name. For example, you can specify logging parameters for each module as follows.

Option	Description
IBM Daeja ViewONE Virtual Viewer	<code>virtual.logfile=WEB-INF/viewone.log</code> <code>virtual.traceNet=true</code>
ViewONE Document Streaming Server Module	<code>streamer.logfile=WEB-INF/viewone.log</code> <code>streamer.traceNet=true</code>

Option	Description
ViewONE Permanent Redaction Server Module	redaction.logfile=WEB-INF/viewone.log redaction.traceNet=true

3. Save your changes.
4. Restart the web application server.

Logging

Use the logging properties to configure the server log files.

logFile parameter

This parameter determines where log messages are stored.

The parameter value can either be a file name or the special identifier <stdout>.

If you specify the special identifier <stdout>, logging is directed to standard out. In this case, the logging typically appears in the web application server's log files. Any other value is assumed to be a file name and the logging information that is output during operation is stored in the specified file. The file name can be an absolute path or a relative path. Enter any backslashes (\) as a double backslash.

If you leave the value blank, then logging messages are output to the <stdout>.

The value of this parameter is not set by default, and therefore no logging occurs.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
logFile=c:\\viewer\\viewer.log
logFile=<stdout>
```

logFileSize parameter

This parameter specifies the maximum size of the log file.

When the log file reaches the specified size, the servlet rolls over the log files. The **logFileCount** parameter determines the number of log files that are retained.

The valid size units for this parameter are b, kb, mb, and gb.

The default value is 5120kb. The minimum value is 512kb.

Requirements

- ViewONE Document Streaming Server Module, V4.0.34 or later
- ViewONE Permanent Redaction Server Module, V4.0.34 or later
- IBM Daeja ViewONE Virtual, V4.0.34 or later

Example

```
logFileSize=30mb
```

Related reference:

“**logFileCount** parameter”

This parameter control the number of log files that the servlet keeps when it rolls over the log files.

logFileCount parameter

This parameter control the number of log files that the servlet keeps when it rolls over the log files.

The servlet rolls over the log files when the current log file reaches the size that is specified for the **logFileSize** parameter. The servlet then renames the log files by appending the numbers in sequence up to the number specified for this parameter minus one. The servlet creates a new log file without a number included in the name. For example, if you set the **logFileCount** to 10 and the log file name is log.txt then the oldest log file is named log_9.txt and the most recent is named log_1.txt.

The default value is 10. The minimum value is 2.

Requirements

- ViewONE Document Streaming Server Module, V4.0.34 or later
- ViewONE Permanent Redaction Server Module, V4.0.34 or later
- IBM Daeja ViewONE Virtual, V4.0.34 or later

Example

```
logFileCount=5
```

Related reference:

“**logFileSize** parameter” on page 58

This parameter specifies the maximum size of the log file.

Cache control

The cache control parameters determine how the server caches data for faster retrieval. You can adjust the cache size, age limits, and so on to improve server performance.

cacheResources parameter

This parameter determines whether IBM Daeja ViewONE files types can be cached in memory.

The following values are valid for this parameter:

Value	Description
all	Files are cached.
none	Files are not cached.

By default, this parameter is set to all.

Requirements

- IBM Daeja ViewONE Virtual, V4.1.2 or later

Example

```
cacheResources=none
```

cacheResourcesMaxItems parameter

This parameter specifies the maximum number of resource files to be cached in memory.

By default, this parameter is set to 20 files.

Requirements

- IBM Daeja ViewONE Virtual, V4.1.2 or later

Example

```
cacheResourcesMaxItems=20
```

cacheResourcesMaxSize parameter

This parameter specifies the maximum size for a file that is to be cached in memory.

The valid size units for this parameter are b, kb, mb, and gb. If you do not specify a unit, the unit is assumed to be megabytes.

By default, this parameter is set to 10 MB.

Requirements

- IBM Daeja ViewONE Virtual, V4.1.2 or later

Example

```
cacheResourcesMaxSize=20
```

cacheRoot parameter

This parameter determines where the servlet keeps its cached document store.

Specify this parameter value in your production environment.

The size controls for the cached document store are determined by the following parameters: **cacheEmptyOnRestart**, **cacheSizeLimit**, and **cacheAgeLimit**

You can specify the path to the cached document store as an absolute path or a relative path. A relative path is relative to the root of the web application (the deployed streamer directory). Enter any backslashes (\) in the path as double backslashes (\\).

Important: Ensure that multiple streamer instances do not share the **cacheRoot** location. For example, if you set up a high availability environment, specify a unique **cacheRoot** location on each node. Take particular care when you specify network locations.

If a value for this parameter is not specified in the properties file, the default value is `WEB-INF/data/servletname`. Where *servletname* is the name of the servlet as defined in the `web.xml`.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Examples

```
cacheRoot=c:\\streamerdemo\\cache  
cacheRoot=cache
```

Related reference:

“**cacheAgeLimit** parameter” on page 62

This parameter determines the maximum age of a document in the cache.

“**cacheEmptyOnRestart** parameter”

This parameter determines whether all the cached documents in the document store are expired when the ViewONE Document Streaming Server Module servlet is restarted.

“**cacheSizeLimit** parameter”

This parameter determines the size of the cache.

cacheEmptyOnRestart parameter

This parameter determines whether all the cached documents in the document store are expired when the ViewONE Document Streaming Server Module servlet is restarted.

Consider setting this parameter to `false` so that the cache is not emptied. If the documents remain in the cache, they are ready for fast retrieval the next time that they are accessed.

Valid values for this parameter are `true` or `false`.

When this parameter is set to `false`, the cache is not emptied when the server is started. The default value is `false`.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
cacheEmptyOnRestart=false
```

cacheCheckTime parameter

This parameter determines how often the system evaluates its cache to see whether it needs expiring. The criteria for whether the cache needs expiring are determined by the **cacheSizeLimit** and **cacheAgeLimit** parameters.

The valid time units for this parameter are `s` (seconds), `m` (minutes), `h` (hours), `d` (days), and `w` (weeks).

The default value is `1h`.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
cacheCheckTime=1h
```

cacheSizeLimit parameter

This parameter determines the size of the cache.

This parameter is a soft limit, which means that the cache might exceed this value between the last cache check and the time that is remaining before the next check. The time is specified by the **cacheCheckTime** parameter. If the cache is found to exceed this size limit, cached document data is deleted, oldest document first until the size is back within the limit.

The valid size units for this parameter are b, kb, mb, and gb.

The default value is 100mb.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
cacheSizeLimit=100mb
```

cacheAgeLimit parameter

This parameter determines the maximum age of a document in the cache.

The parameter is a soft limit, which means that documents in the cache might exceed this value between the last time a cache check was performed and the time that is remaining before the next check is done. The cache check is specified by **cacheCheckTime**. If data for a document inside the cache is found to exceed this age limit, the cached document data is deleted.

The valid time units for this parameter are s (seconds), m (minutes), h (hours), d (days), and w (weeks).

The default value is 1d.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
cacheAgeLimit=1d
```

openDocumentWaitTime parameter

This parameter determines the amount of time to wait before it opens a second instance of a document that is already in the streamer cache.

Deprecated: This parameter is deprecated. The following parameters are recommend instead of this parameter: **correctPageWaitTime**, **correctDocumentWaitTime**, or **anyDocumentWaitTime**.

Use this parameter only where it is likely that a large document that is being pre-cached is being viewed at the same time, and the page to be viewed is not generated yet.

The default value is 0s, which blocks a page that is not generated yet from being viewed, until the pre-cache process is complete.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
openDocumentWaitTime=2s
```

Related reference:

“**correctPageWaitTime** parameter”

This parameter determines the amount of time to wait before a new instance of a document that is already in use on the server and on the correct page is opened.

“**correctDocumentWaitTime** parameter”

This parameter determines the amount of time to wait before a new instance of a document that is already in use on the server is opened.

“**anyDocumentWaitTime** parameter” on page 64

This parameter determines the amount of time to wait before a new processing instance on the server is created. The maximum wait time is controlled by the **poolMaxEntryCount** parameter.

correctPageWaitTime parameter

This parameter determines the amount of time to wait before a new instance of a document that is already in use on the server and on the correct page is opened.

The default value is 60s, which prevents a second version of the page from being opened until either the first version is rendered or 60 seconds elapsed.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
correctPageWaitTime=30s
```

Related reference:

“**openDocumentWaitTime** parameter” on page 62

This parameter determines the amount of time to wait before it opens a second instance of a document that is already in the streamer cache.

correctDocumentWaitTime parameter

This parameter determines the amount of time to wait before a new instance of a document that is already in use on the server is opened.

Use the **correctDocumentWaitTime** parameter to fine-tune performance. For example, if you are running IBM Daeja ViewONE Virtual on a large server with few users, you might want to reduce this value to a few milliseconds. Reducing the value causes more processor and RAM resource to be used to display the document faster.

If the deprecated **openDocumentWaitTime** parameter is specified, the default for the **correctDocumentWaitTime** parameter is half the value of the **openDocumentWaitTime** parameter. If the deprecated **openDocumentWaitTime** parameter is not specified, the default for the **correctDocumentWaitTime** parameter is 2s.

Requirements

- IBM Daeja ViewONE Virtual, any version
- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version

Example

```
correctDocumentWaitTime=2s
```

Related reference:

“**openDocumentWaitTime** parameter” on page 62

This parameter determines the amount of time to wait before it opens a second instance of a document that is already in the streamer cache.

anyDocumentWaitTime parameter

This parameter determines the amount of time to wait before a new processing instance on the server is created. The maximum wait time is controlled by the **poolMaxEntryCount** parameter.

Use the **anyDocumentWaitTime** parameter to fine-tune performance. For example, if you are sharing a server with limited memory, you might want to increase this value of this parameter to minimize RAM usage at the expense of speed.

If the deprecated **openDocumentWaitTime** parameter is specified, the default for the **anyDocumentWaitTime** parameter is the same value of the **openDocumentWaitTime** parameter. If the deprecated **openDocumentWaitTime** parameter is not specified, the default for the **anyDocumentWaitTime** parameter is 4s.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
anyDocumentWaitTime=4s
```

Related reference:

“**openDocumentWaitTime** parameter” on page 62

This parameter determines the amount of time to wait before it opens a second instance of a document that is already in the streamer cache.

“**poolMaxEntryCount** parameter” on page 70

This parameter determines the maximum number of concurrent documents that are held open in memory.

Server sessions

The following parameters are used for general requests in a server session.

docURLPrototype parameter

This parameter is the prototype for the URL that is passed to the HTTP repository store when a user requests a file through the viewer and the ViewONE Document Streaming Server Module.

Tokens can be specified as part of this prototype by using `{` to start the token and `}` to end it. The values for these tokens are taken from the URL that the server is called with.

The simplest method of using this configuration parameter is to use the originalDocURL token name.

Alternatively, a more complex method is available that provides greater control over the URL. You can substitute values from the original request URL. For example, assume that your ViewONE Document Streaming Server Module URL, which is specified in the file name parameter in the HTML page for the IBM Daeja ViewONE Professional client is the following URL:

```
http://localhost:8080/viewone/streamer?path=images&file=test
```

The following **docURLPrototype** is your parameter:

```
http://host/fileserver?type=${path}&name=${file}
```

The resulting request:

```
http://host/fileserver?type=image&name=test
```

The default value is `${originalDocURL}`.

Requirements

- ViewONE Document Streaming Server Module

Example

```
docURLPrototype=${originalDocURL}  
docURLPrototype=http://host/repository/${file}
```

Special tokens:

Sometimes it is necessary to rewrite or alter a document repository URL because the streamer's view of the network is different from the browser.

This difference might be due to a firewall (such as port redirection, IP address remapping) or network configuration (such as DNS resolution differences).

Requirements

- ViewONE Document Streaming Server Module V1.1.172 or later

Example

To obtain a document from within the browser, a sample URL might read:

```
http://docstore.myhost.com/docserver?id=29
```

The same document request for the streamer in an example intranet might read:

```
http://docstore:8080/docserver?id=29
```

A selection of special tokens can be used to cater to this situation. These tokens extract discrete parts of the original document repository URL such as the protocol (for example, http or https) or the port (for example, 80, 8080, 443). The following list is the complete list. Each of the examples listed next to the tokens are based on the previous URL sample.

```
${originalDocURL.protocol}, for example, http  
${originalDocURL.host}, for example, docstore.myhost.com  
${originalDocURL.port}, for example, 80  
${originalDocURL.path}, for example, /docserver  
${originalDocURL.query}, for example, id=29  
${originalDocURL.authority}, for example, docstore.myhost.com
```

Note: If no port is supplied, 80 is the default.

For the previous sample, remap both the host name and the port. This action can be achieved with the following code. Do not insert line breaks in the configuration file.

```
docURLPrototype=${originalDocURL.protocol}://docstore:8080${originalDocURL.path}?  
    ${originalDocURL.query}
```

This value for **docURLPrototype** takes any host name from the incoming document repository URL and replaces it with `docstore:8080`.

docIdPrototype parameter

The ViewONE Document Streaming Server Module needs to identify every document internally. This parameter provides a method for assigning a unique identifier to documents.

If you are using a value of `${originalDocURL}` for **docURLPrototype**, it represents a unique identifier for the document and as a result the **docIdPrototype** can be set to the same value. In most cases, this setting is adequate. If you access documents by using a proxy that might result in differing URLs for the same document, then consider the more advanced usage of **docIdPrototype**.

The default value is `${originalDocURL}`.

Requirements

- ViewONE Document Streaming Server Module, any version

Example

```
docIdPrototype=${originalDocURL}
```

The unique identifier can alternatively be built up directly from your original ViewONE Document Streaming Server Module URL in a similar fashion to the **docURLPrototype**. Tokens can be specified as part of this prototype by using `{` to start the token and `}` to end it. The values for these tokens are taken from the URL that the ViewONE Document Streaming Server Module is called with. For example, assume that the following ViewONE Document Streaming Server Module URL is specified in the file name parameter in the HTML page for the IBM Daeja ViewONE Professional client:

```
http://hostname/streamer/streamer?docId=89&version=20
```

And the following **docIdPrototype** is yours:

```
docIdPrototype=${docId}-${version}
```

Then, the resulting unique identifier is 89-20.

sessionAuthenticationTimeout parameter

After the time that is specified by this parameter, the user is reauthenticated with the document repository. After the timeout is reached through inactivity, the document information is no longer served from the cache. A new query to the repository is made to ensure that access is still available.

This action is to cater for cases where a user is already viewing a document, but access to it is revoked in the middle of viewing. The user is not able to view any subsequent parts of the document after their access is revoked.

The valid time units for this parameter are s (seconds), m (minutes), h (hours), d (days), and w (weeks).

The default value for this parameter is 1 minute.

If you set this parameter to -1, session caching is disabled which causes the server to check the original document on every request. Setting this parameter to -1 is not recommended for IBM Daeja ViewONE Virtual.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
sessionAuthenticationTimeout=60s
```

sessionCookieList parameter

This parameter specifies a comma-separated list of cookies that must be present on the request from the client.

To set up cookie forwarding within IBM Daeja ViewONE, use the **cookieAllowedDomainList** HTML parameter.

If your document repository requires a user to be logged in before they can retrieve documents, then the value of this parameter is important. It indicates to the system which cookies to forward to the document repository for authentication purposes. It consists of a comma-separated list of cookie names that are used by the document repository to identify individual logged in users. An asterisk (*) wildcard is acceptable. If this parameter is specified as in the following example:

```
sessionCookieList=ASPSESSION*
```

Then, the following named cookies (for example) would be forwarded to the repository:

```
ASPSESSION1293
```

```
ASPSESSIONID
```

If this value is omitted, the default value is *JSESSIONID*.

Requirements

- IBM Daeja ViewONE Virtual, any version
- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module

Example

```
sessionCookieList=JSESSIONID
```

Related reference:

“**cookieAllowedDomainList** parameter” on page 257

This parameter defines a comma-separated list of hosts that the client IBM Daeja ViewONE Professional is allowed to send cookies to from its containing HTML page. This parameter is required when the source document repository uses cookies to identify logged in users.

signInResource parameter

This parameter specifies a string for checking whether a user logged out from a document repository.

If a user makes a request for a document and the session expired, then the web server usually returns a redirect status (302 - moved temporarily). In this case, extra logic is used to check whether the user was logged out by looking at the Location: header field for the string that is specified by **signInResource**. If this string is found, then the ViewONE Document Streaming Server Module knows that user was logged out.

If this parameter is unspecified, then no attempt is made to see whether a user was logged out by a document repository. If the document repository does not support sessions or logins, then this value can be left unspecified.

The default value is unspecified.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
signInResource=SignIn.jsp
```

serverSupportsHeadRequestMethod parameter

This parameter determines whether the server supports the HTTP HEAD method for collecting information about a document without retrieving the entire document over the network.

When set to true, this parameter uses the HTTP HEAD method for collecting information about a document without retrieving the entire document over the network. Support of this feature allows for more efficient protocol dialog. If the document repository server does not support the HEAD method or it does not return meaningful values for HTTP header fields such as Content-Length, Content-Type, or Last-Modified, set this parameter to false.

The default value is true.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
serverSupportsHeadRequestMethod=true
```

modifiedDocumentMinimumCheckTime parameter

This parameter determines how often the system checks the document repository to see whether a particular document was modified since it was last accessed.

The system checks no more often than the time that is specified in this parameter. It is more efficient at a network level to avoid requerying the document repository every time a document is accessed. If you want the repository to be queried with

every document access, set the value to 0. Setting the value to 0 is not recommended for IBM Daeja ViewONE Virtual.

The valid time units for this parameter are s (seconds), m (minutes), h (hours), d (days), and w (weeks).

The default value is 120s.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
modifiedDocumentMinimumCheckTime=5m  
modifiedDocumentMinimumCheckTime=1d
```

modifiedDocumentCheckMethod parameter

This parameter determines the whether the server checks the modified date or the unique id to determine if a cached document is out of date.

When documents are cached, the server needs to determine if a document has been modified since it was cached to prevent sending out of date information to a user.

Most document repositories provide an HTTP header that is called Last-Modified to indicate when a document was last changed. When this parameter is set to lastmodified, the server checks the Last-Modified value in the header. If you are serving files from an HTTP server, this option is probably the correct option.

Some document repositories generate a different document identifier when a document is modified. In these situations, the modified date becomes irrelevant. When this parameter is set to uniqueid, the server checks the document identifier instead of the modification date. It is important when you use this mode that the URL for one document version is not the same as the URL for a different version of the same document.

The default value is lastmodified.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Examples

```
modifiedDocumentCheckMethod=lastmodified  
modifiedDocumentCheckMethod=uniqueid
```

documentExpiryTime parameter

This parameter determines how long a document is kept in the cache after it is accessed.

The internal counter is reset for every access, and after no more accesses occurred after the time specified in this parameter, the document is closed. The ideal value for this parameter is the length of time that the average user browses pages in a single document.

The valid time units for this parameter are s (seconds), m (minutes), and h (hours).

If a value for this parameter is not specified in the properties file, the default value is 120s.

Requirements

- ViewONE Document Streaming Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
documentExpiryTime=120s
```

poolInitialEntryCount parameter

This parameter determines the number of concurrent document holders that are created when the system servlet is initialized.

This value grows automatically up to the limit of **poolMaxEntryCount**. It is provided as a fine-tuning option and can be safely left at its default value.

The default value is 1.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
poolInitialEntryCount=1
```

poolMaxEntryCount parameter

This parameter determines the maximum number of concurrent documents that are held open in memory.

A low value (for example 1) results in low server memory usage, but can cause bottlenecks if you have more than one user that is viewing separate documents. A large number might be limited by the amount of available memory on your web application server, but it might also improve performance for the situation when many users are viewing documents.

The default value is 1000000.

Requirements

- ViewONE Document Streaming Server Module, any version
- ViewONE Permanent Redaction Server Module, any version
- IBM Daeja ViewONE Virtual, any version

Example

```
poolMaxEntryCount=10
```

Configuring the viewer by using HTML parameters

You can use HTML parameters to configure and customize the viewer.

About this task

Use the following format for adding a parameter to the HTML file that loads the viewer:

```
<param name="name1" value="value1" />
```

Replace *name1* with the parameter name and *value1* with the parameter value.

Use the internet URL address format to specify file names and hyperlink addresses, for example `http://mysite/myimage.tif`. If you provide a relative location, for example, `myimage.tif`, the viewer assumes a base address that is the same as the viewer location (the codebase).

Except for file names and hyperlink addresses, all parameters are not case-sensitive.

Procedure

1. Use a text editor to open the HTML file that loads the viewer.
2. Add the parameters and values.
3. Save your changes.
4. Restart the web application server.

What to do next

You can check the parameters that are enabled in the viewer by pressing Shift+T. This command shows a list of the parameters that are processed by the viewer.

Related tasks:

“Configuring the viewer by using JavaScript API methods” on page 264
The IBM Daeja ViewONE JavaScript API offers an alternative mechanism to configure and control IBM Daeja ViewONE Professional and IBM Daeja ViewONE Virtual.

Checking your parameters

If you are unsure whether IBM Daeja ViewONE processed any of your parameters correctly, or are having problems with a particular parameter, you can check the actual list of processed parameters.

About this task

Let this check be your first check after you add a parameter. If you find that the parameter fails to do what is expected, always check that it was processed correctly before you take the next step.

Procedure

To access the list of processed parameters in the viewer

1. Start IBM Daeja ViewONE by going to your web page that contains the viewer.
2. Click in the center of the viewer and then press Shift+T.
3. Review the list. If a parameter that you specified in your HTML does not appear in the list that is provided by this check, then this check indicates that there is either a syntax error in the inclusion of the parameter name in the HTML, or there is a deployment issue.

Advanced Cache Manager parameters

Advanced Cache Manager (ACM) parameters can provide significant performance benefits because the applet archive files are considerably smaller and the browser processes the files quicker. IBM Daeja ViewONE can handle all cache processing more effectively for applet and module extensions.

About this task

The ACM parameters apply only to IBM Daeja ViewONE Professional.

The parameters configure the following aspects of the cache:

- Where the cached document store is kept
- The size of the cache
- The maximum age of a document in the cache
- When cached documents expire
- The amount of time to wait before a new instance of a page or document that is already in use on the server is opened
- The amount of time to wait before a new processing instance on the server is created

ACMDownloadPrompt parameter

This parameter determines whether a warning prompt is displayed when the predicted time to download a module exceeds a preset limit.

By default, IBM Daeja ViewONE monitors the network performance as a user downloads a module such as ViewONE PDF Module. If the **ACMDownloadPrompt** parameter is set to true, the user is warned if the predicted time exceeds the preset limit.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMDownloadPrompt" value="false" />
<param name="ACMDownloadPrompt" value="true" />
```

ACMDownloadPromptTrigger parameter

This parameter determines the time interval at which a user is prompted for downloads. This parameter sets the time interval in seconds before the user gets prompted.

The prompt that is displayed is defined by the **ACMDownloadPrompt** parameter.

The default value for this parameter is 15 seconds.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMDownloadPromptTrigger" value="10" />
```

ACMPreInstall parameter

This parameter determines whether the viewer is to download and cache all the required files to the client workstation.

If set to true, this parameter causes the viewer to download and cache all the required files to the client workstation. The viewer displays progress bars as this process is carried out.

After the files are cached, the web page that is identified in the **ACMRedirect** parameter is displayed.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMPreInstall" value="true" />
```

Related reference:

“**ACMRedirect** parameter”

This parameter opens a specified web page after IBM Daeja ViewONE is reinstalled or removed. You might use this parameter to confirm that a reinstallation or removal occurred.

ACMRedirect parameter

This parameter opens a specified web page after IBM Daeja ViewONE is reinstalled or removed. You might use this parameter to confirm that a reinstallation or removal occurred.

Use this parameter only with the following parameters: **ACMPreInstall**, **ACMReInstall**, and **ACMRemove**.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMRedirect" value="http://mysite/mywebpage.html" />
```

Related reference:

“**ACMPreInstall** parameter”

This parameter determines whether the viewer is to download and cache all the required files to the client workstation.

“**ACMReInstall** parameter” on page 74

This parameter determines whether IBM Daeja ViewONE is to update its cache regardless of the version that is already cached.

“**ACMRemove** parameter” on page 75

This parameter determines whether Advanced Cache Manager is to remove the current cached version of IBM Daeja ViewONE from users' workstations.

ACMRedirectTarget parameter

This parameter identifies the window to which the browser is to be redirected.

You can specify the following values for this parameter:

_blank

Redirects the browser to open in a new window.

_self Redirects the browser to open in the same window or frame that contains IBM Daeja ViewONE.

_top Redirects the browser to open in the top-level frame of the IBM Daeja ViewONE window (if **_self** frames are not used).

_parent

Redirects the browser to open in the parent frame of IBM Daeja ViewONE (if **_self** frames are not used).

name Redirects the browser to open in the specified window. If this window does not exist, IBM Daeja ViewONE creates a top-level window with this name.

Important: This parameter must be used with the **ACMRedirect** parameter.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMRedirectTarget" value="_blank" />
```

Related reference:

“**ACMRedirect** parameter” on page 73

This parameter opens a specified web page after IBM Daeja ViewONE is reinstalled or removed. You might use this parameter to confirm that a reinstallation or removal occurred.

ACMReInstall parameter

This parameter determines whether IBM Daeja ViewONE is to update its cache regardless of the version that is already cached.

If set to true, this parameter forces IBM Daeja ViewONE to update its cache regardless of the version that is already cached. The update forces the version on the server to be cached on the users' workstation.

After the reinstallation, IBM Daeja ViewONE does not run. To run IBM Daeja ViewONE or perform other functions after a reinstallation, you must use this parameter with the **ACMRedirect** parameter.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMReInstall" value="true" />
```

Related reference:

“**ACMRedirect** parameter” on page 73

This parameter opens a specified web page after IBM Daeja ViewONE is reinstalled or removed. You might use this parameter to confirm that a reinstallation or removal occurred.

ACMRemove parameter

This parameter determines whether Advanced Cache Manager is to remove the current cached version of IBM Daeja ViewONE from users' workstations.

If this parameter is set to true, the current cached version of IBM Daeja ViewONE is removed from the users' workstations.

After the cached version is removed, IBM Daeja ViewONE does not run. If you want to run IBM Daeja ViewONE or perform other functions following a reinstall, you must use this parameter with the **ACMRedirect** parameter.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMRemove" value="true" />
```

Related reference:

“**ACMRedirect** parameter” on page 73

This parameter opens a specified web page after IBM Daeja ViewONE is reinstalled or removed. You might use this parameter to confirm that a reinstallation or removal occurred.

ACMUpdate parameter

This parameter determines whether the automatic update feature is enabled. The automatic update feature ensures that the version of IBM Daeja ViewONE Professional that is cached on a user's system is the same version that is on the server.

If set to false, this parameter disables the automatic update feature of the Advanced Cache Manager.

You might use the **ACMUpdate** parameter to help schedule update downloads while you control network bandwidth usage. For example, if you have many users, you might want to schedule only a few users to update per day. In this case, you can set the **ACMUpdate** parameter to false for all users except those users that you want to update.

The default value for this parameter is true, which enables automatic update for all users.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="ACMUpdate" value="false" />
```

cachePath parameter

This parameter designates a specific folder for the IBM Daeja ViewONE working area.

The default value is the designated temporary folder as defined by the client workstation environment variable. If this area is inaccessible, the default becomes the working folder of the JVM that is used to run IBM Daeja ViewONE.

Important:

- You must use a forward-slash (/) for directory separators. For example, `c:/temp` is a valid directory, but `c:\temp` is not a valid directory.
- This location must be a unique location for each user. The images and user preference file is also cached in this working folder. If the location is not unique to the user, users share these files, which can cause unforeseen problems.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="cachePath" value="c:/mycache" />
```

useWindowsTempPath parameter

When the `cachePath` HTML parameter is not specified, the Java `java.io.tmpdir` system property is used to determine where the IBM Daeja ViewONE working folder is located.

This parameter is only recognized on Windows operating systems.

When this parameter is set to `false`, IBM Daeja ViewONE does not use the Java `java.io.tmpdir` system property to choose where to put its file. Instead, it uses the Java `user.home` system property directly.

The default for this parameter is `true`.

Requirements

- IBM Daeja ViewONE Standard V3.0.346 or later
- IBM Daeja ViewONE Professional V1.0.346 or later
- No additional modules

Example

```
<param name="useWindowsTempPath" value="false" />
```

Annotations options

You can use the annotations parameters to determine annotation options such as availability, color, deletion, error message display, text, redaction, and save options.

Angles and rulers

You can use the angle and ruler parameters specify the decimal places, units, and scale that are displayed by the ruler and angle annotations.

angleUnits parameter:

This parameter determines the units that the angle annotation displays.

You can set this parameter to the following values:

- Degrees (centimeters)
- Rad (radians)
- Dms (degrees, minutes, and seconds)

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="angleUnits" value="rad" />
```

rulerScale parameter:

This parameter specifies the scale that is applied to the ruler units.

The scale is basically a multiplier, so a scale of 2.0 means the numbers that the ruler displays are multiplied by 2. The default value is 1.0.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="rulerScale" value="2.0" />
```

rulerUnits parameter:

This parameter specifies the units that the ruler annotation displays.

The optional values are:

- cm
- mm
- inches
- inches and cm

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="rulerUnits" value="cm" />
```

unitDecimalPlaces parameter:

This parameter specifies the number of decimal places the ruler and angle annotations display.

The default is 3 decimal places.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="unitDecimalPlaces" value="5" />
```

Availability

You can use the availability parameters determine whether annotations features are available.

annotate parameter:

This parameter determines whether all annotation features are available in the viewer.

If this parameter is set to false, all annotation features are disabled in the viewer. Users cannot view or edit annotations; annotations are not retrieved.

The default value of this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotate" value="true" />
```

annotateEdit parameter:

This parameter determines whether the **Annotations** toolbar is available to users.

If this parameter is set to true, users can use the **Annotations** toolbar to edit editable annotations and add new annotations.

If this parameter is set to false, the **Annotations** toolbar is removed from the user interface. Users can view annotations, but not edit or add annotations.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotateEdit" value="true" />
```

annotationFile parameter:

This parameter specifies the annotations definition file that contains the list of the annotations to display and their associated parameters.

Typically, you do not specify the location of a physical file for the parameter value. Instead, you specify the location of the server-side object that streams an annotations file to the viewer. Examples of a server-side object include a Common Gateway Interface (CGI) script, an Active Server Page (ASP), or an executable (EXE) file.

You can add parameters, such as a user ID or a document ID, to the query string part of the URL to provide user-specific annotations or document-specific annotations. If you specify a server-side object to stream the file, this object must handle any parameters that you include.

The annotations file must conform to the IBM Daeja ViewONE annotations file format.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationFile" value="http://mysite/myscript.pl?
userID=12&docID=8791" />
```

Related information:

 [Creating an annotation template](#)

Color

You can use the color parameters determine the default color for various annotation types.

annotationDefaultFillColor<N> parameter:

This parameter determines the default fill color for an annotation type.

The value for this property uses the following format:

```
value = "type: R, G, B"
```

The *type* variable indicates the type of annotations to which the default fill color applies. Use one of the following values for this variable:

- all
- arrow
- freehand
- highlight
- highlightPoly
- line
- note
- openPoly
- oval
- poly
- rectangle
- redact
- redactPoly
- stamp
- text

The *R*, *G*, and *B* variables specify the red, green, and blue intensities.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

The following examples change the default fill color of a highlight annotation to red and the default fill color of an oval annotation to blue.

```
<param name="annotationDefaultFillColor1" value="highlight: 255, 0, 0" />
<param name="annotationDefaultFillColor2" value="oval: 0, 0, 255" />
```

annotationDefaultLineColor<*N*> parameter:

This parameter determines the default line color for an annotation type.

The value for this property uses the following format:

```
value = "type: R, G, B"
```

The *type* variable indicates the type of annotations to which the default line color applies. Use one of the following values for this variable:

- all
- arrow
- freehand
- highlight
- highlightPoly
- line
- note
- openPoly
- oval
- poly

- rectangle
- redact
- redactPoly
- stamp
- text

To set the default line color for a specific stamp annotation, set the *type* variable to the menu name for that stamp annotation.

The *R*, *G*, and *B* variables specify the red, green, and blue intensities.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

The following example sets the default line color for line annotations to red and the default line color of arrow annotations to blue:

```
<param name="annotationDefaultLineColor1" value="line: 255, 0, 0" />
<param name="annotationDefaultLineColor2" value="arrow: 0, 0, 255" />
```

The following example sets the default line color of the standard IBM Daeja ViewONE stamp *approved* to red:

```
<param name="annotationDefaultLineColor1" value="approved: 255, 0, 0" />
```

annotationHighlightColor parameter:

This parameter determines the default color for highlight annotations.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 255, 255, 0, which is yellow.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationHighlightColor" value="0, 255, 0" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

annotationLimitedColors parameter:

This parameter determines the colors that users can choose for annotations.

Setting this parameter to true has the following effect:

- Users cannot select white or near-white colors from the color chooser dialog boxes.
- White or near-white colors cannot be used in the annotation definition files that the viewer reads or writes. If a white or near-white color is found in a definitions file, IBM Daeja ViewONE automatically darkens the color.
- Users cannot create redaction annotations with a white or near white color.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationLimitedColors" value="true" />
```

annotationLineColor parameter:

This parameter determines the default color for line annotations.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 255, 0, 0, which is red.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationLineColor" value="0, 0, 0" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

annotationNoteColor parameter:

This parameter determines the default color for note annotations.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 255, 255, 153, which is pale yellow.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationNoteColor" value="255, 175, 175" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

annotationRedactColor parameter:

This parameter determines the default color for redaction annotations.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 0, 0, 0, which is black.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationRedactColor" value="255, 0, 0" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

Cursor

You can use the cursor parameters determine the actions of the cursor in annotations drawing mode and across page changes.

annotationsStickyAcrossPages parameter:

This parameter determines whether annotations sticky mode persists across page changes.

This parameter applies only when sticky annotations mode is enabled by using the **annotationsSticky** HTML parameter or by right-clicking when an annotation is started.

When this parameter is set to true, annotations sticky mode persists across page changes. The user does not need to select the annotation again after moving to a new page. The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.434 or later
- IBM Daeja ViewONE Professional V1.0.434 or later
- ViewONE Annotations Module

Example

```
<param name="annotationsStickyAcrossPages" value="false" />
```

Related reference:

“**annotationsSticky** parameter”

This parameter determines whether the cursor remains in annotations drawing mode.

annotationsSticky parameter:

This parameter determines whether the cursor remains in annotations drawing mode.

When this parameter is set to true, the cursor remains in annotations drawing mode and allows the reuse of a selected annotation type without having to reselect it from the annotations toolbar.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationsSticky" value="true" />
```

Dates

You can use the date parameters determine the locale that is to be used for dates and times. The date format that is used in the **annotationStamp**<N> parameter and **annotationDateStyle** parameter is also determined.

annotationDateStyle parameter:

This parameter determines the date format that is used for *date* and *dateonly* values in the **annotationStamp**<N> parameter. The **annotationDateStyle** parameter also alters the date format that is displayed in the sticky note dialog box.

You can specify the data format as either long or short. Although the exact format is determined by the user's regional settings, the following examples illustrate the difference in these formats:

- *long*: 24 September, 2002, 13:00:01, GMT+01:00
- *short*: 24-Sep-02, 13:00:01, GMT+01:00

The default value for this parameter is long.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- ViewONE Annotations Module

Example

```
<param name="annotationDateStyle" value="short" />
```

Related reference:

“**annotationStamp**<N> parameter” on page 114

This parameter specifies the stamps that are used in place of IBM Daeja ViewONE default stamps. The stamp that is specified can be either text or image based.

annotationDateTimeOutputLocale parameter:

This parameter determines the locale that is to be used for dates and times.

The following table lists the codes that you can use as values for this parameter:

Code	Language	Code	Language
ar	Arabic	ko	Korean
bg	Bulgarian	nl	Dutch
ca	Catalan	no	Norwegian
cs	Czech	pl	Polish
da	Danish	pt	Portuguese
de	German	pt_br	Portuguese Brazil
en	English	ro	Romanian
es	Spanish	ru	Russian
fi	Finnish	sk	Slovak
fr	French	sv	Swedish
hr	Croatian	th	Thai
hu	Hungarian	tr	Turkish
it	Italian	zh_cn	Simplified Chinese
iw	Hebrew	zh_tw	Traditional Chinese
ja	Japanese		

The default value for this parameter is the locale set for the user's workstation.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- ViewONE Annotations Module

Example

```
<param name="annotationDateTimeOutputLocale" value="en" />
```

Deletion

You can use the deletion parameters determine whether confirmation dialog boxes are opened when all annotations are deleted from the current page or current document.

annDeleteAllDocConfirm parameter:

This parameter determines whether a confirmation dialog box is opened when a user selects the menu item to delete all annotations from the current document from the **Delete Annotation** button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V3.1.168 or later
- ViewONE Annotations Module

Example

```
<param name="annDeleteAllDocConfirm" value="false" />
```

Related reference:

“**annDeleteToolBarButton** parameter” on page 127

This parameter determines whether the annotation toolbar contains a button to display a menu for deleting annotations.

annDeleteAllPageConfirm parameter:

This parameter determines whether a confirmation dialog box opens when a user selects the menu item to delete all annotations on the current page.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V3.1.168 or later
- ViewONE Annotations Module

Example

```
<param name="annDeleteAllPageConfirm" value="false" />
```

Related reference:

“**annDeleteToolBarButton** parameter” on page 127

This parameter determines whether the annotation toolbar contains a button to display a menu for deleting annotations.

Encoding

You can use the encoding parameters determine whether the viewer uses a text encoding type that allows multiple-byte characters to be used, or whether the viewer converts characters to their HTML-encoded counterparts.

annotationEncoding parameter:

This parameter determines whether IBM Daeja ViewONE uses a text encoding type that allows multiple-byte characters to be used.

If a workstation fully supports a specific multiple-byte character language such as Chinese, then this parameter is not required. However, set this parameter if a workstation only partially supports the multiple-byte character language.

To enable this parameter, set the value to UTF-8.

Important: IBM Daeja ViewONE uses this encoding to send annotations. Therefore, you must ensure that the server object can receive and preserve the encoding in the save and retrieval phases.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationEncoding" value="UTF8" />
```

htmlEncoding parameter:

This parameter determines whether the viewer converts the following characters to their HTML-encoded counterparts: less than symbol (<), greater than symbol (>), ampersand (&). You might use this parameter if the server-side object that handles the annotations data stream does not accept data that is not HTML encoded data for security reasons.

If you set this parameter to true, the viewer parses the annotations data when the annotations are saved. The viewer converts these characters as follows.

Character	HTML encoding
Less than symbol (<)	<
Greater than symbol (>)	>
Ampersand (&)	&

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="htmlEncoding" value="true" />
```

Error messages

You can use the error messages parameters to determine whether error messages are displayed.

About this task

annotationSuppressEmptyMessages parameter:

This parameter determines whether an error message is displayed when the supplied annotations definition file is empty and there is no [EMPTY] marker. This parameter is for compatibility with earlier IBM Daeja ViewONE versions where empty annotations files were ignored.

When this parameter is set to true, the viewer does not display an error message if the supplied annotations definition file is empty and there is no [EMPTY] marker. The image is displayed.

The default value is false, which always displays the error message and stops viewing.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSuppressEmptyMessages" value="true" />
```

Hyperlinks

You can use the hyperlink parameters determine the default base location for web hyperlinks and specifies a list of optional targets for web hyperlinks to use.

About this task

annotationHyperlinkWeb parameter:

This parameter determines the default base location for web hyperlinks. If you define this parameter, relative hyperlinks are then relative to the specified URL.

This parameter makes it easier for users to specify hyperlinks for their annotations. For example, assume that you set this parameter to `http://mysite`. If a user then specifies `page.html` as a hyperlink in an annotation, the hyperlink path resolves to `http://mysite/page.html`.

The default value for this parameter is the codebase path.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationHyperlinkWeb" value="http://mysite" />
```

annotationTarget<N> parameter:

This parameter specifies a list of optional targets for web hyperlinks to use in place of IBM Daeja ViewONE default targets.

The value that is specified follows the format:

```
<target><menu label text>
```

The following table lists the values for the target option.

Target	Description of target
_self	Current® web page
_blank	New Window
_parent	Parent frame
_top	Top most parent frame
Any frame window name	This window must exist for a successful link

This parameter is useful for specifying common windows in a web application, for example, index terms, contents, or help.

If this parameter is used, then all hyperlink targets must be specified. For example, you cannot have annotationTarget2, but not annotationTarget1.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTarget1" value="<_self><This window>" />
<param name="annotationTarget2" value="<_blank><New Window>" />
<param name="annotationTarget3" value="<banner><Banner frame>" />
```

Related reference:

“IBM Daeja ViewONE Default Stamps” on page 263

The viewer's default stamps are listed in the following table. The stamps are used on documents that need to be tagged with a predefined set of annotations.

Kodak WANG file format

The viewer supports the now largely redundant Kodak WANG annotation file format for TIF images where the annotations are embedded in the actual image file itself.

The WANG annotations support parameters:

- Read and write local annotations and annotations from the server.
- Specify a server-side component for saving the image data by using a post request or by using a servlet.
- Determine whether an error message is displayed if annotations are saved as WANG and unsupported types of annotation are found.
- Set the save image function to use a parameter; otherwise, the function writes a local copy of the image by using the Save dialog box.
- Configure the displayed border around text annotations that are loaded from WANG TIF files.

The supported annotation types are detailed in *WANG Compliant Annotation Types*. The HTML parameters that allow the reading in and writing out of these particular types of annotations are detailed.

imageSavePost parameter:

This parameter specifies a server component for saving the image data by using a post request.

This server component is called when the document is saved only if you also set the **saveButtonToServer** parameter to true.

If you set the **webAnnotationWriteMode** parameter or **localAnnotationWriteMode** parameter to WANG, the specified server component is also called when annotations are saved. The annotation data is also sent to the server component.

Annotation data and image data are sent separately to the server component. The viewer automatically appends "type=IMAGE" or "type=ANNOTATION" to the specified servlet URL to indicate the data type being sent.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageSavePost" value="http://myserver/save.asp" />
```

Related reference:

“**saveButtonToServer** parameter” on page 92

This parameter sets the save image function to use the **imageSaveServlet** parameter or the **imageSavePost** parameter; otherwise the function writes a local copy of the image by using the Save dialog box.

“**webAnnotationWriteMode** parameter” on page 94

The annotation writing mode is for images from the server, that is, images that are specified as a URL.

“**localAnnotationWriteMode** parameter” on page 91

This parameter determines whether the viewer saves annotations as TIFF files that use the WANG format.

imageSaveServlet parameter:

This parameter specifies a servlet that is used to save image data to a server component.

This servlet is called when the document is saved only if you also set the **saveButtonToServer** parameter to true.

If you set the **webAnnotationWriteMode** parameter or **localAnnotationWriteMode** parameter to WANG, the specified servlet is also called when annotations are saved. The annotation data is also sent to the server component.

Annotation data and image data are sent separately to the server component. The viewer automatically appends "type=IMAGE" or "type=ANNOTATION" to the specified servlet URL to indicate the data type being sent.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageSaveServlet" value="http://myserver/annSave" />
```

Related reference:

“**saveButtonToServer** parameter” on page 92

This parameter sets the save image function to use the **imageSaveServlet** parameter or the **imageSavePost** parameter; otherwise the function writes a local copy of the image by using the Save dialog box.

“**webAnnotationWriteMode** parameter” on page 94

The annotation writing mode is for images from the server, that is, images that are specified as a URL.

“**localAnnotationWriteMode** parameter”

This parameter determines whether the viewer saves annotations as TIFF files that use the WANG format.

localAnnotationReadMode parameter:

This parameter determines whether the viewer reads annotations that are saved as TIFF files that use the WANG format.

This parameter can be specified by using an integer or string. The integers and strings are in the following table.

Integer	String	Reading mode
0	ANT	Reads only those annotations that are saved in .ant files that use the typical IBM Daeja ViewONE format.
1	WANG	Reads only those annotations that are saved in .tif files that use the WANG format.
3	ANTnWANG	Reads annotations that are saved in .ant files and in .tif files.

The default value is 0 or ANT.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="localAnnotationReadMode" value="ANTnWANG" />
```

localAnnotationWriteMode parameter:

This parameter determines whether the viewer saves annotations as TIFF files that use the WANG format.

This parameter can be specified by using an integer or string. The integers and strings are in the following table.

Integer	String	Writing mode
0	ANT	Annotations are saved as .ant files that use the typical IBM Daeja ViewONE format.
1	WANG	Annotations are saved as .tif files that use the WANG format. The image can be saved by using a server component if required.

The default value for this parameter is 0 (ANT).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="localAnnotationWriteMode" value="1" />
```

saveButtonToServer parameter:

This parameter sets the save image function to use the **imageSaveServlet** parameter or the **imageSavePost** parameter; otherwise the function writes a local copy of the image by using the Save dialog box.

When this parameter is set to false, and you are using WANG annotations, the **Save annotations** command writes to the server by using the save parameters. If the parameter is set to null, the **Save annotations** command writes to the local file system. The default value is null.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="saveButtonToServer" value="true" />
```

Related reference:

“**imageSaveServlet** parameter” on page 90

This parameter specifies a servlet that is used to save image data to a server component.

“**imageSavePost** parameter” on page 89

This parameter specifies a server component for saving the image data by using a post request.

unsupportedWangError parameter:

This parameter tells IBM Daeja ViewONE whether to put up an error message if annotations are saved as WANG and unsupported types of annotation are found.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="unsupportedWangError" value="false" />
```

wangTextBorder parameter:

When this parameter is set to true, a border is displayed around text annotations that are loaded from WANG TIF files.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.754 or later
- IBM Daeja ViewONE Professional V1.0.754 or later
- ViewONE Annotations Module

Example

```
<param name="wangTextBorder" value="true" />
```

webAnnotationReadMode parameter:

The annotation reading mode is for images from the server, that is, images that are specified as a URL.

This parameter can be specified by using an integer or string. The integers and strings are in the following table.

Integer	String	Reading mode
0	<i>ANT</i>	Read IBM Daeja ViewONE annotations as normal. (default)
1	<i>WANG</i>	Read WANG annotations only from tif files
3	<i>ANTnWANG</i>	Read WANG annotations only from tif files and also normal IBM Daeja ViewONE annotations from Ant file.

The default value for this parameter is 0 (*ANT*).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="webAnnotationReadMode" value="1" />
```

webAnnotationWriteMode parameter:

The annotation writing mode is for images from the server, that is, images that are specified as a URL.

This parameter can be specified by using an integer or string. The integers and strings are in the following table.

Integer	String	Writing mode
0	<i>ANT</i>	Write IBM Daeja ViewONE annotations as normal. (default)
1	<i>WANG</i>	Write WANG annotations into the tif image. One of the save image parameters must be set so the new image, with WANG annotations, can be sent back to the server.

The default value for this parameter is 0 (*ANT*).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="webAnnotationWriteMode" value="1" />
```

Module version and migration

Use the annotation migration support parameters to configure version checking, determine when the user is notified of annotation changes, and how the viewer handles misplaced annotations.

disableAnnotInfo parameter:

This parameter determines whether version checking is disabled.

If you set this parameter to true, the viewer disables all version checking and the **ANNOTINFO** marker is ignored. This marker is not created or deleted with subsequent annotation saves.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="disableAnnotInfo" value="true" />
```

enableSaveButtonAfterModuleVersionChange parameter:

This parameter determines whether the **Save** button is enabled after a module version changed or the module that is used to display a document changed. The parameter also determines whether the **save** option is enabled if the annotations file does not contain the ANNOTINFO marker.

The default value for this parameter is false, which disables the **Save** button under these conditions.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="enableSaveButtonAfterModuleVersionChange" value="true" />
```

migrateAnnotationsNotification parameter:

This parameter determines whether the user is notified that an annotation was moved.

For example, when you upgrade to a different module, the new module might render the same document differently and the annotations might not be in the same place.

When this parameter is set to true, and the annotations were moved, the viewer displays a message to warn the user.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="migrateAnnotationsNotification" value="false" />
```

migrateAnnotationsOnModuleVersionChange parameter:

This parameter determines whether the viewer detects and moves any annotations that fall outside the boundaries of the document to a visible location.

When this parameter is set to true, the viewer detects and moves any annotations that fall outside the boundaries of the document to a visible location.

The **Save** action becomes active if any annotations are moved.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="migrateAnnotationsOnModuleVersionChange" value="false" />
```

officeEmailInstalled parameter:

This parameter specifies whether email annotations that were added before Version 4.0.2 were saved by using the ViewONE Email Viewing Module or the ViewONE Universal Viewing Module. The viewer uses this setting to display a warning message if email annotations are now being viewed with a different module.

If this parameter is set to true, IBM Daeja ViewONE sets the ANNOTINFO marker in the saved annotation data to the ViewONE Email Viewing Module.

If this parameter is set to false, IBM Daeja ViewONE sets the ANNOTINFO marker in the saved annotation data to the ViewONE Universal Viewing Module.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="officeEmailInstalled" value="true" />
```

Related reference:

“**blockEmailModule** parameter” on page 188

This parameter determines whether email can be viewed.

“ViewONE Universal Viewing Module options” on page 261

IBM Daeja ViewONE Professional introduced an optional ViewONE Universal Viewing Module.

useAnnotationDefinedModule parameter:

This parameter controls which module is loaded when the detected module or version differs from the one that was used to save the annotations for the document.

The way this parameter works is that you can assign one of the following values:

always Always load module that is defined in the annotations file.

never Always load the module that is detected by the viewer.

prompt Prompt the user as follows:

- If the module that was defined by the annotation is licensed, the user can choose to load it or load the module that is detected by the viewer.
- If the module that was defined by the annotation is not licensed, the user can choose to load the module that is detected by the viewer, or cancel.

Or you can assign any combination of the following values:

hasuneditable

Load the module that is defined in the annotations file if any uneditable annotations are present. Otherwise, load the module that is detected by the viewer.

hassolid

Load the module that is defined in the annotations file if any solid annotations are present. Otherwise, load the module that is detected by the viewer.

hasoutofbounds

Load the module that is defined in the annotations file if any out of bounds annotations are present. Otherwise, load the module that is detected by the viewer.

Values can be set as a comma-separated list (for example, *hassolid,hasoutofbounds*). The defined filter is loaded if any of the conditions are met.

Important: If you enter an invalid combination of values (for example *always, outofbounds*), then the viewer ignores whatever is set and uses the default value instead.

The default value for this parameter is *always*.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Annotations Module

Example

```
<param name="useAnnotationDefinedModule" value="prompt" />
```

Notes

You can use the note parameters to the default style for note annotations. You can also configure sticky note annotations and tooltips.

annotationNoteRectangular parameter:

This parameter determines whether the rectangular style without leaf turn up is used by default for note annotations.

The default value for this parameter is false. When this parameter is set to false, the rectangle has leaf turn up on the lower right corner.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationNoteRectangular" value="true" />
```

annotationNoteSize parameter:

This parameter determines the default size for note annotations.

The value for this parameter specifies the width and height of note annotations in pixels. The first number in the value is the width. The second number is the height.

The default value of this parameter is 50, 65.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationNoteSize" value="40, 55" />
```

annotationNoteTextWrapping parameter:

This parameter determines whether text wraps in Sticky Note annotations.

If this parameter is set to true, text that is entered in a Sticky Note annotation word wraps at the edge of the entry box. If this parameter is set to false, scrolls rather than wraps.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.436 or later
- IBM Daeja ViewONE Professional V1.0.436 or later
- ViewONE Annotations Module

Example

```
<param name="annotationNoteTextWrapping" value="true" />
```

annotationSaveNoteId parameter:

This parameter is used to store the *StickyNote* ID that is displayed.

When the annotations are reloaded, the IDs remain the same.

The default value of this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.130 or later
- IBM Daeja ViewONE Professional V1.1.130 or later
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSaveNoteId" value="true" />
```


showNoteIds parameter:

This parameter determines whether an identifying number is displayed inside sticky note markers.

When this parameter is set to true, the identifier is displayed in the marker. The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.316 or later
- IBM Daeja ViewONE Professional V1.0.316 or later
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="showNoteIds" value="false" />
```

showNoteTooltips parameter:

This parameter determines whether tooltips that contain the contents of sticky notes are displayed.

When this parameter is set to true, the tooltips for sticky notes are displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.326 or later
- IBM Daeja ViewONE Professional V1.0.326 or later
- ViewONE Annotations Module

Example

```
<param name="showNoteTooltips" value="false" />
```

Printing

Use the printing parameter to determine which annotations file is used as a printer header or footer.

annotationTemplate parameter:

This parameter provides a URL to an annotations file to be used as a print header or footer. It specifies the location of the annotations definition file relative to the applet's codebase.

The value can also specify a server object (such as CGI, EXE, or ASP), that streams an annotations file to the applet.

Parameters can be added to the URL, such as **user ID** or **document ID**, to provide user or document-specific annotations. If you use a server object to supply or stream this file, then the server object must handle any parameters that you include.

The annotations file must conform to the ViewONE Annotations Module file specification described in *Creating an annotation template*. However, a few extra rules apply:

- Template definition annotations always have a PAGE value of -1.
- Annotation substitutions might be used within the annotations file.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTemplate" value="../templates/t1.ant" />
<param name="annotationTemplate" value="http://mysite/myscript.pl?tmpID=8791" />
```

Related tasks:

 Creating an annotation template

Related reference:

“annotationSubstitution<N> parameter” on page 123

This parameter specifies a list of options that match up to corresponding options within text type annotations from the annotations definition data, which is specified by the annotationTemplate parameter.

Related information:

Creating an annotation template

Annotation data is stored in a separate file from the document. You can create or edit an annotation definition file to use as a template to apply annotations consistently to your documents.

Redaction

You can use the redaction parameters to determine which redaction and text stamps are available and configure the redaction parameters.

redactAnnotationStamp<N> parameter:

When the user clicks the **Redact annotation toolbar** icon, a menu of items that can be used to create predefined redaction or text stamps is displayed. This parameter determines which predefined redaction or text stamps are displayed in the menu.

This parameters are works with the **redactAnnotationStampProperties<N>** parameter. By default, if no corresponding **redactAnnotationStampProperties<N>** parameter is defined, the created stamp is a text annotation.

To allow the user to draw a rectangular redaction, you must set the type property. You also use the **redactAnnotationStampProperties<N>** parameter to set additional properties, such as background color, foreground color, and text size. For more information, see the **redactAnnotationStampProperties<N>**.

You must number each stamp sequentially. For example, to define **redactAnnotationStamp2**, you must also define **redactAnnotationStamp1**.

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V1.1.168 or later
- ViewONE Annotations Module

Example

```
<param name="redactAnnotationStamp1" value="Redaction" />
<param name="redactAnnotationStamp2" value="Confidential" />
```

Related reference:

“**redactAnnotationStampProperties**<N> parameter”

This parameter specifies the properties for a redaction annotation and is valid only when the **redactAnnotationStamp**<N> parameter is used.

redactAnnotationStampProperties<N> parameter:

This parameter specifies the properties for a redaction annotation and is valid only when the **redactAnnotationStamp**<N> parameter is used.

Each instance of the parameter corresponds to an instance of the **redactAnnotationStamp**<N> parameter. For example, **redactAnnotationStampProperties1** corresponds to **redactAnnotationStamp1**.

You must number each stamp sequentially. For example, to define **redactAnnotationStampProperties2**, you must also define **redactAnnotationStampProperties1**.

The format of the properties string is as follows:

```
"<property1=value1><property2=value2>...
```

For example, `<color=blue>`, or

```
<color=blue><border=0><type="redact">
```

The following table contains the full list of properties.

Property name	Description and value options
Color	Color of text or line (or transparency color for image stamps)
FillColor	Background Color of text or fill color of shape
Border	<ul style="list-style-type: none">• 1 = The annotation has a border• 0 = The annotation does not have a border
Menu	Menu text to display (for stamp menus)
Rotation	Rotation of annotation (0, 45, 90, 180 or 270)
Label	Text for the annotation label
HelpText	Tooltip text
Owner	Owner (user name) text
FontHeight	Height in pixels for text or stamp annotations
Print	<ul style="list-style-type: none">• True = users can print annotation• False = users cannot print annotation
Read	<ul style="list-style-type: none">• True = users can read annotation• False = users cannot read annotation
Modify	<ul style="list-style-type: none">• True = users can modify annotation• False = users cannot modify annotation

Property name	Description and value options
Delete	<ul style="list-style-type: none"> • True = users can delete annotation • False = users cannot delete annotation
Scale	Scale factor for image stamp annotations (for example, 1.5)
ModifySecurity	<ul style="list-style-type: none"> • True = users can modify annotation security • False = users cannot modify annotation security
Type	<ul style="list-style-type: none"> • Text = the created annotation is a text stamp • Redact = the created annotation is a normal redaction annotation

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V1.1.168 or later
- ViewONE Annotations Module

Example

To create a normal rectangular black redaction with a redaction menu entry that specifies the text "Black redaction", use the following code:

```
<param name="redactAnnotationStampProperties1" value="<menu=black redaction>
  <fillcolor=black><border=0><type=redact>">
```

To create a white redaction with black text with a redaction menu entry that specifies the text "White text redaction", use the following code. The text that appears on the annotation is defined in the **redactAnnotationStamp**<N> parameter.

```
<param name="redactAnnotationStampProperties2" value="<menu=white text redaction/>
  <fillcolor=white><border=0><color=black><type=text>" />
```

Related reference:

"**redactAnnotationStamp**<N> parameter" on page 100

When the user clicks the **Redact annotation toolbar** icon, a menu of items that can be used to create predefined redaction or text stamps is displayed. This parameter determines which predefined redaction or text stamps are displayed in the menu.

reEnableTextAnnotClipping parameter:

This parameter determines whether a text redaction stamp is automatically resized to display all of the text if the user draws the stamp smaller than the text that it contains.

When a redaction stamp is created, the user can create the stamp at any size they prefer. Some or all of the text in the stamp might be missing if the user draws the stamp smaller than the text that it contains.

When this parameter is set to true, the stamp auto-resizes the redaction to ensure that all of the text fits into the container.

When this parameter is set to false, the text might be cropped. The default value is false.

Requirements

- IBM Daeja ViewONE Standard V4.0.38 or later
- IBM Daeja ViewONE Professional V4.0.38 or later
- ViewONE Annotations Module

Example

```
<param name="reEnableTextAnnotClipping" value="true" />
```

transparentRedactionColor parameter:

This parameter specifies the fill color that is used for redaction annotations in semi-transparent mode.

This parameter applies only when redactions are made semi-transparent with either the **setRedactionIsSemiTransparent** JavaScript call or the **setAnnotationsSemiTransparent** JavaScript call. See the *IBM Daeja ViewONE JavaScript Reference Manual* for further details.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is RGB 245, 245, 245.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="transparentRedactionColor" value="yellow" />;  
<param name="transparentRedactionColor" value="200,200,200" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

“setRedactionIsSemiTransparent method” on page 292

Sets the redaction mode in IBM Daeja ViewONE to semitransparent.

“setAnnotationsSemiTransparent method” on page 290

Sets whether the annotations are semitransparent.

Resolution

Use the **tiff1dpiAllowed** parameter to determine the resolution to be applied to annotations that are associated with 1 DPI images.

tiff1dpiAllowed parameter:

This parameter determines whether the size and position of annotations that are associated with 1 DPI images are to be calculated using a resolution of 1 DPI. The annotation size and position are modified, if they were previously calculated at 100 DPI, when the image data is loaded for a 1 DPI image.

If this parameter is set to false, the size and position are calculated using a resolution of 100 DPI. You might choose this setting if you are also using another viewer or algorithm that treats 1 DPI images as 100 DPI images to calculate annotation size and position.

This parameter applies only when the annotation size and position are specified in inches.

By default, this parameter is set to true.

Requirements

- IBM Daeja ViewONE Professional V4.1.3 or later
- ViewONE Annotations Module

Example

```
<param name="tiff1dpiAllowed" value="false" />
```

Saving

You can use the save parameters to determine whether annotations are saved automatically and whether users are prompted to save annotations. POST data is also configured by the save parameters. You can also determine how annotations are handled by server-side components. By managing server-side components, you can reduce the amount of data that is sent to the server.

annotationAutoPrompt parameter:

This parameter determines whether users are prompted to save annotations and whether unsaved annotations are automatically saved.

If this parameter is set to false, users are not prompted to save annotations. Unsaved changes to annotations are not saved. Instead, annotations are saved only when users specifically click **Save annotation changes** or when the equivalent JavaScript method is called.

If the **annotationAutoSave** parameter is specified, the **annotationAutoPrompt** parameter overrides that parameter.

This parameter applies when the applet is closed by a web page change or when a document is closed by a document open or close action.

This parameter takes effect regardless of whether you are using the **annotationSaveServlet** parameter or **annotationSavePost** parameter.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationAutoPrompt" value="false" />
```

Related reference:

“**annotationSaveServlet** parameter” on page 108

This parameter specifies the location of the server object that implements the POST request to save annotations in a single stream of data.

“**annotationSavePost** parameter” on page 107

This parameter specifies the location of the server object that implements the POST request to save annotations with the URL-encoded content type to deliver the annotation data in chunked data segments.

annotationAutoSave parameter:

This parameter determines whether annotations are saved automatically.

If this parameter is set to true, users are not prompted to save annotations. Instead, annotations are sent automatically to the server-side save object.

If this parameter is set to false, the user is prompted to save annotations.

The parameter applies when the viewer is closed by a web page change, or when the document is closed by an open or close action. The parameter takes effect regardless of whether you are using the **annotationSaveServlet** parameter or **annotationSavePost** parameter.

The default value of this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationAutoSave" value="true" />
```

Related reference:

“**annotationSaveServlet** parameter” on page 108

This parameter specifies the location of the server object that implements the POST request to save annotations in a single stream of data.

“**annotationSavePost** parameter” on page 107

This parameter specifies the location of the server object that implements the POST request to save annotations with the URL-encoded content type to deliver the annotation data in chunked data segments.

annotationGzip parameter:

This parameter determines whether annotation save data is compressed before it is sent to the server-side component. You might use this parameter to reduce the amount of data that is sent to the server.

If you set this parameter to true, IBM Daeja ViewONE uses gzip compression encoding to send annotations that are to be saved to the server-side component. That component must be able to extract the streamed annotation data.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.126 or later
- IBM Daeja ViewONE Professional V1.1.126 or later
- ViewONE Annotations Module

Example

```
<param name="annotationGzip" value="true" />
```

annotationLocalSave parameter:

This parameter determines whether annotations are saved by calling the server-side object.

If this parameter is set to true and IBM Daeja ViewONE uses the `file:///` protocol to retrieve an image, IBM Daeja ViewONE does not call the server-side object to save annotations. Instead, IBM Daeja ViewONE saves an annotations definition file with the image.

If this parameter is set to false, IBM Daeja ViewONE always saves annotations by calling the specified server-side object.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationLocalSave" value="false" />
```

annotationPostPrefix parameter:

This parameter works with the **annotationSavePost** parameter to specify parameters to be added to the POST data for the **annotationSavePost** feature. For example, you might use these parameters to provide additional document or security parameters to the POST request.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

The following example demonstrates prefixing `p1=value1` and `p2=value2` to the **POST** data. The **POST** object is then parses the posted data to extract all the parameters. In this example, the extracted parameters include **p1** and **p2** along with the usual parameters such as **size**, **numdata**, **data01**, and **data02**.

```
<param name="annotationPostPrefix" value="p1=value1&p2=value2" />
```

Related reference:

“**annotationSavePost** parameter”

This parameter specifies the location of the server object that implements the POST request to save annotations with the URL-encoded content type to deliver the annotation data in chunked data segments.

annotationSavePost parameter:

This parameter specifies the location of the server object that implements the POST request to save annotations with the URL-encoded content type to deliver the annotation data in chunked data segments.

You must supply the server-side object, which might be a CGI script, an EXE program, an Active Server Page (ASP), or a JSP page. This object must conform to the specification for save POST objects in ViewONE Annotations Module.

If you specify a relative address for the **annotationSavePost** parameter value, the location it defines is relative to the applet codebase.

You can add parameters to the POST data by using the **annotationPostPrefix** parameter. For example, you might add as user ID or document ID to send user-specific or document-specific annotations, IBM Daeja ViewONE then adds on its own annotations data.

If you are unsure what IBM Daeja ViewONE is sending through to the object, enable the annotations debug output by setting the following parameter:

```
<param name="traceAnnotation" value="true" />
```

When a user clicks the annotations save button, the server-side object is called only if the document that is displayed was retrieved through a web server. The object is not called if the document was not retrieved through a web server, for example, if the document was retrieved through the file:/// protocol.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSavePost" value="http://mysite/annotationsave.dll" />
```

Related reference:

 [Configuring the object by using URL-encoded data](#)

“**annotationPostPrefix** parameter” on page 106

This parameter works with the **annotationSavePost** parameter to specify parameters to be added to the POST data for the **annotationSavePost** feature. For example, you might use these parameters to provide additional document or security parameters to the POST request.

annotationSavePostParamNames parameter:

This parameter determines the names of the following HTTP POST parameters: **numData**, **size**, and **data<N>**. These parameters are specified in the URL that is used to send save annotations.

The default value for this parameter is an empty string, which indicates that the default parameter names are used.

Requirements

- IBM Daeja ViewONE Standard V3.1.50 or later
- IBM Daeja ViewONE Professional V1.1.50 or later
- ViewONE Annotations Module

Example

```
<param name="annotationSavePostParamNames" value="size=p_size;numdata=p_numdata" />
```

annotationSaveServlet parameter:

This parameter specifies the location of the server object that implements the POST request to save annotations in a single stream of data.

You must supply the server object, which might be a CGI script, an EXE program, an Active Server Page (ASP), or a JSP page. This object must conform to the specification for save POST objects in ViewONE Annotations Module.

If you specify a relative address for the **annotationSaveServlet** parameter value, the location it defines is relative to the viewer codebase.

You can add parameters to the URL. For example, you might add as user ID or document ID to send user-specific or document-specific annotations.

When a user clicks the annotations save button, the server object is called only if the document that is displayed was retrieved through a web server. The object is not called if the document was not retrieved through a web server, for example, if the document was retrieved through the `file:///` protocol.


Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSaveServlet" value="http://mysite/servlet/annotationsave.class?userID=12&docID=8791" />
```

Related reference:

 [Configuring the object as a servlet](#)

[“annotationFile parameter” on page 79](#)

This parameter specifies the annotations definition file that contains the list of the annotations to display and their associated parameters.

Security

You can use the annotation security parameters to determine the default security settings for specified annotations. You can specify whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security. The parameters also configure the administrator that can edit security settings and password options in the security dialog box.

annotationDefaults parameter:

This parameter determines the default security settings for specified annotations.

Tip: If you are using the ViewONE Permanent Redaction Server Module, you can also use this parameter to specify the annotation types that are burnable. For more information, see “**annotationDefaults** parameter” on page 259.

The value of this parameter uses the following format:

```
value = "type {init='1,1,0,1,0,0'}"
```

If you are also using this parameter to specify the annotation types that are burnable, the value uses the following format:

```
value="[type {burnable=burnable_state}];  
type {init='1,1,0,1,0,0'}"
```

The *type* variable indicates the type of annotations to which the security settings apply. Use one of the following values for this variable:

- all
- line
- arrow
- text
- note
- highlight
- highlightPoly
- rectangle
- redact
- redactPoly
- poly
- openPoly
- oval
- freehand
- stamp

Specify all to set the security defaults for all types of annotations.

The init values turn on or off the following default security settings:

- **READ**
- **MODIFY**
- **EXECUTE**
- **PRINT**
- **DELETE**
- **MODIFYSECURITY**

A value of 1 turns on the settings, and a value of 0 turns off the settings.

Important: The security settings are set only if the viewer is in extended annotation security mode.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

The following example turns on the **READ**, **MODIFY**, and **PRINT** default security settings for all annotations:

```
<param name="annotationDefaults" value="all {init='1,1,0,1,0,0'}" />
```

The following example turns on the **READ**, **MODIFY**, and **DELETE** default security settings for line and arrow annotations:

```
<param name="annotationDefaults" value="line {init='1,1,0,0,1,0'}  
      arrow {init='1,1,0,0,1,0'}" />
```

annotationEditPasswordModify parameter:

When this parameter is set to true, a further option is presented in the security dialog box when you use extended annotation security.

This dialog box allows the user to set a password for modifying annotations. The annotation text cannot be edited, and the annotation cannot be moved without the password. Only the owner or an administrator can change these passwords after they are created. The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationEditPasswordModify" value="true" />
```

annotationEditPasswordSecurity parameter:

When this parameter is set to true, a further option is presented in the security dialog box when you use extended annotation security.

This dialog box allows the user to set a password for setting the security for annotations. The password is required to modify the annotation security. Only the owner or an admin user is able to change these passwords after they are created. The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationEditPasswordSecurity" value="true" />
```

annotationEditPasswordText parameter:

This parameter is set to true, a further option is presented in the security dialog box when you use the extended annotation security.

This dialog box allows the user to set a password for changing the text in a text annotation. Only the owner or an admin user is able to change these passwords. The annotation can still be moved.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationEditPasswordText" value="true" />
```

annotationJavascriptExtensions parameter:

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

This parameter affects the following JavaScript methods:

- reloadAnnotations()
- setAnnotateEdit(boolean)
- int getNumAnnotations(type, page)
- string getAnnotationLabels(type, page)
- string getAnnotationLabels(label)
- addAnnotation(annotationProperties)
- modifyAnnotation(label, annotationProperties)
- deleteAnnotation(label)
- deleteAllAnnotations(type, page)
- startAnnotation(type)
- setRedactionIsSemiTransparent(boolean)
- setAnnotationsSemiTransparent(boolean, type)

The default value for this parameter is false, which enables the viewer to accept calls for these methods.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationJavascriptExtensions" value="true" />
```

annotationSecurityModel parameter:

This parameter determines which model is used for annotation security.

The following table lists the values for this parameter.

Value	Description
1	Simple Annotation Security
2	Extended Annotation Security

The default value for this parameter is 1.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSecurityModel" value="2" />
```

Related reference:

 [Simple annotation security model](#)

Related information:

Simple annotation security model

You can implement basic security for the ViewONE Annotations Module by using the EDIT and VIEW annotation properties to define whether a user can edit, view, or print a particular annotation. You can also use the **annotateEdit** HTML parameter to disable the edit option for all annotations.

userAdmin parameter:

When this parameter is set to true, the current user is specified as being an administrator, with the ability to edit security settings for any annotation.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="userAdmin" value="true" />
```

userID parameter:

This parameter specifies the value to identify a user for user-specific areas of IBM Daeja ViewONE.

Within the context of annotations, this parameter applies to annotation's **createdID** and **modifiedID** properties. When IBM Daeja ViewONE sends new or modified

annotations to the server, it includes the **createdID** and **modifiedID** properties for each annotation to be set to the appropriate user ID. There is no default value.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="userID" value="user1" />
```

Selection

You can use the selection parameters to determine the mouse-click operation that is used to activate annotations and to determine whether annotation select mode is automatically turned on.

annotationDb1Click parameter:

This parameter determines the mouse-click operation that is used to activate annotations.

Activating an annotation is not the same as selecting an annotation. Double-click an annotation to activate it and you can enter text. Single-click an annotation to select it, and the annotation can be moved around the page.

This parameter applies only when the user interface is not in annotation edit mode, when the annotations toolbar is hidden or disabled. This parameter impacts sticky notes and annotation hyperlinks.

If this parameter is set to true, the user double-clicks to activate an annotation.

The default value of this parameter is false. The user single-clicks to activate an annotation.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationDb1Click" value="true" />
```

defaultAnnotationSelection parameter:

This parameter determines whether the annotation select mode is automatically turned on. When the select mode is on, the mouse highlights annotations and the user can click a highlighted annotation to edit that annotation.

The default value for this parameter is false, which turns off annotation selection mode.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="defaultAnnotationSelection" value="true" />
```

Stamps

You can use the stamp parameters to specify stamps that are used instead of the default stamps, and configure the appearance of the stamps.

annotationStamp<N> parameter:

This parameter specifies the stamps that are used in place of IBM Daeja ViewONE default stamps. The stamp that is specified can be either text or image based.

An image stamp is an overlaid image, which might be in any of the image formats supported by IBM Daeja ViewONE. An image-based stamp is recognized by the use of the `image:` option. The path that follows the option specifies the location of the annotations definition file relative to the viewer's codebase.

A text-based stamp can contain several options that are substituted for actual values at run time.

The following table lists the options.

Option	Description
<date>	Inserts the date, time, and time zone from the user's workstation at the time the stamp is placed.
<dateonly>	Inserts the date from the user's workstation at the time the stamp is placed.
<userid>	Inserts the user ID as specified by the viewer's userID parameter.

More properties can be specified by using the **annotationStampProperties**<N> parameter, such as a transparent color for image stamps, or color and size for text stamps. You can change the date style for the <date> and <dateonly> options through the **annotationDateStyle** parameter.

IBM Daeja ViewONE default stamps are listed in *IBM Daeja ViewONE Default Stamps*, but these stamps are removed with use of this parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationStamp1" value="image:logo.tif" />  
<param name="annotationStamp2" value="Today's date is <date>" />
```


Related reference:

“IBM Daeja ViewONE Default Stamps” on page 263

The viewer's default stamps are listed in the following table. The stamps are used on documents that need to be tagged with a predefined set of annotations.

“**annotationStampProperties**<N> parameter”

This parameter specifies various properties for stamps to use in place of IBM Daeja ViewONE's default stamps.

“**annotationDateStyle** parameter” on page 84

This parameter determines the date format that is used for *date* and *dateonly* values in the **annotationStamp**<N> parameter. The **annotationDateStyle** parameter also alters the date format that is displayed in the sticky note dialog box.

annotationStampProperties<N> parameter:

This parameter specifies various properties for stamps to use in place of IBM Daeja ViewONE's default stamps.

The stamp properties that are specified can be either text or image based.

The parameter is for use only when the **annotationStamp**<N> parameter is used. Each use of the parameter corresponds to an instance of the **annotationStamp**<N> parameter. For example, **annotationStampProperties1** corresponds to **annotationStamp1**. If the parameter is used at all, then all stamp properties for the stamps menu must be specified. For example, you cannot have **annotationStampProperties2**, but not **annotationStampProperties1**.

The format of the properties string is as follows:
<property1=value1><property2=value2>...

Example strings:

<color=blue>, or <color=blue><rotation=45>

The following table contains the full list of properties.

Property name	Description and value options
Color	Color of text or line (or transparency color for image stamps)
FillColor	Background Color of text or fill color of shape
Border	<ul style="list-style-type: none">• 1 = The annotation has a border• 0 = The annotation does not have a border
Menu	Menu text to display (for stamp menus)
Rotation	Rotation of annotation (0, 45, 90, 180 or 270)
Label	Text for the annotation label
HelpText	Tooltip text
Owner	Owner (user name) text
FontHeight	Height in pixels for text or stamp annotations
Print	<ul style="list-style-type: none">• True = users can print annotation• False = users cannot print annotation

Property name	Description and value options
Read	<ul style="list-style-type: none"> • True = users can read annotation • False = users cannot read annotation
Modify	<ul style="list-style-type: none"> • True = users can modify annotation • False = users cannot modify annotation
Delete	<ul style="list-style-type: none"> • True = users can delete annotation • False = users cannot delete annotation
Scale	Scale factor for image stamp annotations (for example, 1.5)
ModifySecurity	<ul style="list-style-type: none"> • True = users can modify annotation security • False = users cannot modify annotation security

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationStampProperties1" value="<menu=Company logo><color=white>" />
<param name="annotationStampProperties2" value="<menu=Today's Date>
  <color=black><rotation=45>" />
```

Related reference:

“**annotationStamp**<N> parameter” on page 114

This parameter specifies the stamps that are used in place of IBM Daeja ViewONE default stamps. The stamp that is specified can be either text or image based.

matchStampResolution parameter:

This parameter changes the size of image stamp annotations to match the resolution of the image.

When this parameter is set to true, the sizes of image stamp annotations automatically adjust to match the resolution of the main image. For example, if a 100 DPI stamp annotation is added to a 200 DPI image, the stamp annotation size is reduced by a factor of 2 to match the main image resolution.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.508 or later
- IBM Daeja ViewONE Professional V1.0.508 or later
- ViewONE Annotations Module

Example

```
<param name="matchStampResolution" value="true" />
```

stampAnnotationKeepAspect parameter:

This parameter determines whether image stamps maintain their original aspect when stamp (image) annotation resizing is enabled.

When this parameter is set to true, the width and height cannot be resized independently. The width and height are automatically adjusted to ensure that the original image aspect is maintained. The default value is true.

When this parameter is set to false, the width and height of an image stamp annotations can be resized independently, and the original aspect ratio is not maintained.

Tip: Image stamps always display an anchor point (yellow rectangle) on each corner of the image while the image is being edited. This parameter allows each corner to be moved and to allow width and height to be adjusted independently when this parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard V3.0.550 or later
- IBM Daeja ViewONE Professional V1.0.550 or later
- ViewONE Annotations Module

Example

```
<param name="stampAnnotationKeepAspect" value="true" />
```

stampAnnotationResize parameter:

This parameter determines whether stamp annotations can be resized.

When this parameter is set to false, stamp annotations that are added to an image can be placed but not resized.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.460 or later
- IBM Daeja ViewONE Professional V1.0.460 or later
- ViewONE Annotations Module

Example

```
<param name="stampAnnotationResize" value="true" />
```

Style

You can use the style parameters to determine the default line width for new annotations and determine the maximum number of points for a freehand annotation.

annotationFreehandLimit parameter:

This parameter determines the maximum number of points for a freehand annotation.

When this limit is reached during the drawing of a freehand annotation, the annotation drawing is automatically stopped.

The value of this parameter must be greater than 1.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationFreehandLimit" value="20" />
```

defaultLineWidth parameter:

This parameter determines the default line width for new annotations.

By default, IBM Daeja ViewONE creates new annotations with a line width of eight pixels. You can use this parameter to change this value.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="defaultLineWidth" value="5" />
```

Text and rotation

You can use the text and rotation parameters to configure the style of the text for various annotations. You can also determine how annotations and images are rotated and whether text-related error messages are displayed.

allowAnnotationRotationOnTags parameter:

This parameter determines when an image with annotations is rotated based on the rotation flag in the header of the TIFF file.

If this parameter is set to false, IBM Daeja ViewONE rotates the image before applying the annotations. In effect, IBM Daeja ViewONE rotates the only the image itself.

If this parameter is set to true, IBM Daeja ViewONE rotates the image applying the annotations. Therefore, IBM Daeja ViewONE rotates the annotation data with the image. This value is the equivalent of clicking **Rotate Image** after the image is loaded.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.434 or later
- IBM Daeja ViewONE Professional V1.0.434 or later

- ViewONE Annotations Module

Example

```
<param name="allowAnnotationRotationOnTags" value="true" />
```

allowResizeAnnotationRotatedDocument parameter:

This parameter allows users to use a bounding box to resize text-based annotations in a document that is rotated. (The bounding box is disabled by default for rotated documents.)

By default, this parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard, V3.1.126 or later
- IBM Daeja ViewONE Professional, V3.1.126 or later
- ViewONE Annotations Module

Example

```
<param name="allowResizeAnnotationRotatedDocument" value="true" />
```

allowResizeRotatedAnnotation parameter:

This parameter allows users to use a bounding box to resize text-based annotations that are rotated. (The bounding box is disabled by default for rotated annotations.)

By default, this parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard, V3.1.126 or later
- IBM Daeja ViewONE Professional, V3.1.126 or later
- ViewONE Annotations Module

Example

```
<param name="allowResizeRotatedAnnotation" value="true"/>
```

allowTextRubberband parameter:

This parameter determines whether users can resize text annotations.

If this parameter is set to false, the bounding box that is used to resize text annotations is disabled. (The bounding box is sometimes called the rubber band box.)

The default value of this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="allowTextRubberband" value="false" />
```

annotationAllowControlCodes parameter:

This parameter determines whether users can enter control codes in text annotations. Control codes are characters with below code 32 in the ASCII table.

The default value for this parameter is false. Note that users can still enter CR and LF characters in text annotations when this parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationAllowControlCodes" value="true" />
```

annotationAutoWrap parameter:

This parameter determines whether text annotations wrap at the end of the page.

The default value of this parameter is true. Text annotations are automatically wrapped to multiple lines when the text reaches the edge of a page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationAutoWrap" value="false" />
```

annotationClearTextList<N> parameter:

This parameter determines the content of text annotations that are to be cleared when users select the annotations.

This parameter is used with the **annotationClearTextOnSelection** parameter. If the **annotationClearTextOnSelection** parameter is set to true and the text of an annotation matches exactly the value that is specified for the **annotationClearTextList<N>** parameter, the annotation is cleared.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationClearTextList1" value="Name expected here" />  
<param name="annotationClearTextList2" value="Comments go here" />
```

Related reference:

“annotationClearTextOnSelection parameter”

This parameter determines the behavior of a text annotation when a user selects that annotation.

annotationClearTextOnSelection parameter:

This parameter determines the behavior of a text annotation when a user selects that annotation.

If this parameter is set to true, the text annotation is cleared automatically when the user selects the annotation.

This parameter might be useful for applications where users regularly click existing text annotations with the intent of changing the entire text. This parameter assists those users by removing the need to first select and delete the text.

You can also use the **annotationClearTextList<N>** parameter to control this behavior.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationClearTextOnSelection" value="true" />
```

Related reference:

“annotationClearTextList<N> parameter” on page 120

This parameter determines the content of text annotations that are to be cleared when users select the annotations.

annotationDynamicTextCreation parameter:

This parameter determines whether IBM Daeja ViewONE is to render annotation text for complex languages by the word instead of by the character.

When a user enters annotation text in complex languages, such as Thai, each character the user enters can affect previous characters. For such languages, you might want to set this parameter to true so that IBM Daeja ViewONE renders a word at a time instead of individual characters. (A word in this case includes the characters from one space or carriage return to the next.)

If this parameter is set to false, IBM Daeja ViewONE renders the text a letter at a time as is done for less complex languages.

The default for this value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.368 or later
- IBM Daeja ViewONE Professional V1.0.368 or later
- ViewONE Annotations Module

Example

```
<param name="annotationDynamicTextCreation" value="false" />
```

annotationFont parameter:

This parameter determines the default font that is used for text annotations.

You must specify one of the following font types for this parameter: arial and courier. If you specify a different font, IBM Daeja ViewONE uses the default font type.

The default value for this parameter is arial.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationFont" value="courier" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

annotationInputKeyMapping<N> parameter:

This parameter is used to override direct keyed input for text annotations, that is, text entry that is not used within a dialog box.

The parameter value consists of the following parts that are separated by commas:

- The keyboard key on which the associated text is to be changed, specified as the key text or Unicode value.
- The replacement string, again, specified as text or Unicode.
- A list of modifiers for Ctrl, Shift, and Alt to specify the modification keys that are held down.

The first example changes any instance of CTRL-ALT-4 to the Euro symbol. This combination is the standard keyboard combination for the Euro on UK keyboards. This combination does not function correctly for some JVMs. Users that require direct typing of the Euro symbol on UK keyboards for Oracle Java plug-in Version 1.4 onwards use this parameter to override the key map combination.

The second example sets a shortcut for the text (Δ), by using CTRL-e.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationInputKeyMapping1" value="4, 0x20AC, ALT, CTRL" />  
<param name="annotationInputKeyMapping2" value="e, (Δ), CTRL" />
```


annotationSubstitution<N> parameter:

This parameter specifies a list of options that match up to corresponding options within text type annotations from the annotations definition data, which is specified by the annotationTemplate parameter.

When the template annotations file is downloaded and interpreted, the specified options are replaced with the values associated with them through this parameter.

The number of annotationSubstitution parameters that are specified is not limited to the number of pages in the document, but is limited to the number of options that are in the annotations template file. Each instance of an annotationSubstitution parameter can contain only the options and values for one page at a time.

Important: The value string must contain only one colon after the page marker. No other colons are allowed in the value string. Also, the only comma characters that are allowed are those characters separating the substitution values; you cannot use commas in your substitution text.

The value string uses the following syntax:

```
page_number: <option>=associated_value[, <option>=associated_value]
```

For example, to review the text annotations on page one of the document and replace any instances of the <document> option with my document and replace any instances of the <page> option with 1, use the following example code:

```
1: <document>=my document, <page>=1
```

For example, to review the text annotations on page two of the document and replace any instances of the <page> option with 2 and any instances of the <id> option with AGH13534, use the following example code:

```
2: <page>=2, <id>=AGH13534
```

The two special date options <DATE> and <DATEONLY> are substituted with the system date of the local workstation. <DATEONLY> substitutes the date only. <DATE> substitutes the current date and time. The format of the date string can be changed by using the annotationDateStyle parameter.

The <EMPTY> substitutes tags with empty strings to remove a tag that is contained in an annotation. For example, to clear any <DOCUMENT> options in the annotation on page 1, use the following code:

```
1: <document>=<EMPTY>
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationSubstitution1" value="1: <document>=my document, <page>=1" />  
<param name="annotationSubstitution2" value="2: <page>=2, <id>=AGH13534" />
```

Related reference:

“**annotationTemplate** parameter” on page 99

This parameter provides a URL to an annotations file to be used as a print header or footer. It specifies the location of the annotations definition file relative to the applet's codebase.

“**annotationDateStyle** parameter” on page 84

This parameter determines the date format that is used for *date* and *dateonly* values in the **annotationStamp**<N> parameter. The **annotationDateStyle** parameter also alters the date format that is displayed in the sticky note dialog box.

annotationTabLength parameter:

This parameter sets the default tab length to the value specified.

This length specifies the number of spaces to insert into a text annotation when the user presses the TAB key. The default tab value is 3.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTabLength" value="4" />
```

annotationTemplateValuesFile parameter:

This parameter is an alternative to the **annotationSubstitution**<N> parameter. It points IBM Daeja ViewONE to a text file that lists the **annotationTemplate** text annotation options and their associated values.

The file location is specified through a URL that is relative to the applet's codebase. The URL can specify a server object (such as CGI, EXE, or ASP) that streams a template values file to the applet.

Parameters can be added to the URL, such as **user ID** or **document ID**, to provide user or document-specific template values. If you use a server object to supply or stream this file, the server object must handle any parameters that you include.

The text file only contains the string values. The format is similar to the **annotationSubstitution**<N> parameter format, as follows:

```
page_number: <option>=associated_value[, <option>=associated_value]
```

For example, to review the text annotations on page one of the document and replace any instances of the <document> option with my document and replace any instances of the <page> option with 1, use the following example code:

```
1: <document>=my document, <page>=1
```

For example, to review the text annotations on page two of the document and replace any instances of the <page> option with 2 and any instances of the <id> option with AGH13534, use the following example code:

```
2: <page>=2, <id>=AGH13534
```

The values are listed one per line in the text file. A carriage-return, a line-feed, or a carriage-return and line-feed then delimit the end of the line. For example:

```
1: <document>=my document, <page>=1
2: <page>=2, <id>=AGH13534
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTemplateValuesFile" value="../templates/t1.txt" />
<param name="annotationTemplateValuesFile"
  value="http://mysite/myscript.pl?tmpID=8791" />
```

annotationTextAlignment parameter:

This parameter determines whether the buttons for aligning text within text-based annotations are enabled.

When this parameter is set to false, the buttons for aligning text within text-based annotations are disabled.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V4.0.18 or later
- IBM Daeja ViewONE Professional V4.0.18 or later
- IBM Daeja ViewONE Virtual
- ViewONE Annotations Module

Example

```
<param name="annotationTextAlignment" value="false" />
```

annotationTextLimit parameter:

This parameter specifies the maximum number of characters for the text and sticky note annotations. When this limit is reached when a user types a text annotation, drawing is automatically stopped.

When this limit is reached when a user types a sticky note annotation, a message is displayed to the user after the user clicks **OK**. The user must close this message, and then the sticky note text is automatically redisplayed so that the text can be changed (or canceled).

The value for this parameter must be greater than 1.

Requirements

- IBM Daeja ViewONE Standard V3.0.374 or later
- IBM Daeja ViewONE Professional V1.0.374 or later
- ViewONE Annotations Module

Example

```
<param name="annotationTextLimit" value="64" />
```

annotationTextLimitReachedMessage parameter:

This parameter determines whether an error message is displayed when the maximum number of characters is reached in a text annotation.

When this parameter is set to false, no error message is displayed to the user when the number of characters is reached, and no more text input is allowed. The maximum number of characters is defined by the **annotationTextLimit** parameter.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.434 or later
- IBM Daeja ViewONE Professional V1.0.434 or later
- ViewONE Annotations Module

Example

```
<param name="annotationTextLimitReachedMessage" value="false" />
```

Related reference:

“**annotationTextLimit** parameter” on page 125

This parameter specifies the maximum number of characters for the text and sticky note annotations. When this limit is reached when a user types a text annotation, drawing is automatically stopped.

defaultFontHeight parameter:

This parameter specifies the default font height for new text annotations.

By default, IBM Daeja ViewONE creates new text annotations with a font height of 28 pixels for image documents and 48 pixels for text documents. You use this parameter to adjust these values. Both values are set to the specified value.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="defaultFontHeight" value="20" />
```

Text color

You can use the text color parameters to specify the default color for annotation text and the default fill color for filled text annotations.

annotationTextColor parameter:

This parameter specifies the default color for annotation text.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default values are 255, 0, 0 (red). For example,

```
<param name="annotationTextColor" value="255, 0, 0" />
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTextColor" value="0, 0, 0" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

annotationTextFillColor parameter:

This parameter specifies the default fill color for filled text annotations.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default values are 255, 255, 153 (pale yellow). For example,

```
<param name="annotationTextFillColor" value="255, 255, 153" />
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="annotationTextFillColor" value="255, 255, 255" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

Toolbars, buttons, and menus

You can use the parameters to determine which annotation types are affected when you display and hide toolbars, buttons, and menus.

annDeleteToolbarButton parameter:

This parameter determines whether the annotation toolbar contains a button to display a menu for deleting annotations.

If this parameter is set to true, the button is added to the annotation toolbar. The menu that is associated with this button provides options to delete all annotations on the current page or to delete all annotations in the current document.

By default, both menu options prompt the user to confirm the deletion of the annotations. You can control the confirmation dialog boxes by using the **annDeleteAllPageConfirm** parameter and **annDeleteAllDocConfirm** parameter.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V3.1.168 or later
- ViewONE Annotations Module

Example

```
<param name="annDeleteToolBarButton" value="true" />
```

Related reference:

“**annDeleteAllPageConfirm** parameter” on page 86

This parameter determines whether a confirmation dialog box opens when a user selects the menu item to delete all annotations on the current page.

“**annDeleteAllDocConfirm** parameter” on page 86

This parameter determines whether a confirmation dialog box is opened when a user selects the menu item to delete all annotations from the current document from the **Delete Annotation** button.

annotationAllowHideAll parameter:

This parameter determines whether users can click **Hide Annotations** to hide all annotations, including those annotations that the users cannot edit.

If this parameter is set to true, users can hide all annotations.

The default value for this parameter is false.

Tip: To use **Hide Annotations** to hide PDF annotations, you must also set the **alwaysShowPDFAnnotations** parameter to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationAllowHideAll" value="true" />
```

Related reference:

“**alwaysShowPDFAnnotations** parameter” on page 205

This parameter determines whether the annotations that are stored in a PDF are shown when the user clicks **Show/Hide Annotations**.

annotationEditButton parameter:

This parameter determines whether the **Annotation Edit** button appears in the top toolbar.

The **Annotation Edit** button is used to hide and show the annotation toolbar.

The default value for this parameter is true, which displays the **Annotation Edit** in the top toolbar.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example



```
<param name="annotationEditButton" value="false" />
```








annotationHideButtons parameter:

This parameter determines which buttons are hidden on the annotations toolbar. It also determines whether the **Show Annotations** button is hidden on the IBM Daeja ViewONE status bar.

Hiding a button disables only the ability to create an annotation through its user interface. JavaScript methods can still be used to create annotations.

The following table lists the values you can specify for this parameter. To hide multiple buttons, use a comma-delimited string for the values. The values are not case-sensitive.

Button	Value	Button name
	angle	Draw angle
	arrow	Draw arrow
	circle	Draw circle
	freehand	Draw freehand line
	highlight	Draw highlight
	highlightPoly	Draw polygon highlight
	line	Draw line
	note	Draw sticky note
	openPoly	Draw open polygon
	oval	Draw oval
	poly	Draw polygon
	rectangle	Draw rectangle

Button	Value	Button name
	redact	Draw redaction
	redactPoly	Draw polygon redaction
	restore	Restore annotations
	ruler	Draw ruler
	save	Save annotation changes
	select	Select mode
	show	Show annotations
	solidText	Draw solid text
	square	Draw square
	stamp	Draw stamp
	text	Draw transparent text
	Hyperlink	Hyperlink annotation

If you disable all buttons, the annotations toolbar is disabled. User cannot create annotations. Instead, users can only edit existing annotations.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module




















Example

```
<param name="annotationHideButtons" value="freehand, hyperlink, text" />
```

annotationHideContextButtons<N> parameter:

When used with the **annotationHideContextButtonsIds**<N> parameter, this button determines which context buttons are hidden for specified annotation types.

The following table lists the values that you can specify for this parameter. The values are not case-sensitive. To hide multiple buttons, enter the values in a comma-delimited string.

Button	Value	Name
	save	Fix button
	security	Edit security button
	text	Text editor button
	increaseline	Increase line width
	decreaseline	Decrease line width
	increasefont	Increase text font size
	decreasefont	Decrease text font size
	increasearrowhead	Increase arrowhead size
	decreasearrowhead	Decrease arrowhead size
	linecolor	Line color chooser
	fillcolor	Fill color chooser
	strikethrough	Strike through text (Version Standard 3.0.332+, Professional 1.0.332+)
	transparent	Make text semi-transparent
	and help texthyperlink	Hyperlink dialog box
	rotater	Rotate clockwise
	rotatel	Rotate counterclockwise
	angleflip	Flip angle (changes between obtuse & acute) (V3-only)
	behind	Move behind
	delete	Delete (delete key is still enabled)

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- ViewONE Annotations Module

Example

The following example hides the **Save** button and **Hyperlink** button for note annotations and just the **Save** button for freehand annotations:

```
<param name="annotationHideContextButtons1" value="save, hypertext" />
<param name="annotationHideContextButtonsIds1" value="note" />
<param name="annotationHideContextButtons2" value="save" />
<param name="annotationHideContextButtonsIds2" value="freehand" />
```

The following example also hides the **Save** button and **Hyperlink** button for note annotations and just the **Save** button for freehand annotations:

```
<param name="annotationHideContextButtons1" value="save" />
<param name="annotationHideContextButtonsIds1" value="note, freehand" />
<param name="annotationHideContextButtons2" value="hypertext" />
<param name="annotationHideContextButtonsIds2" value="note" />
```

Important: You can hide buttons for all annotation types by not specifying a corresponding **annotationHideContextButtonsIds<N><** value. However, this action works only for the last pair. Therefore, the following example does not hide the **Hyperlink** button for all annotation types:

```
<param name="annotationHideContextButtons1" value="hypertext" />
<param name="annotationHideContextButtons2" value="save" />
<param name="annotationHideContextButtonsIds2" value="note" />
```

The following example shows the correct way to hide the **Hyperlink** button for all annotation types:

```
<param name="annotationHideContextButtons1" value="save" />
<param name="annotationHideContextButtonsIds1" value="note" />
<param name="annotationHideContextButtons2" value="hypertext" />
```

Related reference:

“**annotationHideContextButtons** parameter” on page 134

This parameter determines which buttons are hidden on the annotation context-sensitive toolbar.

annotationHideContextButtonsIds<N> parameter:

When used with the **annotationHideContextButtons<N>** parameter, this button determines which context buttons are hidden for specified annotation types.

You can specify the following values in a comma-delimited string for this parameter:

- arrow
- freehand
- highlight
- highlightpoly
- line
- note
- openpoly
- oval
- poly
- rectangle
- redact

- redactpoly
- stamp
- text

The values are not case sensitive.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationHideContextButtonsIds1" value="note, text" />
```

Related reference:

“**annotationHideContextButtonsIds** parameter”

This parameter determines the annotation types to which the **annotationHideContextButtons** parameter value is applied.

annotationHideContextButtonsIds parameter:

This parameter determines the annotation types to which the **annotationHideContextButtons** parameter value is applied.

For example, if you specify hyperlink as an **annotationHideContextButtons** value and specify note as an **annotationHideContextButtonsIds** value, then **context-sensitive hyperlink** dialog box is disabled for the note annotation type.

You can specify the following values in a comma-delimited string for this parameter:

- arrow
- freehand
- highlight
- highlightpoly
- line
- note
- openpoly
- oval
- poly
- rectangle
- redact
- redactpoly
- stamp
- text

The values are not case sensitive.

By default, the **annotationHideContextButtons** value is applied to all annotation types.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationHideContextButtonsIds" value="note, text" />
```

Related reference:

“**annotationHideContextButtons** parameter”












This parameter determines which buttons are hidden on the annotation context-sensitive toolbar.









annotationHideContextButtons parameter:

This parameter determines which buttons are hidden on the annotation context-sensitive toolbar.

Hiding a button disables only the ability to modify an annotation's default properties through the user interface.

The following table lists the values that you can specify for this parameter. The values are not case-sensitive. To hide multiple buttons, enter the values in a comma-delimited string.

Button	Value	Name
	save	Fix button
	security	Edit security button
	text	Text editor button
	increaseline	Increase line width
	decreaseline	Decrease line width
	increasefont	Increase text font size
	decreasefont	Decrease text font size
	increasearrowhead	Increase arrowhead size
	decreasearrowhead	Decrease arrowhead size
	linecolor	Line color chooser
	fillcolor	Fill color chooser

Button	Value	Name
	strikethrough	Strike through text (Version Standard 3.0.332+, Professional 1.0.332+)
	transparent	Make text semi-transparent
	and help texthyperlink	Hyperlink dialog box
	rotater	Rotate clockwise
	rotatel	Rotate counterclockwise
	angleflip	Flip angle (changes between obtuse & acute) (V3-only)
	behind	Move behind
	delete	Delete (delete key is still enabled)

If you disable all buttons, the context-sensitive toolbar is hidden.

The default is to display all buttons.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationHideContextButtons" value="fillcolor, hyperlink" />
```

annotationToolBarCollapsed parameter:

This parameter overrides the user preference for the default state of the annotation toolbar.

When this parameter is set to true, the toolbar always starts in a collapsed (folded) state.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.168 or later
- IBM Daeja ViewONE Professional V1.1.168 or later
- ViewONE Annotations Module

Example

```
<param name="annotationToolBarCollapsed" value="true" />
```

hideAnnotationToolbar parameter:

This parameter determines whether the annotation toolbar is hidden, but users can still create or edit annotations.

Typically, if the annotation toolbar is not visible, users cannot create or edit annotations. You might use this parameter, for example, if you want to allow users to edit existing annotations while preventing them from creating new annotations. To implement this scenario, you set the **annotateEdit** HTML parameter and this **hideAnnotationToolbar** parameter to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="hideAnnotationToolbar" value="true/false" />
```

Related reference:

“**annotateEdit** parameter” on page 78

This parameter determines whether the **Annotations** toolbar is available to users.

prefOverrideAnnotationToolbar parameter:

The parameter overrides the user preference for showing the annotation toolbar.

Ordinarily the toolbar is hidden by default, and after the user shows the toolbar by clicking **Show Annotation toolbar**, it is reshown at the start of the user's next session.

Setting this parameter to true forces the toolbar to be visible for all sessions, irrespective of how the user last left it.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="prefOverrideAnnotationToolbar" value="true" />
```

Tooltips

You can use the tooltip parameters to determine the default tooltips for annotations.

alwaysShowCustomAnnotationToolTip parameter:

This parameter determines whether annotation tooltips are always replaced with the tooltip that is defined by the **customAnnotationToolTip** parameter.

If this parameter is set to true, IBM Daeja ViewONE always uses the tooltip that is defined by the **customAnnotationToolTip** parameter for an annotation.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.86 or later
- IBM Daeja ViewONE Professional V1.1.86 or later
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="alwaysShowCustomAnnotationToolTip" value="true" />
```

Related reference:

“**customAnnotationToolTip** parameter”

This parameter determines the default tooltips for annotations.

customAnnotationToolTip parameter:

This parameter determines the default tooltips for annotations.

If no tooltip is defined for an annotation, IBM Daeja ViewONE uses the text in this parameter as the tooltip.

You can include the following replaceable tokens in the tooltip.

Token	Description
<user>	The user ID of the owner of the annotation.
<dateonly>	The date that the annotation was last modified. This date does not include time.
<date>	The date and time that the annotation was last modified,
<creator>	The user ID of the creator of the annotation
<createdateonly>	The date that the annotation was created. This date does not include time.
<createdate>	The date and time that the annotation was created.
<tooltip>	The tooltip that is defined for the annotation. This token is empty unless a tooltip is for the annotation and the HTML parameter alwaysShowCustomAnnotationToolTip is set to true.

The default value for this parameter is an empty string.

Requirements

- IBM Daeja ViewONE Standard V3.1.86 or later
- IBM Daeja ViewONE Professional V1.1.86 or later
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
<param name="customAnnotationToolTip" value="someText" />
```

Related reference:

“**alwaysShowCustomAnnotationToolTip** parameter” on page 136

This parameter determines whether annotation tooltips are always replaced with the tooltip that is defined by the **customAnnotationToolTip** parameter.

Document Thumbnails and Index Lists

IBM Daeja ViewONE Professional can show a thumbnail, a text index, or both for either single documents or multiple documents.

A thumbnail or text index are specified by using the HTML parameter **doc<N>** parameter. You can use other parameters to customize the thumbnail or text index.

Related reference:

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

defaultThumbsDisplayMode parameter

This parameter determines the default thumbnail display mode for IBM Daeja ViewONE.

You can set this parameter to the values listed in the following table.

Value	Description
text	IBM Daeja ViewONE defaults to the text index view for pages in a document, if enabled.
image	IBM Daeja ViewONE defaults to the thumbnail view for pages, if enabled.
docText	IBM Daeja ViewONE defaults to the document text index view, if enabled.
docImage	IBM Daeja ViewONE defaults to the document thumbnail view, if enabled.

If a view is not enabled, no thumbnails are displayed when you first load the document.

The default value for this parameter is image.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="defaultThumbsDisplayMode" value="text" />
```

documentIndex<N> parameter

This parameter determines the text indexes that are used for documents. For example, you might use *index1* for document 1 and *index2* for document 2.

The text index that you specify for a document appears in the document text index list.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="index1" value="Click for Page 1" />
<param name="index2" value="Click for Page 2" />
<param name="index3" value="Click for Page 3" />
```

documentIndexList parameter

This parameter determines the file that contains the replacement text for the page indexes.

The file that you specify must be a simple text file. Use a new line for each page index:

```
index1=Click for Page 1
index2=Click for Page 2
```

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="documentIndexList" value="http://mysite/myindex.txt" />
```

docThumb<N> parameter

This parameter determines whether a separate thumbnail file is specified for each document. Having separate files for document thumbnails can be advantageous because the smaller files are quicker to download and view.

Important: You must use this parameter with the **doc<N>** parameter and **docThumbsEnabled** parameter. In addition, you must specify a file for all pages of the document. The pages must match the corresponding full-resolution document files that are specified by using the **doc<N>** parameter.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="docThumb1" value="images/doc1-t.tif" />
<param name="docThumb2" value="images/doc2-t.tif" />
<param name="docThumb3" value="images/doc3-t.tif" />
```

Related reference:

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

“**docThumbsEnabled** parameter” on page 140

This parameter determines whether the thumbnail views include a list of available documents and a thumbnail of the first page of each document in the document list.

docThumbsEnabled parameter

This parameter determines whether the thumbnail views include a list of available documents and a thumbnail of the first page of each document in the document list.

For the two thumbnail views to be enabled, you must set this parameter to true and use the **doc<N>** parameter.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="docThumbsEnabled" value="true" />
```

Related reference:

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

docThumbsImageEnabled parameter

This parameter determines whether the document thumbnail view is enabled.

The **docThumbsImageEnabled** parameter is only valid when the **docThumbsEnabled** parameter is set to true.

If you set this parameter to false, the viewer removes the icon for the document thumbnail view from the thumbnail toolbar. If the viewer removes all thumbnail views from the thumbnail toolbar, the toolbar is no longer displayed.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="docThumbsImageEnabled" value="false" />
```

Related reference:

“**docThumbsEnabled** parameter”

This parameter determines whether the thumbnail views include a list of available documents and a thumbnail of the first page of each document in the document list.

docThumbLabel <N> parameter

This parameter determines the label for each document thumbnail. The label appears in the corner of the corresponding thumbnail.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

The following example sets the label for the first document thumbnail to My first page.

```
<param name="docThumbLabel1" value="My first page" />
<param name="docThumbLabel2" value="My second page" />
<param name="docThumbLabel3" value="My third page " />
```

docThumbLabelColor<N> parameter

This parameter determines the colors of the document thumbnail labels.

You can specify the label colors by using comma-delimited RGB values or by specifying the values from *IBM Daeja ViewONE color scheme text*. If you do not specify the colors, the document thumbnail labels are white.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="docThumbLabelColor1" value="255,255,192" />
<param name="docThumbLabelColor2" value="255, 192, 255" />
<param name="docThumbLabelColor3" value="192,255, 255" />
```

Related reference:

"IBM Daeja ViewONE color scheme text" on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

docThumbsTextEnabled parameter

This parameter determines whether the document text index list view is enabled.

If you set this parameter to false, the viewer removes the icon for the document text index list view from the thumbnail toolbar. If the viewer removes all thumbnail views from the thumbnail toolbar, the toolbar is no longer displayed.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="docThumbsTextEnabled" value="false" />
```

hideThumbnailCloseIcon parameter

This button determines whether the **Hide Index** button is hidden.

The default value for this parameter is false, which displays the button.

Requirements

- IBM Daeja ViewONE Professional V1.1.16 or later
- No additional modules

Example

```
<param name="hideThumbnailCloseIcon" value="true" />
```

index<N> parameter

This parameter determines the text index that is used for a page in a multiple-page document. The specified value appears in the text index list.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="index1" value="Click for Page 1" />  
<param name="index2" value="Click for Page 2" />  
<param name="index3" value="Click for Page 3" />
```

indexList parameter

This parameter specifies a file that contains the replacement text that is displayed in the thumbnail pane for the pages. This method is an alternative to the **index<N>** parameter.

You must specify a simple text file that uses line feeds to delimit successive page indexes.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="indexList" value="http://mysite/myindex.txt" />
```

prefOverrideThumbsDisplayMode parameter

This parameter determines whether the **defaultThumbsDisplayMode** parameter value is always used for the thumbnail display mode.

When this parameter is set to false, the value that is stored in the preferences file is used to determine the thumbs display mode.

If the display mode is never changed, then the **defaultThumbsDisplayMode** value is used.

When this parameter is set to true, the value that is stored in the preferences file is ignored and the **defaultThumbsDisplayMode** value is always used.

The default value is false.

Requirements

- IBM Daeja ViewONE Professional V1.0.176 or later
- No additional modules

Example

```
<param name="prefOverrideThumbsDisplayMode" value="true" />
```

Related reference:

“**defaultThumbsDisplayMode** parameter” on page 138

This parameter determines the default thumbnail display mode for IBM Daeja ViewONE.

thumbsImageEnabled parameter

This parameter determines whether the page thumbnail view and the thumbnail icon on the thumbnail toolbar are displayed.

Important: If all available thumbnail views are removed from the thumbnail toolbar, the toolbar is no longer displayed.

When this parameter is set to false, the page thumbnail view and the thumbnail icon are not displayed.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="thumbsImageEnabled" value="false" />
```

thumbsTextEnabled parameter

This parameter determines whether the thumbnail text index list view and the text index list icon are displayed.

Important: If all available thumbnail views are removed from the thumbnail toolbar, the toolbar is no longer displayed.

When this parameter is set to false, the thumbnail text index list view and the text index list icon are not displayed.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="thumbsTextEnabled" value="false" />
```

thumbsTextModeAllowed parameter

This parameter determines whether the horizontal bar in the thumbnail toolbar that contains icons for text list, page or document thumbnails, or the PDF search is displayed.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.0.490 or later
- No additional modules

Example

```
<param name="thumbsTextModeAllowed" value="false" />
```

thumbsText parameter

This parameter determines whether the thumbnails pane allows a text index to be displayed.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thumbsText" value="false" />
```

Font Substitution options

With the **blockEmailModule** parameter, the viewer is configured to make font substitutions to display a Microsoft Word or Microsoft PowerPoint document.

This substitution is sometimes necessary where the document requires a particular font to render the document but this font is not available on the client workstation. In this case, the viewer can make a "closest match" font substitution to display the document.

The following parameters are configurable for font substitution if applicable. Where font substitution occurs, it can mean that there is a change to the layout of the document. See *Annotations and "flow layout" format documents* for further detail on layouts. So allowing of font substitutions must be considered carefully where annotations are used that cover sensitive information on documents since they can lead to a potential security risk.

Related concepts:

 [Annotations and "flow layout" format documents](#)

Related reference:

["blockEmailModule" parameter](#) on page 188

This parameter determines whether email can be viewed.

allowDocumentFontSubstitutions parameter

This parameter determines whether IBM Daeja ViewONE substitutes fonts for specific document types that use system fonts.

If this parameter is set to true, IBM Daeja ViewONE substitutes fonts where necessary and always shows the entire document.

If this parameter is set to false, IBM Daeja ViewONE does not substitute fonts. Instead, IBM Daeja ViewONE truncates the document at the page before the first substitution would be made.

The default value for this parameter is true.

Important: The **allowDocumentFontSubstitutions** parameter affects the content of the warning message that is set by the **enableFontSubstitutionWarning** parameter. Whether you allowed font substitution is displayed in the message.

Requirements

- IBM Daeja ViewONE Standard V4.0.48 or later
- IBM Daeja ViewONE Professional V4.0.48 or later
- ViewONE Office Module

Example

```
<param name="allowDocumentFontSubstitutions" value="<N>" />
```

Related reference:

“**enableFontSubstitutionWarning** parameter”

This parameter determines whether a warning message is displayed when the viewer needs to substitute fonts in a document of a type that uses system fonts.

enableFontSubstitutionWarning parameter

This parameter determines whether a warning message is displayed when the viewer needs to substitute fonts in a document of a type that uses system fonts.

Set this parameter to true to display a dialog box that prompts the user for how to proceed and render the document.

This warning message that is displayed depending on the **allowDocumentFontSubstitutions** parameter. If the **allowDocumentFontSubstitutions** parameter is set to true, the message states that substitutions were made. If this parameter is set to false, the message states that it truncated the document at the page before the first substitution was made.

The default value for the **enableFontSubstitutionWarning** parameter is false.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- ViewONE Office Module

Example

```
<param name="enableFontSubstitutionWarning" value="true" />
```

Related reference:

“**allowDocumentFontSubstitutions** parameter” on page 144

This parameter determines whether IBM Daeja ViewONE substitutes fonts for specific document types that use system fonts.

Graphics

The graphics parameters display and configure graphics options.

activeButtons parameter

Active buttons are buttons and scroll bars that highlight as the mouse is moved over them. If set to true, buttons and scroll bars are highlighted in a darker color.

To disable this function, set the parameter to false.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="activeButtons" value="false" />
```

activeColor parameter

This parameter determines the background color for active scroll bars and thumbnails.

The value specifies the red, green, and blue intensities (RGB), in that order. You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="activeColor" value="223, 223, 223" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

animateTransforms parameter

This parameter determines whether the rotation and zooming animations in the viewer are disabled. You might want to disable the animations for accessibility.

The default for this parameter is true, which enables the rotation and zooming animations in the viewer.

Requirements

- IBM Daeja ViewONE Virtual V4.1.0 or later
- No additional modules

Example

```
<param name="animateTransforms" value="false" />
```

backColor parameter

This parameter specifies the background color for the viewer. The value represents the red, green, and blue intensities, in that order (RGB).

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

This color is used for UI elements such as the background of buttons, toolbars, scrollbars, and windows. You can use this parameter to integrate IBM Daeja ViewONE into a web page that does not match the system default color.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="backColor" value="192,192,192" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

bar<N>button<N> parameter

This parameter specifies custom buttons to insert into any of the top, bottom, annotations, or context-sensitive toolbars. The custom buttons initiate calls to JavaScript methods that are embedded in your HTML page.

For this parameter to function correctly, you must include the **mayscript** property in your applet tag.

Buttons are added to the left side of the specified toolbar.

The parameter value is a comma-delimited sequence of strings as follows:

```
value="scriptN(), helpN, enabledN.jpg, disabledN.jpg,  
      true|false"
```

scriptN()

This script is the name of any JavaScript function that you want IBM Daeja ViewONE to call when the user clicks the button. You can include one parameter in this script name. For example, `script1(1234)`. The script name must be in lowercase.

helpN This parameter is the help text that is used in the button tooltip and status bar text.

enabledN.jpg

This parameter is the file name for the image that is seen when the button is enabled. This file must exist in the location indicated by your codebase parameter and the image must measure 22 by 19 pixels.

disabledN.jpg

This parameter is the file name for the image that is seen when the button is disabled. This file must exist in the location indicated by your codebase parameter and the image must measure 22 by 19 pixels.

true / false

When this parameter is set to true, the button is enabled all the time. When it is set to false, the button is enabled only when a document is open.

evalscriptN()

Important: This parameter allows for an optional extra script to be used to evaluate the button state.

The `evalscriptN()` script is the name of any additional JavaScript function that you want IBM Daeja ViewONE to call when the user clicks the button.

You can include one parameter in this script name, for example, `script1(1234)`. The script name must be in lowercase.

The function returns a bitmask value that is used control the display state of the button itself. The following table contains the values.

Return Value	Visible	Depressed	Enabled
0	No	No	No
1	Yes	No	No
2	No	Yes	No
3	Yes	Yes	No
4	No	No	Yes
5	Yes	No	Yes
6	No	Yes	Yes
7	Yes	Yes	Yes

Important:

- Unlike the rest of the comma limited sequence (all of which is mandatory), this last function is optional and can be left off entirely if you prefer.
- Button images can be JPEGs or GIFs and can have transparent areas. Set any area of the image that you want to be the background to transparent so that IBM Daeja ViewONE can display them with a matching background to the rest of the viewer.

Important: This function includes **bar3** (annotations toolbar) and **bar4** (annotations context-sensitive toolbar)

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
<param name="bar1button1" value="script1(), help1, enabled1.jpg, disabled1.jpg, true" />
<param name="bar2button1" value="script2(), help2, enabled2.jpg, disabled2.jpg, true" />
<param name="bar3button1" value="script1(), help1, enabled1.jpg, disabled1.jpg, true, eval1()" />
<param name="bar4button1" value="script2(), help2, enabled2.jpg, disabled2.jpg, true, eval2()" />
```

border parameter

This parameter determines whether the viewer has a border.

When this parameter is set to true, the border is displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
<param name="border" value="false" />
```

customResource<N> parameter

This parameter supplies multiple image files for custom buttons within a single resource file.

By using this parameter, and ensuring all images use for custom buttons are included within a single resource file. You ensure optimum start performance (because IBM Daeja ViewONE does not then need to complete multiple retrievals from the server).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="customResource1" value="mybuttonimages.zip" />
```

extraRetrievalMessages parameter

This parameter determines whether the viewer displays progress messages when the viewer connects and retrieves files from a server.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.384 or later
- IBM Daeja ViewONE Professional V1.0.384 or later
- No additional modules

Example

```
<param name="extraRetrievalMessages" value="false" />
```

highlightColor parameter

This parameter determines the color of the zoom area highlight.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The value represents the red, green, and blue intensities, in that order (RGB). If the value *none* is used instead of a value, then the zoom area highlight is transparent.

The default value is 255, 255, 0 (yellow).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="highlightColor" value="252, 145, 66" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

imageDecodeTextDelay parameter

This parameter determines the time in milliseconds that the viewer waits to display a message for a length image decode operation. Use this parameter to see the status of operations that are taking a long time to complete.

If the image is decoded within this time, no message is displayed. The default value is 500 milliseconds.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageDecodeTextDelay" value="2000" />
```

imageToolbarCollapsed parameter

This parameter determines whether the top image toolbar is expanded or collapsed.

If this parameter is set to true, the toolbar always starts in the collapsed state.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageToolbarCollapsed" value="true" />
```

retrievalTextDelay parameter

This parameter specifies the time in milliseconds that IBM Daeja ViewONE waits until it displays the retrieval delay message.

If the image is retrieved within this time, no message is displayed.

The default value is 500 milliseconds.

When this parameter is set to 0, the message is always displayed.

This parameter is valid only if the **extraRetrievalMessages** parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="retrievalTextDelay" value="2000" />
```

Related reference:

“**extraRetrievalMessages** parameter” on page 149

This parameter determines whether the viewer displays progress messages when the viewer connects and retrieves files from a server.

showHotKeys parameter

This parameter determines whether the keyboard shortcut text is displayed in the menus.

When this parameter is set to true, the keyboard shortcut text is displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="showHotKeys" value="false" />
```

splitBarAllowed parameter

This parameter determines whether the split-bar that separates thumbnails from the main viewing area is displayed.

When this parameter is set to true, the split-bar that is used to separate thumbnails from the main viewing area is displayed. Disabling the split-bar prevents the user from changing the width of the thumbnails pane.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="splitBarAllowed" value="false" />
```

statusBar parameter

This parameter determines whether the status bar is displayed.

When this parameter is set to true, the status bar is displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="statusBar" value="false" />
```

theme parameter

This parameter defines the look (UI) of the IBM Daeja ViewONE viewer.

IBM Daeja ViewONE Standard and IBM Daeja ViewONE Professional

There are currently two themes available:

Modern

This theme represents the new look that was created for Version 4 of IBM Daeja ViewONE. It has a new color scheme and a new set of buttons. This theme is the default theme.

Classic

This theme represents the look that IBM Daeja ViewONE had before Version 4.

IBM Daeja ViewONE Virtual

There are currently three themes available:

Modern

This theme represents the new look that was created for Version 4 of IBM Daeja ViewONE. It has a new color scheme and a new set of buttons. This theme is the default theme for a desktop service.

White This theme has a predominantly white background and layout. This theme is the default theme for a mobile device.

Black This theme has a predominantly black background and layout.

Requirements

- IBM Daeja ViewONE Standard V4.0.2 or later
- IBM Daeja ViewONE Professional V4.0.2 or later
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="theme" value="classic" />
```

toolbarCollapseDelay parameter

This parameter is used to change the default delay when a toolbar is auto-collapsed (following an auto-expand).

The parameter specifies the number of milliseconds for the delay.

The default value for this parameter is 1000 milliseconds (that is, 1 second).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

In the following example, auto-collapse is set to 2 seconds.

```
<param name="toolbarCollapseDelay" value="2000" />
```

toolbarExpandAllowed parameter

This parameter disables the collapse-and-expand feature for toolbars.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="toolbarExpandAllowed" value="false" />
```

toolbarExpandDelay parameter

This parameter is used to change the default delay when a toolbar is auto-expanded.

A toolbar is auto-expanded if it is first collapsed (by clicking **collapse**) and then the mouse pointer is hovered over the toolbar in a collapsed state. The parameter specifies the number of milliseconds for the delay before IBM Daeja ViewONE auto-expands the toolbar.

The default value for this parameter is 300 milliseconds (1/3 of a second).

Important: Auto-expand is not available for Macintosh users due to problems with the Macintosh Java rendering, but collapsing and expanding is available by clicking **expand** and **collapse** as usual.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

In the following example, the toolbar is expanded after 1 second.

```
<param name="toolbarExpandDelay" value="1000" />
```

toolTipsBackground parameter

This parameter can be used to change the background color of the tooltips.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="toolTipsBackground" value="0,0,192" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

toolTipsForeground parameter

This parameter can be used to change the foreground color of the tooltips text and border.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="toolTipsForeground" value="255,255,255" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

toolTipsMaxLength parameter

This parameter limits the size of tooltips. The default value is set to 256 characters.

This limitation helps prevent large tooltips such as those associated with Sticky Notes[®] from misdrawing or occupying too much of the screen.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="toolTipsMaxLength" value="64" />
```

toolTips parameter

When set to false, this parameter disables tooltips.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="toolTips" value="false" />
```

viewToolBarCollapsed parameter

This parameter overrides the user preference for the default state of the bottom view toolbar.

When this parameter is set to true, the toolbar always starts in a collapsed state.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="viewToolBarCollapsed" value="true" />
```

Hyperlinks

Use the following parameters to configure hyperlinks.

hyperlink<N> parameter

This parameter specifies the hyperlink that is to be used to access a specific page in a multiple-page document.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

The following example specifies the hyperlink as a file name that is relative to the codebase:

```
<param name="hyperlink1" value="newsite1.html" />
<param name="hyperlink2" value="newsite2.html" />
```

The following example specifies the full URL for the hyperlink:

```
<param name="hyperlink1" value="http://mysite/newsite1.html" />
```

hyperlinkDoubleClick parameter

This parameter determines whether a user must double-click an image to activate a hyperlink.

This parameter also affects how the **openOnClick** parameter works.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="hyperlinkDbClick" value="false" />
```

Related reference:

“**openOnClick** parameter”

This parameter determines whether IBM Daeja ViewONE redirects the browser to open the specified page when the user clicks the image area.

hyperlink parameter

This parameter specifies a hyperlink to a cross-referenced document that is activated when the user clicks the image area.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

You can specify the hyperlink either as a file name relative to the codebase or by providing the full URL as shown in the following examples:

```
<param name="hyperlink" value="newsite.html" />  
<param name="hyperlink" value="http://mysite/newsite.html " />
```

openOnClick parameter

This parameter determines whether IBM Daeja ViewONE redirects the browser to open the specified page when the user clicks the image area.

This parameter is similar to the **hyperlink** parameter. However, this parameter includes extra options.

The value of this parameter can be a web page, or any server-side script, object, or executable that generates the HTML for a web page.

If the **hyperlinkDbClick** parameter is set to true, the link is followed only after the user double-clicks the image. The default is a single-click operation.

If you include any of the following optional identifiers, then IBM Daeja ViewONE substitutes them as follows:

- <f> Replaced by the current page file name or URL as appropriate.
- <fo> Replaced by the current page file name only, not including path.
- <x> The "X" coordinate on the image, where the user clicked with the mouse.
- <y> The "Y" coordinate on the image, where the user clicked with the mouse.
- <r> The rotation of the image display, when the user clicked with the mouse.
- <z> The zoom factor of the image display, when the user clicked with the mouse.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="openOnClick" value="http://myserver/myopenscript.cgi?file=<f>" />
```

Related reference:

“**hyperlinkDbClick** parameter” on page 155

This parameter determines whether a user must double-click an image to activate a hyperlink.

openOnClickKey parameter

This parameter determines whether IBM Daeja ViewONE redirects the browser when a specific key is pressed while you click the image.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="openOnClickKey" value="g" />
```

openOnClickTarget parameter

This parameter determines how IBM Daeja ViewONE redirects the browser.

This parameter can have one of the following values.

_blank

Opens in a new window.

_self Opens in the same window/frame that contains IBM Daeja ViewONE.

_top Opens in the top-level frame of the IBM Daeja ViewONE window. If frames are not used, this value has the same effect as **_self**.

_parent

Opens in the IBM Daeja ViewONE parent frame. If frames are not used, this value has the same effect as **_self**.

name Opens in a window that is named **name**. If a window with this name does not exist, then a top-level window is created with this name.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="openOnClickTarget" value="_blank" />
```

Image options

Use the following parameters to enable and configure image settings.

adjustBars parameter

This parameter determines whether the adjustment bars are displayed in the status bar.

If you set this parameter to false, the adjustment bars are removed from the status bar. However, the user can still use the menu options to change image parameters.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="adjustBars" value="false" />
```

adjustControls1 parameter

This parameter determines the adjustment controls and menus that are enabled for monochrome images (1-bit black and white).

The three Boolean values for this parameter represent the brightness control, the contrast control, and the luminance control.

The default value for this parameter is true, false, false, which enables only the brightness control.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="adjustControls1" value="true, false, false" />
```

adjustControls4 parameter

This parameter determines which of the adjustment controls and menus are enabled for 4-bit color images (16 colors).

The three Boolean values for this parameter represent the brightness control, the contrast control, and the luminance control.

The default value for this parameter is false, false, true, which enables only the luminance control.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="adjustControls4" value="false, false, true" />
```

adjustControls8 parameter

This parameter determines which of the adjustment controls and menus are enabled for 8-bit color images (256 colors).

The three Boolean values for this parameter represent the brightness control, the contrast control, and the luminance control.

The default values for this parameter is false, false, true, which enables only the luminance control.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="adjustControls8" value="false, false, true" />
```

adjustControls24 parameter

This parameter determines which of the adjustment controls and menus are enabled for 24-bit color images (16 million colors).

The three Boolean values for this parameter represent the brightness control, the contrast control, and the luminance control.

The default value for this parameter is true, true, true, which enables all three controls.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="adjustControls24" value="true, true, true" />
```

adjustOptions parameter

This parameter determines whether the image adjustment controls and menus are available to users.

By default, IBM Daeja ViewONE displays the following controls

- The brightness control for monochrome images (1-bit black and white)
- The brightness control, contrast control, and luminance control for color images.

You can change the display of individual controls by using the following parameters: **adjustControls1**, **adjustControls4**, **adjustControls8**, or **adjustControls24**.

The default value for the **adjustOptions** parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
<param name="adjustOptions" value="false" />
```

allowHeaderTags parameter

This parameter determines whether IBM Daeja ViewONE automatically rotates images to their original orientation when they are opened based on tags that are included in a TIFF file.

A TIFF file can contain standard header tags that indicate whether an image was rotated when the image was saved. By default, IBM Daeja ViewONE checks for these tags and automatically rotates the image to the original orientation when the file is opened. The user can then view the image without selecting any of the IBM Daeja ViewONE rotation buttons or menu options.

If the TIFF writer entered the header tag in error, IBM Daeja ViewONE might rotate images unnecessarily. Because it is not always convenient to rewrite the image file without the tag, you can set the **allowHeaderTags** parameter to false to prevent IBM Daeja ViewONE from rotating images automatically.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="allowHeaderTags" value="true" />
```

allowHeaderTagsOnPrint (rotation) parameter

This parameter determines whether the value set for the **allowHeaderTags** HTML parameter is applied when an image is printed.

By default, IBM Daeja ViewONE uses the TIFF header tag settings as defined by the **allowHeaderTags** parameter when an image is printed. You override this behavior by using the **allowHeaderTagsOnPrint** parameter.

Requirements

- IBM Daeja ViewONE Standard V3.1.150 or later
- IBM Daeja ViewONE Professional V1.1.150 or later
- No additional modules

Example

```
<param name="allowHeaderTagsOnPrint" value="false" />
```

allowResolutionTags parameter

This parameter determines whether IBM Daeja ViewONE is to process the X and Y resolution header tags that are found a TIFF image.

If this parameter is set to false, the X and Y resolution header tags are not processed. Instead, IBM Daeja ViewONE uses a resolution ratio of 1:1. You might set this parameter to false if your library contains images in which the X and Y resolution header tags are incorrect.

The default value for this parameter is true.

Important: Some images have X and Y resolutions that are intentionally different. If the resolution values are ignored for these images, the images use an incorrect aspect and appear stretched or compressed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="allowResolutionTags" value="true" />
```

areaZoom parameter

This parameter specifies the initial mode for the mouse pointer.

When this parameter is set to false, dragging the mouse pans the image. The default value is false.

When this parameter is set to true, dragging the mouse zooms the selected area of the image, as if the user selected **Zoom Area**.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="areaZoom" value="true" />
```

areaZoomSticky parameter

This parameter determines whether the cursor remains in zoom area mode or returns to dragging mode after an area is zoomed.

When this parameter is set to false, the cursor mode returns to dragging after the user zooms into an area. The default value is false.

When this parameter is set to true the cursor remains in the area zooming mode after the user zooms into an area.

Restriction: If the **areaZoom** parameter is set to true, then this parameter cannot be used.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="areaZoomSticky" value="false" />
```

autoEnhanceBlack parameter

This parameter specifies the image reduction level at which the viewer switches to preserve black anti-aliasing mode to compensate for loss of detail.

When the viewer's enhance mode is set to "auto select" (signified by a default value of "+" in the **enhance** icon), IBM Daeja ViewONE decides what level of anti-aliasing is used when images are displayed. When images are reduced so that they can fit on the screen, fine detail can be lost. The viewer switches to the preserve black anti-aliasing mode to compensate for loss of detail.

The default point at which the switch is made is when the image is reduced by 16 times. You can change this threshold zomm level.

For example, if you want to display all images with preserve black, set this parameter value to "0".

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="autoEnhanceBlack" value="10" />
```

brightness parameter

This parameter sets the default brightness, for all pages, of the display area.

The value represents a percentage 0 - 100, with 50 being the default value. 0 = minimum brightness (dark) and 100 = maximum brightness (light).

If brightness is set to 0, then the viewer starts an effect that is known as preserve black. With preserve black, any black pixel on the source image results in a black pixel on the screen. Without this effect graduated gray pixels might be used due to anti-aliasing.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="brightness" value="65" />
```

contrast parameter

This parameter sets the default contrast of the display area on all pages. The value represents a percentage 0-100, with 50 being the default value.

The minimum contrast is 0 (flat) and 100 is maximum brightness (not flat).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="contrast" value="40" />
```

defaultTransformDoc (previously defaultRotateDoc) parameter

This parameter determines whether the **rotate**, **flip**, and **invert** menus and keys apply to all pages in the current document.

If this parameter is set to true, **rotate**, **flip**, and **invert** operations apply to all pages in the current document.

The default value for this parameter is false, which applies the operations only to the current page.

Requirements

- IBM Daeja ViewONE Standard V3.0.498 or later
- IBM Daeja ViewONE Professional V1.0.498 or later
- No additional modules

Example

```
<param name="defaultTransformDoc" value="false" />
```

enableSharpEnhanceModes parameter

This parameter determines whether users can select the **Sharp** and **Sharper** enhance modes.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.1.66 or later
- IBM Daeja ViewONE Professional V1.1.66 or later
- No additional modules

Example

```
<param name="enableSharpEnhanceModes" value="false" />
```

enhanceButton parameter

This parameter determines whether the **image enhance** button is displayed.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="enhanceButton" value="true" />
```

enhanceColor parameter

This parameter determines whether a 24-bit color image is displayed with anti-aliasing on.

Setting this parameter to true turns on anti-aliasing, and is the same as omitting the parameter. The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="enhanceColor" value="true" />
```

enhanceMenu parameter

This parameter determines whether the **Image Enhance** menu is displayed when the user right-clicks in the viewer.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="enhanceMenu" value="true" />
```

enhanceMode parameter

This parameter determines the default enhance (anti-aliasing) level.

You can specify the following values for this parameter.

Value	Description
0	Automatic
1	Level 1 for typical images
2	Level 2 for drawings
3	Level 3 for very high-resolution drawings
4	Level 4 for auto sharp
5	Level 5 for auto sharper

The default value for this parameter is 0.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="enhanceMode" value="3" />
```

enhance parameter

This parameter determines whether a monochrome image is displayed with anti-aliasing on.

The default value for this parameter is true; monochrome images are anti-aliased.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="enhance" value="true" />
```

flipOptions parameter

This parameter determines whether the buttons and menu options are available to enable users to flip the visible page horizontally and vertically.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="flipOptions" value="true" />
```

flip parameter

This parameter determines the default flip mode that is used to display pages.

You can enter the following values for this parameter:

Value	Flip mode
0	None
1	Horizontal or mirror
2	Vertical
3	Horizontal and vertical

The default value is 0.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="flip" value="3" />
```

imageBackColor parameter

This parameter determines the default color that is used for the background of monochrome images.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 255, 255, 255 (white).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageBackColor" value="255,255,255" />
```

Related reference:

"IBM Daeja ViewONE color scheme text" on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

imageButtons parameter

This parameter determines whether certain image buttons are shown in the viewer.

If this parameter is set to true, the following image buttons are shown in the viewer:

- **Best fit**
- **Fit to height**
- **Fit to width**
- **Rotate clockwise**
- **Rotate counterclockwise**
- **Rotate 180 degrees**
- **Zoom area**
- **Zoom in**
- **Zoom out**

You can use the **visibleImageButtons** parameter to show or hide these image buttons individually.

The default value for this parameter is true.

To show or hide the remaining image buttons, use the following parameters:

Button	Parameter
Flip Horizontal	flipOptions
Flip Vertical	flipOptions
Invert	invertButton
Magnify	magOptions
Zoom 100%	zoom100Options

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="imageButtons" value="false" />  
<param name="imageButtons" value="true" />
```

imageForeColor parameter

This parameter determines the default color that is used for the text of monochrome images.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

The default value for this parameter is 0, 0, 0 (black).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageForeColor" value="0,0,0" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

imageKeys parameter

This parameter determines whether the keyboard shortcuts for image operations are enabled.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageKeys" value="false" />
```

Related reference:

 Accessibility features of IBM Daeja ViewONE

imageMenus parameter

This parameter determines whether the image menus are available when a user right-clicks in viewer.

The default for this parameter is true, which enables the menus. You can control the individual image menus with the **visibleImageMenus** parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="imageMenus" value="false" />
```

infoOptions parameter

This parameter specifies whether the **Image Properties** button and menus are displayed.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="infoOptions" value="true" />
```

infoOptionsShowSizeCm parameter

This parameter determines whether image width and height are given in inches or centimeters in the Image Information dialog box.

The default value for this parameter is false, which displays size in inches.

Requirements

- IBM Daeja ViewONE Standard V3.0.716 or later
- IBM Daeja ViewONE Professional V1.0.716 or later
- No additional modules

Example

```
<param name="infoOptionsShowSize" value="true" />
```

infoOptionsShowSize parameter

This parameter determines whether the image width and height are displayed the Image Information dialog box.

The default value for this parameter is true, which displays the width and height. These values are displayed in either inches or centimeters depending on the value of the **infoOptionsShowSizeCm** parameter.

Requirements

- IBM Daeja ViewONE Standard V3.0.716 or later
- IBM Daeja ViewONE Professional V1.0.716 or later
- No additional modules

Example

```
<param name="infoOptionsShowSize" value="false" />
```

Related reference:

“**infoOptionsShowSizeCm** parameter” on page 168

This parameter determines whether image width and height are given in inches or centimeters in the Image Information dialog box.

invertButtons parameter

This parameter determines whether the **Invert** button is available in the viewer.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="invertButtons" value="false" />
```

invertKeys parameter

This parameter determines whether the keyboard shortcut for the image invert operation is enabled.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="invertKeys" value="false" />
```

Related reference:

 Accessibility features of IBM Daeja ViewONE

invertMenus parameter

This option allows separate control over the invert menu.

Important: To see an invert menu, the image menus must first be enabled.

The default value for this parameter is true. If the parameter is set to true, the menu is displayed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="invertMenus" value="false" />
```

invert parameter

This parameter determines whether the viewer inverts image colors to negative colors by default when pages are displayed.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="invert" value="true" />
```

Luminance parameter

This parameter sets the default luminance of the display area for all pages.

The value represents a percentage 0-100, where 0 is minimum luminance, and 100 is maximum brightness. The default value is 50.

Brightness increases the brightness of all colors (dark and light) uniformly. Luminance causes already bright areas of the image to increase in brightness further and darker areas to increase in brightness also, but by a lesser amount. Luminance simulates a light source that is shining on the image and can be more effective at making color images clearer to read.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="luminance" value="60" />
```

magOptions parameter

This parameter determines whether the magnifier, its keyboard shortcuts, and the menu option are enabled.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
<param name="magOptions" value="true" />
```

pageRotation/ parameter

This parameter specifies the angle at which each page of a multipage image is displayed by default.

Values of 0, 90, 180, or 270 are accepted. The default value is 0.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="pageRotation1" value="90" />
<param name="pageRotation2" value="180" />
<param name="pageRotation3" value="270" />
```

Pages must be contiguous and start from the first page. That is, you cannot set page 1 and page 3 without setting page 2 but you can set page 1 and page 2 without setting page 3. For example, if you want page 2 in the example to not be rotated, use the following HTML:

```
<param name="pageRotation1" value="90" />
<param name="pageRotation2" value="0" />
<param name="pageRotation3" value="270" />
```

persistViewProcessingProperties parameter

This parameter determines whether the brightness, contrast, and luminance settings that were defined by the user are persisted as a preference.

When this parameter is set to true, the user settings are persisted. The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.1.64 or later
- IBM Daeja ViewONE Professional V1.1.64 or later
- No additional modules

Example

```
<param name="persistViewProcessingProperties" value="false" />
```

photometric parameter

The **photometric** HTML parameter determines how IBM Daeja ViewONE processes the **photometric Interpretation** TIFF tag.

Some bilevel (black and white) TIFF image and grayscale TIFF images can have their data stored inverted. In this case, the **photometric Interpretation** TIFF tag value is set to 1 to change the display of the image so that the image displays correctly.

Important: Beginning with Version 4.0.4, the default value of the **photometric** parameter is true so that TIFF images with the **photometric Interpretation** value of 1 have this TIFF tag processed.

To stop IBM Daeja ViewONE from processing this TIFF tag, set the value for **photometric** to false.

The default value for this parameter was previously set to false because in the past some TIFF image creation software did not correlate the **photometric Interpretation** TIFF tag with the image data that they created. Since the data was not correlated, the TIFF tag was set to 1, which meant the image data has colors

that are reversed, but the image data did not comply with this inversion. In the unlikely case that you have TIFF images that have this mismatch between the data, and the **photometric Interpretation** TIFF tag is set to 1, setting this HTML parameter to false might solve the problem.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="photometric" value="false" />
```

rotation parameter

This parameter specifies the angle at which pages are displayed by default.

Values of 90, 180, or 270 are accepted. The default value is 0.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="rotation" value="90" />  
<param name="rotation" value="180" />  
<param name="rotation" value="270" />
```

scale parameter

This parameter specifies the scaling algorithm to use when pages are displayed.

Valid values are as follows:

- best** The page is scaled to fit into the window area so that all of the page is visible.
- ftow** The page is scaled so that the width of the page matches the width of the window area. This scaling might require a vertical scroll bar if the visible page height exceeds the available window height.
- ftoh** The page is scaled so that the height of the page matches the height of the window area. This scaling might require a horizontal scroll bar if the visible page width exceeds the available window width.

The default value is best.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="scale" value="best" />  
<param name="scale" value="ftow" />  
<param name="scale" value="ftoh" />
```

thirdPartyFileCache parameter

This parameter determines whether auto-file caching is used when large PNG images are decompressed.

If sufficient memory is available, then turning off this parameter results in faster decompression performance but it uses more memory.

When this parameter is set to false, auto-file caching is disabled while large PNG images are decompressed.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thirdPartyFileCache" value="false" />
```

visibleImageButtons parameter

The image buttons are visible by default, but this parameter can be used to set those individual buttons that are not controlled by other parameters.

This parameter must include a list of nine values that represent the buttons that are visible, true, or invisible, false.

The image buttons that are affected by this parameter are (in order): **Zoom area**, **Zoom in**, **Zoom out**, **Fit to width**, **Fit to height**, **Best fit**, **Rotate clockwise**, **Rotate counterclockwise**, and **Rotate 180 degrees**.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="visibleImageButtons" value="true, true, true, true, true, true,  
true, true, true" />
```

visibleImageKeys parameter

The image keys are available by default, but this parameter can be used to set those individual keys that are not controlled by other parameters.

This parameter must include a list of ten values that represent that those keys are available, true, or not available, false.

The image keys that are affected by this parameter are (in order): **Zoom area**, **Zoom in**, **Zoom out**, **Fit to width**, **Fit to height**, **Best fit**, **Rotate clockwise**, **Rotate counterclockwise**, **Rotate 180 degrees**, and **Display as grayscale**.

The default for this parameter is true, true, true, true, true, true, true, true, true, true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="visibleImageKeys" value="true, true, true, true, true, true, true, true, true, true, true" />
```

visibleImageMenus parameter

The image menus are visible by default, but this parameter can be used to set those individual menus that are not controlled by other parameters.

This parameter must include a list of nine values that represent menus that are visible, true, or invisible, false.

The image menus that are affected by this parameter are (in order): **Zoom area**, **Zoom in**, **Zoom out**, **Fit to width**, **Fit to height**, **Best fit**, **Rotate clockwise**, **Rotate counterclockwise**, **Rotate 180**, and **Display as grayscale**.

The default for this parameter is true, true, true, true, true, true, true, true, true, true, true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="visibleImageMenus" value="true, true, true, true, true, true, true, true, true, true, true" />
```

zoom100options parameter

This parameter disables **zoom100** and the **zoom100** menu option.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="zoom100options" value="true" />
```

zoom100 parameter

This parameter displays the image at 100% when it is first loaded.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="zoom100" value="true" />
```

JavaScript options

You can use the JavaScript options to prepare the viewer to accept JavaScript methods.

annotationJavascriptExtensions parameter

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

This parameter affects the following JavaScript methods:

- reloadAnnotations()
- setAnnotateEdit(boolean)
- int getNumAnnotations(type, page)
- string getAnnotationLabels(type, page)
- string getAnnotationLabels(label)
- addAnnotation(annotationProperties)
- modifyAnnotation(label, annotationProperties)
- deleteAnnotation(label)
- deleteAllAnnotations(type, page)
- startAnnotation(type)
- setRedactionIsSemiTransparent(boolean)
- setAnnotationsSemiTransparent(boolean, type)

The default value for this parameter is false, which enables the viewer to accept calls for these methods.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="annotationJavascriptExtensions" value="true" />
```

javascriptExtensions parameter

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

JavaScript methods that must be enabled by the **javascriptExtensions** parameter are identified in the *IBM Daeja ViewONE JavaScript Reference Manual*.

eventHandler parameter

This parameter determines the JavaScript method that handles events for the viewer.

The viewer JavaScript methods are asynchronous and do not wait for a previous method to complete before they run. Therefore, the viewer uses an event handler to track events. You can use the event handler to monitor user activity and other selected actions that the IBM Daeja ViewONE viewer performs.

You must set the **eventHandler** parameter to the function name of the JavaScript method that handles the events.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="eventHandler" value="myEventHandler" />
```

Related reference:

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

eventHandlerJson parameter

This parameter determines that JavaScript method that is to handle events. You can use this parameter to extend the standard **eventHandler** parameter to allow a server-side process to return a **JSON** object. This **JSON** object contains a map of parameters that can be parsed. You can specify extra actions to be taken depending on your requirements.

Set this parameter to the function name of the JavaScript method that is to handle the events.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="eventHandlerJson" value="myEventHandler" />
```

Related reference:

“**eventHandler** parameter”

This parameter determines the JavaScript method that handles events for the viewer.

eventInterest parameter

This parameter determine IBM Daeja ViewONE events are sent to the event logger.

This parameter is used with the **eventHandler** parameter. By default, the **eventHandler** parameter sends all events that are generated to the event logger. You can use the **eventInterest** parameter to filter which events that are sent to the event logger.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="eventInterest" value="5,22" />
```

Related reference:

“**eventHandler** parameter” on page 176

This parameter determines the JavaScript method that handles events for the viewer.

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

serverEventHandler parameter

This parameter determines whether event handling is enabled for use with the JavaScript API.

The JavaScript API introduces the concept of JavaScript and server event logging. You can use JavaScript or a server object to monitor user activity and other selected actions that are performed by the IBM Daeja ViewONE applet.

You can enable event handling with this parameter or with the **eventHandler** parameter.

To allow a server script, object, or executable file to receive notification of events as they occur, set this parameter to the URL path that represents the object.

The following example is a Perl script. IBM Daeja ViewONE tags the following text onto the end of the value that is specified: `Id=n`
&description=description+of+event

Where *n* is a number that represents the event and the description contains more textual data (usually page number but it depends on the event id). See the *IBM Daeja ViewONE JavaScript Reference Manual* for information about the specific events that can be monitored.

Any spaces in the description are automatically replaced by the + symbol. Your script must substitute the + back to a space. This substitution is a typical method for sending spaces to server-side objects since spaces cannot be used in any valid URL.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="serverEventHandler" value="http://myserver.com/eventScript.pl?" />
```

Related reference:

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

Labels

Use the label parameters to configure, display, and print labels for pages and thumbnails.

displayPageLabels parameter

This parameter determines whether page labels are displayed in the viewer.

If you set this parameter to false, IBM Daeja ViewONE does not display page labels in the viewer. However, any page labels that are set are added when the user prints.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- No additional modules

Example

```
<param name="displayPageLabels" value="false" />
```

docRef<N> parameter

This parameter determines the reference number that is to be used for the specified document. You might use this document reference number in error retrieval files and print separator pages.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="docRef1" value="ref000" />
```

pageLabel<N> parameter

This parameter specifies a different label for each page.

For example, use **pageLabel1** for page 1, **pageLabel2** for page 2, and so on. The text appears in the lower right corner of the full-page display area.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="pageLabel1" value="My first page" />
<param name="pageLabel2" value="My second page" />
<param name="pageLabel3" value="My third page " />
```

pageLabelColor<N> parameter

This parameter specifies the color of each label.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

If you omit this parameter, the default label color is white.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="pageLabelColor1" value="255,255,192" />
<param name="pageLabelColor2" value="255, 192, 255" />
<param name="pageLabelColor3" value="192,255, 255" />
```

Related reference:

"IBM Daeja ViewONE color scheme text" on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

pageLabelNullifier parameter

This parameter allows specific label values to be used as null or blank labels.

For example, if you want to define a label for page 2 but not page 1, then you still need to add **pageLabel1** and **pageLabel2** to your HTML. By adding both labels to your HTML, **pageLabel1** must still have a value. To overcome this problem, define a value for **pageLabelNullifier**, then set the value for **pageLabel1** to this value.

This parameter applies to **pageLabel<N>**, **thumbLabel<N>**, and **printLabel<N>**.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="pageLabelNullifier" value="-" />
```

printLabelBorder parameter

This parameter determines whether the borders for the page label are printed.

When this parameter is set to true, the borders around the labels (lower right of the image) are printed. The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.228 or later
- IBM Daeja ViewONE Professional V1.0.228 or later
- ViewONE Print Accelerator Module

Example

```
<param name="printLabelBorder" value="true" />
```

printLabels parameter

This parameter determines whether page labels are printed.

When this parameter is set to true, the labels are printed. The default value is true.

Important: Thumbnail labels cannot be printed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printLabels" value="false" />
```

thumbLabel<N> parameter

This parameter specifies a different label for each thumbnail; **thumbLabel1** for thumbnail 1, **thumbLabel2** for thumbnail 2, and so on.

The text appears in the lower right corner of each corresponding thumbnail.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="thumbLabel1" value="My first page" />  
<param name="thumbLabel2" value="My second page" />  
<param name="thumbLabel3" value="My third page" />
```

thumbLabelColor<N> parameter

This parameter specifies the color of each thumbnail label.

You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*. The default color is white (that is, when a color label is not found).

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="thumbLabelColor1" value="255,255,192" />
<param name="thumbLabelColor2" value="255, 192, 255" />
<param name="thumbLabelColor3" value="192,255, 255" />
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

Local file options

The parameters specify which buttons are included on the toolbar.

defaultSaveFilename parameter

This parameter determines the default file name that appears in the **File name** field of the Save document dialog box.

This parameter is a document-specific parameter. If you hardcode the value, the file type might not be appropriate to the document. Therefore, parameterize the value so that a document is saved with the same file extension that a user would choose to save the file locally.

You must specify this parameter for each instance of IBM Daeja ViewONE.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="defaultSaveFilename" value="myFile.tif" />
```

fileButtonClose parameter

This parameter determines whether the toolbar includes **Close dialog** button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="fileButtonClose" value="false" />
```

fileButtonOpen parameter

This parameter determines whether the toolbar includes the **Open document** button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="fileButtonOpen" value="false" />
```

fileButtonSave parameter

This parameter determines whether the toolbar includes the **Save document** button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileButtonSave" value="false" />
```

fileButtons parameter

This parameter determines whether the toolbar includes the **Open document** button, **Save document** button, and **Close document** button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="fileButtons" value="false" />
```

fileKeyClose parameter

This parameter determines whether the keyboard shortcut to close a file is enabled.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileKeyClose" value="false" />
```

fileKeyOpen parameter

This parameter determines whether the keyboard shortcut for **Open File** is enabled.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileKeyOpen" value="false" />
```

fileKeySave parameter

This parameter determines whether the keyboard shortcut for the **Save document** operation is enabled.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileKeySave" value="false" />
```

fileKeys parameter

This parameter determines whether the keyboard shortcuts for file operations are enabled.

Tip: For multiple-page TIFF files, the save keyboard shortcut saves the whole document. For other file types, the save keyboard shortcut saves only the current page.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileKeys" value="false" />
```

Related reference:

 [Accessibility features of IBM Daeja ViewONE](#)

fileMenuClose parameter

This parameter determines whether the **Close file** menu is available.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileMenuClose" value="false" />
```

fileMenuOpen parameter

This parameter determines whether the **Open File** menu is available.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileMenuOpen" value="false" />
```

fileMenuSave parameter

This parameter specifies whether the **Save document** menu is available.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileMenuSave" value="false" />
```

fileMenus parameter

This parameter determines whether the file menus are displayed when the user clicks the middle or right mouse button.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileMenus" value="false" />
```

fixedSavePath parameter

This parameter determines the base file path to which a document, page, or selection of pages is saved.

If a user saves a document, page, or selection of pages and the **fixedSavePath** parameter is set, the user is prompted to enter a file name. The item is then saved with that file name in the directory specified by this parameter. If the user enters a subdirectory for the file, that directory is ignored.

If the **fixedSavePath** parameter is not specified, the Save document dialog box is displayed so the user can choose a save location.

Because this is a client-side save, you must ensure that the path that you specify is accessible from the client.

Requirements

- IBM Daeja ViewONE Standard V3.1.148 or later
- IBM Daeja ViewONE Professional V1.1.148 or later
- No additional modules

Example

```
<param name="fixedSavePath" value="\\myserver\ImagePath" />
```

java2FileDialogs parameter

This parameter determines whether the Open document dialog box and Save document dialog box are displayed as Java 2 dialog boxes.

The default value for this parameter is false, which uses Abstract Window Toolkit (AWT) to display the dialog boxes.

Requirements

- IBM Daeja ViewONE Standard V3.0.780 or later
- IBM Daeja ViewONE Professional V1.0.780 or later
- No additional modules

Example

```
<param name="java2FileDialogs" value="true" />
```

makeSaveNameSafe parameter

This parameter controls the checking and manipulation of the save file name.

If this parameter is set to false, the **defaultSaveFilename** parameter is not checked, and the value that is specified in the parameter is used.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="makeSaveNameSafe" value="false" />
```

Related reference:

“**defaultSaveFilename** parameter” on page 181

This parameter determines the default file name that appears in the **File name** field of the Save document dialog box.

newWindowButtons parameter

This parameter specifies whether the **new window** button and menus are available.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="newWindowButtons" value="false" />
```

newWindowKey parameter

This parameter specifies whether the keyboard shortcut for **New Window** operation is enabled.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="newWindowKey" value="false" />
```

saveDocumentWithoutAnnotations parameter

This parameter determines whether a document is saved locally without annotations.

When this parameter is set to true, the displayed image is saved locally without the standard associated annotations file. Typically a locally saved document has the same file name as the image file with the .ant extension.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- No additional modules

Example

```
<param name="saveDocumentWithoutAnnotations" value="true" />
```

tiffSaveColor parameter

Setting this parameter to true changes the default behavior so that when documents are saved locally, they are saved as color TIFF images.

The default is to save images as monochrome TIFF images. IBM Daeja ViewONE saves color images only if the source document is color.

Important: The TIFF standard is often interpreted differently by different viewer and imaging toolkit vendors. It is possible color TIFFs that are created by one product might not be visible by another.

Where this parameter is set to true, all TIFF files are output in the LZW compressed format, which supports color. They are always output in the LZW format even if there is no color present in the image/annotations.

The default for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.824 or later
- IBM Daeja ViewONE Professional V1.0.824 or later
- No additional modules

Example

```
<param name="tiffSaveColor" value="true" />
```

tiffSaveVersion6Jpeg parameter

Setting this parameter to false changes the default behavior of JPEG documents.

When JPEG documents are saved locally, they are saved as color TIFF 7 images by using the **tiffSaveColor** parameter along with a .tiff file extension.

This action occurs because the TIFF6 standard is not stringent enough in its specification of how color TIFFs are written. For that reason TIFF7 was introduced, though few viewer products support TIFF7. Writing color TIFF7 documents does not mean that the document is viewable in all viewing products. IBM Daeja ViewONE supports both TIFF7 and attempts to support most color TIFF6 documents.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.210 or later
- IBM Daeja ViewONE Professional V1.0.210 or later
- No additional modules

Example

```
<param name="tiffSaveVersion6Jpeg" value="false" />
```

Related reference:

“**tiffSaveColor** parameter” on page 186

Setting this parameter to true changes the default behavior so that when documents are saved locally, they are saved as color TIFF images.

Microsoft Office module options

Use the following parameters to enable, disable, and configure email and Excel options.

allowExternalEmailImages parameter

This parameter determines whether external email images can be loaded into a displayed email message.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="allowExternalEmailImages" value="true" />
```

allowSaveEmailAttachments parameter

This parameter determines whether users can save attachments in message files to their local file systems.

This default value is false, which prevents users from saving the attachments to the local file systems.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="allowSaveEmailAttachments" value="true" />
```

blockEmailModule parameter

This parameter determines whether email can be viewed.

When this parameter is set to true, the ViewONE Email Viewing Module is not available, even if the license file permits it.

The default value is false.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="blockEmailModule" value="true" />
```

emailHeaderInfo parameter

If this parameter determines whether email header information such as the available internet mail headers and server routing are displayed in the document information dialog box.

The default value is false, which prevents the display of email header information in the document information dialog box.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="emailHeaderInfo" value="true" />
```

excelGridlines parameter

This parameter determines whether grid lines are displayed in the spreadsheet view.

You can set this parameter to the following values:

pref Grid lines are displayed according to the user preference.

show Grid lines are displayed.

hide Grid lines are not displayed.

The default value for this parameter is pref.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="excelGridlines" value="show" />
```

excelViewMode parameter

This parameter determines the mode that is to be used for the spreadsheet view.

By default, the ViewONE Office Module initially attempts to display spreadsheets in sheet view to more accurately reflect how the document appears in the local viewer. However, the viewer is optimized for performance in this respect and attempts to display the spreadsheet in this mode only if enough resources are available. If there are not enough resources, the viewer reverts to displaying in page view mode and displays a dialog box that informs the user of this action.

You can set this parameter to the following values.

Value	Description
pref	Use the mode specified in the user preferences
sheet	Use sheet view mode for all spreadsheets
page	Use page view mode for all spreadsheets

The default value for this parameter is pref.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="excelViewMode" value="pref" />
```

listEmailAttachmentsInHeader parameter

This parameter determines if attachments are listed in the header section of the displayed email.

If this parameter is set to true, then attachments are listed in the header section of the displayed email.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="listEmailAttachmentsInHeader" value="true" />
```

showExternalEmailImages parameter

This parameter determines whether images from external email are displayed.

When this parameter is set to true and the **allowExternalEmailImages** parameter is set to true, the server retrieves and displays the image.

The default value is false.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="showExternalEmailImages" value="true" />
```

Related reference:

“**allowExternalEmailImages** parameter” on page 187

This parameter determines whether external email images can be loaded into a displayed email message.

showPlainTextForRTFEmail parameter

This parameter determines whether email messages that are stored in RTF are displayed as plain text.

Some Microsoft email messages are stored in RTF format instead of HTML. The ViewONE Email Viewing Module does not currently support RTF format.

When this parameter is set to true, the ViewONE Email Viewing Module looks for a plain text alternative within the email and displays the text where it is available. The RTF content is not interpreted as plain text, and the plain text is displayed only if the email also contains a plain text alternative content.

The default value is false.

Requirements

- IBM Daeja ViewONE Professional V4.0.2 or later
- ViewONE Office Module

Example

```
<param name="showPlainTextForRTFEmail" value="true" />
```

Documents and images

The parameters determine how documents and images are opened and used.

baseFilename parameter

This parameter is used with the **pageCount** parameter to make it easier to view documents by using a server object (such as CGI, ASP, or EXE).

The value that is the basic form of the object's URL.

When the user starts IBM Daeja ViewONE and requests a page, the viewer concatenates the URL value with the **page** parameter and the requested page value before the object is called. For example, the final URL that IBM Daeja ViewONE might use to call a server object for the second page of a document would look something like this URL: `http://www.mysite.com/script.pl?docid=2&page=2`

Important: The **pageCount** parameter is required for this parameter to work.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="baseFilename" value="http://www.mysite.com/script.pl?docid=2" />
```

Related reference:

"**pageCount** parameter" on page 197

This parameter specifies the number of pages in the document.

description parameter

This parameter determines the message that is displayed in the status bar when IBM Daeja ViewONE is retrieving files.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

The following example displays the following messages to be displayed in the status bar: `Opening docXYZ....` or `Opening docXYZ, page 2....`. The specific message depends on the page that is being retrieved.

```
<param name="description" value="docXYZ" />
```

doc<N> parameter

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

The default location for the parameters file is the codebase. However, you can specify a relative location or an absolute URL for the file.

When this parameter is used, extra buttons are made available so that the user can browse the document list.

The parameters file is a text file in which each line contains a single parameter. Each parameter name and value must be specified on a separate line. For example, you might have a docParams1.txt parameters file that contains the following parameters:

```
filename = "myfile.tif "  
printHeader = "My first document "
```

You might have a docParams2.txt parameters file that contains the following parameters:

```
page1 = "page1.tif "  
page2 = "page2.tif "  
page3 = "page3.tif "  
printHeader = "My second document "
```

The parameters file can specify parameters that are also specified in the HTML parameter list. In this case, the settings in the parameters file take precedence. For example, assume the following parameters are specified in the Mywebpage.htm file:

```
<param name="invert" value="true" />  
<param name="doc1" value="docParamsA.txt" />  
<param name="doc2" value="docParamsB.txt" />
```

These parameters initiate the viewer with the invert parameter set to true and configure the viewer to use docParamA.txt and docParamB.txt parameters files for its documents.

The docParamsA.txt parameters file contains the following parameters:

```
filename = "myfileA.tif "  
invert = false
```

These parameters specify an image path and set the invert parameter to false

The docParamsB.txt parameters file contains the following parameter that specifies an image path:

```
filename = "myfileB.tif "
```

When the viewer is started, the invert parameter is set to true as specified in the Mywebpage.htm file. However, the viewer opens the docParamsA.txt file almost immediately, which sets the invert flag to false. Therefore, myfileA.tif image is displayed as normal.

The docParamsB.txt parameters file does not specify the **invert** parameter. Therefore, the invert flag reverts to the value set in Mywebpage.htm file when the user clicks the **Next Document** button to open the docParamsB.txt image and the image is inverted.

You can only specify document-specific parameters in a parameters file. For example, the following parameters are valid: **rotation**, **flip**, **zoom**, **pageLabel**<N>, **thumbLabel**<N>, **fileList**, and **page**<N>. You cannot specify viewer-specific parameters such as the **obfuscate** parameter, button parameters, and menu configuration parameters.

If you specify a parameter in a document parameters file, it is best to set that parameter in all document parameters files rather than relying on the default parameter value for some documents.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="doc1" value="docParams1.txt" />
<param name="doc2" value="docParams2.txt " />
<param name="doc3" value="docParams3.txt " />
```

fileBase parameter

This parameter determines the base file path that is to be used when relative paths are specified for the following parameters: **filename**, **page<N>**, **thumb<N>**, and **annotationFile**.

By default, the viewer uses the **codebase** property to determine the base file path. You can use the **fileBase** parameter to specify a different base file path.

You must specify an absolute path that provides the full URL as the value for this parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileBase" value="http://www.mysite.com/docs/" />
```

fileList parameter

This parameter identifies a file that lists the files that represent individual pages in a document in which the pages are split into separate files. You can use this parameter for a large document instead of specifying the **page<N>** parameter for each page.

By using the **fileList** parameter, you can also keep the HTML file for the viewer constant. For example, you can make changes to the source file for different documents instead of changing the HTML file.

You must specify a text file as the value for this parameter. Separate the page files by using a carriage return or a line feed. You can specify an absolute path to the files or a path that is relative to the viewer codebase. For example, the following file list uses a relative path to specify the page files:

```
mypages/page1.tif
mypages/page2.tif
mypages/page3.tif
```

You can include separate thumbnail files in the file list by using the **t** flag as shown in the following example:

```
mypages/page1.tif  
t mypages/thumb1.tif  
mypages/page2.tif  
t mypages/thumb2.tif  
mypages/page3.tif  
t mypages/thumb3.tif
```

This parameter can also be used to point to a single file name. This function is useful in the scenario where you might not know whether you are dealing with a single document or multiple pages.

To put the viewer in file name mode, return the following in your file list:

```
filename=http://yourhost.com/file.tiff
```

It is invalid to specify anything else in the file if you have a file name specified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="filelist" value="mylist/list.lst" />
```

Related reference:

“**page**<N> parameter” on page 196

If the pages of the document are split into separate files, then this parameter specifies the file for each page.

fileDir parameter

This parameter determines the local drive and directory that the viewer is to search. The viewer searches the specified path, including subdirectories, to locate all files. The viewer then opens the files as a single document with each file assumed to be a single page image file.

You might set this parameter when the user wants to view all images in a directory on a on hard disks that is accessible locally or by using a mapped drive.

You must use forward slashes as path separators in the value.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="fileDir" value="c:/myimages" />
```

filename parameter

This parameter determines the file name of the document that is to be viewed.

You can specify the value for this parameter by entering either the file name relative to the codebase or the full URL for the file. The codebase is specified in the HTML code for the viewer. The following example shows a full URL:

Important: Use a forward slash (/) in the value string. If you must use a backslash, use a double backslash (\) to represent each backslash.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="filename" value="images/myimage.tif" />
```

httpRedirectAllowSSL parameter

This parameter determines whether an HTTPS request is to be automatically redirected to an HTTP request.

By default, the Java runtime environment (JRE) does not redirect an HTTPS request to an HTTP request. To force requests to be redirected, set this parameter to true.

The default value for this parameter is false.

Tip: This parameter is disabled by default unless the **processHttpRedirects** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard V3.1.200 or later
- IBM Daeja ViewONE Professional V1.1.200 or later
- No additional modules

Example

```
<param name="httpRedirectAllowSSL" value="true" />
```

Related reference:

“**processHttpRedirects** parameter” on page 199

By default, the JRE takes care of all redirect response codes that come back from the server. This parameter determines whether the viewer's internal method is used to process these requests.

httpRedirectCountLimit parameter

This parameter determines the number of times the viewer attempts to redirect before returning an error.

The default value for this parameter is 5.

You must set the **processHttpRedirects** parameter is set to true to use this parameter.

Requirements

- IBM Daeja ViewONE Standard V3.1.200 or later
- IBM Daeja ViewONE Professional V1.1.200 or later
- No additional modules

Example

```
<param name="httpRedirectCountLimit" value="10" />
```

Related reference:

“**processHttpRedirects** parameter” on page 199

By default, the JRE takes care of all redirect response codes that come back from the server. This parameter determines whether the viewer's internal method is used to process these requests.

page<N> parameter

If the pages of the document are split into separate files, then this parameter specifies the file for each page.

The example document has three pages.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="page1" value="images/page1.tif" />  
<param name="page2" value="images/page2.tif" />  
<param name="page3" value="images/page3.tif" />
```

This parameter can specify either the file name relative to the codebase or the full URL, for example:

```
<param name="page1" value="http://mysite/images/page1.tif" />
```

Viewing several multipage TIFF or PDF files as a single document:

To open several multipage TIFF or PDF files at the same time, use the **pageN** parameter to specify which pages of as many multipage TIFF or PDF file names as you require. The # qualifier specifies the pages. For example:

```
<param name="page1" value="images/doc1.tif#1" />  
<param name="page2" value="images/doc1.tif#2" />  
<param name="page3" value="images/doc2.tif#1" />
```

Where:

```
doc1.tif#1 = page 1 of doc1.tif  
doc1.tif#2 = page 2 of doc1.tif  
doc1.tif#1 = page 1 of doc2.tif
```

Or:

```
doc1.pdf#1 = page 1 of doc1.pdf  
doc1.pdf#2 = page 2 of doc1.pdf  
doc2.pdf#1 = page 1 of doc2.pdf
```

The first example appears to open a single document with three pages. However, the first two pages come from the multipage tiff file `doc1.tif` and the third page comes from `doc2.tif`.

The number that follows the # qualifier can be any valid page number for the associated TIFF file.

pageCount parameter

This parameter specifies the number of pages in the document.

Important: The **baseFilename** parameter is required for this parameter to be effective.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="pageCount" value="3" />
```

Related reference:

“**baseFilename** parameter” on page 191

This parameter is used with the **pageCount** parameter to make it easier to view documents by using a server object (such as CGI, ASP, or EXE).

pageNumber parameter

This parameter specifies the initial page of the document to be viewed.

The default value is 1.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="pageNumber" value="1" />
```

prefetchPages parameter

This parameter determines the number of pages that are read-ahead by IBM Daeja ViewONE.

This parameter is effective only where you are using either the **page<N>** parameter or the **streamerURL** parameter module, for example, when a document is loaded page-by-page.

When this parameter is set to any value greater than zero, the number of pages that are specified after the current page was viewed are read-ahead (prefetched). The default value is 0.

Depending on the specific needs of the user and the network setup, this read-ahead mechanism can help improve document browsing speed.

The value corresponds to the number of read-ahead pages, in the direction that the user browses a document. For example, if the value is set to 2 and the user views page 1, IBM Daeja ViewONE will read-ahead pages 2 and 3.

If the user then views page 2, page 4 is read-ahead by IBM Daeja ViewONE (page 3 was already read-ahead). Page 3 appears quickly since it was already read-ahead while page 1 was viewed.

If the user then views, say, page 10, IBM Daeja ViewONE will read-ahead pages 11 and 12, on so on.

If then, finally, the user views page 8 (from say page 10), IBM Daeja ViewONE will read-ahead pages 7 then 6, that is, backwards because the user changed their direction of browsing the document.

This process can improve viewing for browsing, but it might also increase the network and server load. The benefit of prefetching depends on the type of browsing performed by users, the network speed, and the ability for the server to keep pace with such read-ahead mechanisms.

If the value of the parameter is set to *-1*, then all pages in the document are prefetched. Use this setting with caution as it certainly increases network and server load, especially on large documents with many pages.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="prefetchPages" value="2" />
```

Related reference:

“**page**<N> parameter” on page 196

If the pages of the document are split into separate files, then this parameter specifies the file for each page.

“**streamerURL** parameter” on page 258

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

prefetchThumbs parameter

This parameter determines the number of thumbnail pages that are read-ahead by IBM Daeja ViewONE.

When this parameter is set to any value greater than zero, and when used with the **thumb**<N> parameter, IBM Daeja ViewONE reads-ahead the specified number of thumbnail pages from the current thumbnail. For example, if the value is set to 2 and the user views thumbnail 1, IBM Daeja ViewONE will read-ahead thumbnails 2 and 3.

This parameter works in the same way as the **streamerURL** parameter for thumbnails.

Depending on the specific needs of the user and the network setup, this read-ahead mechanism can help improve document browsing speed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
<param name="prefetchThumbs" value="2" />
```

Related reference:

“**streamerURL** parameter” on page 258

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

processHttpRequests parameter

By default, the JRE takes care of all redirect response codes that come back from the server. This parameter determines whether the viewer's internal method is used to process these requests.

When this parameter is set to true, the viewer's internal method is used processing the HTTP redirect requests.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.200 or later
- IBM Daeja ViewONE Professional V1.1.200 or later
- No additional modules

Example

```
<param name="processHttpRequests" value="true" />
```

redirectNotFound parameter

This property specifies the URL for the redirect page to display when the first page of a document cannot be retrieved.

When this parameter is set to true, and there is a problem when the first page of a document is retrieved, the viewer redirects the page to the URL that is specified.

The default value is false.

Use this parameter with the **redirectTarget** parameter to determine the frame or window to use for the redirect page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="redirectNotFound" value="http://myserver/docs/retrievalerror.htm" />
```

redirectTarget parameter

This parameter specifies the browser frame or window to use when there is a document retrieval problem.

Use this parameter with the **redirectNotFound** parameter.

The following table contains the full list of values for this parameter.

Value	Description
_blank	Redirect to a new browser window.
_self	Redirect to the same window or frame that contains IBM Daeja ViewONE (default).
_top	Redirect to the top-level frame of the window that contains IBM Daeja ViewONE. This action is the same as _self if frames are not used.
_parent	Redirect to the parent of the frame that IBM Daeja ViewONE is in. This action is the same as _self if frames are not used.
Name	Redirect to the named window. If a window with this name does not exist, then a top-level window is created with this name.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="redirectTarget" value="_blank" />
```

Related reference:

“**redirectNotFound** parameter” on page 199

This property specifies the URL for the redirect page to display when the first page of a document cannot be retrieved.

resourceToUrlEncoding parameter

This parameter determines whether Chinese characters can be used in URLs.

When this parameter is set to *loose*, Chinese characters can be used in URLs.

Requirements

- IBM Daeja ViewONE Standard V3.1.198 or later
- IBM Daeja ViewONE Professional V1.1.198 or later
- No additional modules

Example

```
<param name="resourceToUrlEncoding" value="loose" />
```

routeDocs parameter

This parameter determines how unsupported image files are viewed.

When this parameter is set to true, IBM Daeja ViewONE routes the unsupported image files to the browser for rendering. For example, if the **filename** parameter points to a Microsoft Word document, then IBM Daeja ViewONE allows the browser to handle viewing directly with Microsoft Word.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="routeDocs" value="true" />
```

Related reference:

“**filename** parameter” on page 194

This parameter determines the file name of the document that is to be viewed.

routeHTML parameter

This parameter controls how IBM Daeja ViewONE handles HTML responses from the server.

When this parameter is set to true, any HTML responses from the server are displayed in a separate browser window, rather than inside IBM Daeja ViewONE as an image.

The default value is false.

This parameter works with the optional **routeHTMLTarget** parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="routeHTML" value="true" />
```

routeHTMLTarget parameter

This parameter specifies how HTML responses from the server are redirected.

Use this optional parameter with the **routeHTML** parameter.

The following table lists the valid values.

Value	Description
_blank	Redirect to a new browser window.
_self	Redirect to the same window or frame that contains IBM Daeja ViewONE (default).
_top	Redirect to the top-level frame of the window that contains IBM Daeja ViewONE. This action is the same as _self if frames are not used.
_parent	Redirect to the parent of the frame that IBM Daeja ViewONE is in. This action is the same as _self if frames are not used.
Name	Redirect to the named window. If a window with this name does not exist, then a top-level window is created with this name.

Requirements

- IBM Daeja ViewONE Standard V3.1.166 or later
- IBM Daeja ViewONE Professional V1.1.166 or later
- No additional modules

Example

```
<param name="routeHTMLTarget" value="_blank" />
```

Related reference:

“**routeHTML** parameter” on page 201

This parameter controls how IBM Daeja ViewONE handles HTML responses from the server.

“**routeTarget** parameter” on page 203

This parameter specifies how images are redirected back to the browser when they have unsupported formats.

routeXML parameter

This parameter determines how IBM Daeja ViewONE works with XML responses from the server.

When this parameter is set to true, any XML responses from the server are displayed in a separate browser window, rather than inside IBM Daeja ViewONE as an image.

The default value is false.

This parameter works with the optional **routeXMLTarget** parameter.

Requirements

- IBM Daeja ViewONE Standard 4.014 or later
- IBM Daeja ViewONE Professional 4.014 or later
- No additional modules

Example

```
<param name="routeXML" value="true" />
```

Related reference:

“**routeXMLTarget** parameter”

This parameter determines how XML responses from the server are redirected.

routeXMLTarget parameter

This parameter determines how XML responses from the server are redirected.

Use this optional parameter with the **routeXML** parameter.

The following table lists the valid values.

Value	Description
_blank	Redirect to a new browser window.
_self	Redirect to the same window or frame that contains IBM Daeja ViewONE (default).

Value	Description
<code>_top</code>	Redirect to the top-level frame of the window that contains IBM Daeja ViewONE. This action is the same as <code>_self</code> if frames are not used.
<code>_parent</code>	Redirect to the parent of the frame that IBM Daeja ViewONE is in. This action is the same as <code>_self</code> if frames are not used.
<i>Name</i>	Redirect to the named window. If a window with this name does not exist, then a top-level window is created with this name.

Requirements

- IBM Daeja ViewONE Standard 4.014 or later
- IBM Daeja ViewONE Professional 4.014 or later
- No additional modules

Example

```
<param name="routeXMLTarget" value="_blank" />
```

Related reference:

“**routeXML** parameter” on page 202

This parameter determines how IBM Daeja ViewONE works with XML responses from the server.

“**routeTarget** parameter”

This parameter specifies how images are redirected back to the browser when they have unsupported formats.

routeTarget parameter

This parameter specifies how images are redirected back to the browser when they have unsupported formats.

Use this optional parameter with the **routeDocs** parameter.

The following table lists the valid values.

Value	Description
<code>_blank</code>	Redirect to a new browser window.
<code>_self</code>	Redirect to the same window or frame that contains IBM Daeja ViewONE (default).
<code>_top</code>	Redirect to the top-level frame of the window that contains IBM Daeja ViewONE. This action is the same as <code>_self</code> if frames are not used.
<code>_parent</code>	Redirect to the parent of the frame that IBM Daeja ViewONE is in. This action is the same as <code>_self</code> if frames are not used.
<i>Name</i>	Redirect to the named window. If a window with this name does not exist, then a top-level window is created with this name.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="routeTarget" value="_blank" />
```

Related reference:

“**routeDocs** parameter” on page 200

This parameter determines how unsupported image files are viewed.

suppressMenusDuringOpen parameter

This parameter determines whether the right-click menu is displayed while an image is loading.

When this parameter is set to true, the menu is not displayed while an image is loading if the user right-clicks in IBM Daeja ViewONE or presses the] key.

The default value is false, which allows the menu to be displayed at any time.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="suppressMenusDuringOpen" value="true" />
```

thumb<N> parameter

This parameter allows a separate thumbnail file to be specified for each page in a document.

In some instances, it might be advantageous to have separate files for the thumbnails to help browsing of thumbnails. Smaller files are quicker to download and view.

Important:

- This parameter is valid only when the **page<N>** parameter is used.
- You must ensure that a file is specified for all pages of the document. All pages must match the corresponding full resolution files that are specified by the **page<N>** parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thumb1" value="images/page1-t.tif" />  
<param name="thumb2" value="images/page2-t.tif" />  
<param name="thumb3" value="images/page3-t.tif" />
```

Related reference:

“**page**<N> parameter” on page 196

If the pages of the document are split into separate files, then this parameter specifies the file for each page.

timeOut parameter

This parameter sets and starts a usage timer.

If the user does not use the applet for the number of seconds that are specified, then the applet is automatically disabled. It can be re-enabled only by calling one of the timeout JavaScript methods, or by refreshing the page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="timeOut" value="30" />
```

PDF options

IBM Daeja ViewONE Professional introduced an optional ViewONE PDF Module.

The module comes with various optional parameters as follows.

alwaysShowPDFAnnotations parameter

This parameter determines whether the annotations that are stored in a PDF are shown when the user clicks **Show/Hide Annotations**.

If this parameter is set to true, IBM Daeja ViewONE shows the annotations.

The default value for this parameter is false, which hides the annotations that are in the PDF.

Requirements

- IBM Daeja ViewONE Professional V1.1.168 or later
- ViewONE PDF Module

Example

```
<param name="alwaysShowPDFAnnotations" value="true" />
```

autoLimitPDFResolutionValue parameter

This parameter controls the way that the viewer limits resolution with the **defaultAutoLimitPDFResolution** limiter parameter.

This parameter dictates the maximum amount of memory that is permitted for a decompressed PDF page. If a page exceeds this amount (for example, when high default PDF resolutions are used) then the viewer uses a resolution for that page that brings it under this memory limit. The resolution that is used depends on the page dimensions and color depth. To discover the limited resolution, a page must first be viewed, then you can display the Image Information dialog box.

The default value is 5 MB (5 * 1024 * 1024 = 5242880 bytes).

If the limited resolution results in degraded viewing quality, but you also want to use the auto limiter, then you can use this parameter to increase the default value. You might need to experiment to determine the best value for your environment. The documents, your hardware, your network, and user preferences affect performance.

Requirements

- IBM Daeja ViewONE Professional, any version
- ViewONE PDF Module

Example

```
<param name="autoLimitPDFResolutionValue" value="50000000" />
```

Related reference:

“**defaultAutoLimitPDFResolution** parameter”

This parameter determines whether IBM Daeja ViewONE dynamically limits the PDF resolution to obtain maximum performance when PDF documents are viewed.

defaultAutoLimitPDFResolution parameter

This parameter determines whether IBM Daeja ViewONE dynamically limits the PDF resolution to obtain maximum performance when PDF documents are viewed.

When this parameter is set to true, IBM Daeja ViewONE attempts to use the default PDF resolution. If that resolution is likely to result in slow performance, IBM Daeja ViewONE reduces the resolution on a per document and page basis. The amount of reduction is determined by the **autoLimitPDFResolutionValue** parameter.

With this parameter set to false, the **Optimize for performance** menu option is disabled in the viewer resolution user interface.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Professional, any version
- ViewONE PDF Module

Example

```
<param name="defaultAutoLimitPDFResolution" value="true" />
```

Related reference:

“**autoLimitPDFResolutionValue** parameter” on page 205

This parameter controls the way that the viewer limits resolution with the **defaultAutoLimitPDFResolution** limiter parameter.

defaultPDFPixelDepth parameter

This parameter determines the default pixel-depth that is used to view PDF documents.

Lower pixel depths result in higher viewing performance but fewer colors.

The default value for this parameter is 8, which is grayscale.

Requirements

- IBM Daeja ViewONE Professional, any version
- ViewONE PDF Module

Example

```
<param name="defaultPDFPixelDepth" value="8" />
```

defaultPDFResolution parameter

This parameter determines the default resolution for viewing PDF documents.

Lower resolutions result in higher viewing performance but lower quality. Higher resolutions result in slower viewing performance but higher quality.

The default value for this parameter is 200.

Requirements

- IBM Daeja ViewONE Professional, any version
- ViewONE PDF Module

Example

```
<param name="defaultPDFResolution" value="100" />
```

findResultColor parameter

This parameter determines the fill color that is used to highlight words that are found by a search.

The default value for this parameter is 152, 192, 217 (shade of blue).

Requirements

- IBM Daeja ViewONE Professional V1.1.148 or later
- ViewONE PDF Module

Example

```
<param name="findResultColor" value="255, 255, 0" />
```

findResultHighlightColor parameter

This parameter determines the border color that is used to highlight words that are found by searches.

The default value for this parameter is 0, 0, 0 (black).

Requirements

- IBM Daeja ViewONE Professional V1.1.148 or later
- ViewONE PDF Module

Example

```
<param name="findResultHighlightColor" value="R,G,B" />
```

findResultHistoryListSize parameter

This parameter determines the number of historical searches that are listed in the PDF search facility.

The default value for this parameter is 5.

Requirements

- IBM Daeja ViewONE Professional V1.1.148 or later
- ViewONE PDF Module

Example

```
<param name="findResultHistoryListSize" value="10" />
```

forceJavaPdf parameter

This parameter provides support for displaying PDFs on all client platforms.

Previous versions of IBM Daeja ViewONE Professional display PDFs only on Windows and Mac OSX systems with 32-bit Java and 32-bit web browsers. The **forceJavaPdf** parameter provides support for all client platforms.

To enable the new platform-independent module on Windows and Mac OSX with 32-bit Java, you must set the **forceJavaPdf** parameter to true.

Requirements

- IBM Daeja ViewONE Professional, V4.1.1 or later
- ViewONE PDF Module

Example

```
<param name="forceJavaPdf" value="true" />
```

Print options

The parameters display and configure print buttons, toolbars, and other dialog boxes.

allowPrintHeader parameter

This parameter determines whether the print header is included on items that are sent to the printer.

If this parameter is set to false, the print header is not included on items that are sent to the printer.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="allowPrintHeader" value="false" />
```

defaultPrintDoc parameter

This parameter determines whether the **Print** option prints all pages of a a multipage document by default.

If this parameter is set to false, the **Print** option prints only the current page. Users can still use the menus to print all pages.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="defaultPrintDoc" value="false" />
```

defaultPrintingNotesText parameter

This parameter determines whether IBM Daeja ViewONE prints the text from note annotations on separate pages at the end of an image print job.

The default value is true, which causes the note annotation text to be printed on separate pages.

This parameter can be overridden if the user explicitly sets a preference for printing note annotation text in the user preference menu.

Requirements

- IBM Daeja ViewONE Standard V3.0.334 or later
- IBM Daeja ViewONE Professional V1.0.334 or later
- ViewONE Annotations Module

Example

```
<param name="defaultPrintingNotesText" value="false" />
```

defaultPrintNoteAnnotation parameter

This parameter determines whether note annotation shapes on an image are printed.

If this parameter is set to false, IBM Daeja ViewONE does not include note annotation shapes on an image as part of the printing process.

The default value for this parameter is true.

This parameter can be overridden if a user sets the preference for printing note annotation shapes in the user preference menu.

When note annotation shapes are printed, IBM Daeja ViewONE also prints a number if the **showNoteIds** parameter is set to true or is unspecified. The number is printed unless the applet that is printing is hidden on the web page.

Requirements

- IBM Daeja ViewONE Standard V3.0.334 or later
- IBM Daeja ViewONE Professional V1.0.334 or later
- ViewONE Annotations Module

Example

```
<param name="defaultPrintNoteAnnotation" value="false" />
```

Related reference:

“**showNoteIds** parameter” on page 99

This parameter determines whether an identifying number is displayed inside sticky note markers.

disablePrintCopies parameter

If this parameter is set to true, then the ViewONE Print Accelerator Module ignores the value set in the print dialog box for the number of copies and only ever print a single copy.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.338 or later
- IBM Daeja ViewONE Professional V1.0.338 or later
- ViewONE Print Accelerator Module

Example

```
<param name="disablePrintCopies" value="true" />
```

forcePrintsPerPage parameter

This parameter determines whether the number of images that are specified by the **multiPrintNum** parameter are always printed on the same page.

If the **multiPrintNum** parameter is set to a value other than 1, the printed page is evenly divided to accommodate the specified number of images. Each image is printed with its header and footer.

If the images are too large to fit the page and this parameter is set to true, the ViewONE Print Accelerator Module always prints the specified number of images on a same page. If the images exceed the page width, the images are shrunk to fit the available space. When shrunk, the aspect ratio is maintained.

If the **multiPrintNum** parameter is not used or is set to 1, this parameter has no effect.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="forcePrintsPerPage" value="true" />
```

Related reference:

“**multiPrintNum** parameter” on page 211

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

imageTooLargePrintHeader parameter

This parameter determines the message that is displayed when the image that is being printed exceeds the printer page and the **printOriginalSizeAllowed** is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="imageTooLargePrintHeader" value="your message" />
```

Related reference:

“**printOriginalSizeAllowed** parameter” on page 224

This parameter determines whether the print original size option is available in the preferences menu.

java2PrintSetupDialog parameter

This parameter determines whether the standard Java 2 Print dialog box is used for printing.

This parameter is applied only when IBM Daeja ViewONE Professional is not used.

If you set this parameter false, the Abstract Window Toolkit (AWT) is used to display the dialog boxes.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="java2PrintSetupDialog" value="false" />
```

multiPrintNum parameter

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

This parameter allows images to be stacked vertically, and IBM Daeja ViewONE works out automatically whether there is room on an image-by-image basis. The value of this parameter is the maximum number of images you want to be printed on a single page.

The default value is 1.

If you require the specified number of images to always be printed on the same page, you can use this parameter with the **forcePrintsPerPage** parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="multiPrintNum" value="2" />
```

Related reference:

“**forcePrintsPerPage** parameter” on page 210

This parameter determines whether the number of images that are specified by the **multiPrintNum** parameter are always printed on the same page.

multiPrintGap parameter

The parameter sets the vertical gap between images when multiple images are printed on a single page of a multipage document.

This parameter is only available with the ViewONE Print Accelerator Module, and when the **multiPrintNum** parameter is greater than 1.

When a decimal value is used, the value must include the leading digits. For example, use 0.3, instead of .3.

The unit is in inches, and the default value is 0.02.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="multiPrintGap" value="0.5" />
```

Related reference:

“**multiPrintNum** parameter” on page 211

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

multiPrintSingleHeader parameter

This parameter determines whether a single header is printed for multiple images on a single page.

This parameter is only available with the ViewONE Print Accelerator Module and when the **multiPrintNum** parameter value is greater than 1.

When a print header is defined with the **printHeader** parameter, and more than one image is printed on a page, the header is ordinarily printed above each image. If you want only one header at the top of the printed page, then set this parameter to true.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="multiPrintSingleHeader" value="true" />
```

Related reference:

“**multiPrintNum** parameter” on page 211

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

“**printHeader** parameter” on page 217

Printouts can include custom text at the top of each page.

printAcceleratorForgetSettings parameter

This parameter determines whether the printing options are saved as preferences.

If this parameter is set to true, then the ViewONE Print Accelerator Module does not remember printer settings between print operations. For example, if a user sets the printer settings to landscape mode in one print job, then the next time that the user prints in that viewer session, the setting reverts to portrait mode.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printAcceleratorForgetSettings" value="true" />
```

printAccelerator parameter

This parameter determines whether the advanced print engine feature is enabled.

The ViewONE Print Accelerator Module significantly improves printing. The print engine is available for Windows operating systems only.

This feature is available as a licensed optional extra.

When this parameter is set to true, the ViewONE Print Accelerator Module is used. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printAccelerator" value="true" />
```

printAcceleratorReverseMono parameter

This parameter determines if a print operation inverts the colors in the image.

If this parameter is set to true, then the ViewONE Print Accelerator Module inverts the image at the last moment before printing. For example, if a user prints a

document that has a white background and a black foreground, then the print output is inverted so that it has a black background and a white foreground.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printAcceleratorReverseMono" value="true" />
```

printAnnotations parameter

This parameter determines whether annotations are printed when the image is printed.

When this parameter is set to true, the annotations are printed. The default value is true.

When this parameter is set to false, the annotations are not printed.

Important: Use care when you set the value to false, since it represents a potential security loophole. For example, a user might think that they securely redacted an area of text, but the redaction does print if this parameter is set to false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
<param name="printAnnotations" value="false" />
```

printAutoRotate parameter

This parameter determines whether images are rotated before printing so that as many images can be printed (vertically) on a printed page as possible.

When the **multiPrintNum** parameter is set to more than 1 and this parameter is set to true, then IBM Daeja ViewONE attempts to rotate images automatically.

When the **multiPrintNum** parameter is set to 1 and this parameter is set to true, then IBM Daeja ViewONE attempts to rotate all landscape style images (that is, images where the image width is greater than the image height).

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printAutoRotate" value="true">
```

Related reference:

“**multiPrintNum** parameter” on page 211

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

printButtons parameter

This parameter determines whether the toolbar includes the **Print document** button.

When this parameter is set to true, the **Print document** button is displayed. The default value is true.

When this parameter is set to false, the **Print document** button is not displayed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printButtons" value="false" />
```

printComplete parameter

This parameter determines whether a confirmation message is displayed to the user after a print job completes.

When this parameter is set to true, a Print Complete message is displayed after a print job completes.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printComplete" value="true" />
```

printDialog parameter

This parameter determines whether the standard print dialog box is displayed.

When this parameter is set to false, printing takes place without showing the standard print dialog box. The printer that IBM Daeja ViewONE

uses when the print dialog box is not shown is determined by the **reusePrinter** parameter.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printDialog" value="false" />
```

Related reference:

“**reusePrinter** parameter” on page 226

This parameter determines if the default printer is used or if a previously named printer is used when the print dialog box is disabled.

printDocumentAllowed parameter

This parameter determines whether the print document menu item is available.

When this parameter is set to false, the print document menu item is hidden.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printDocumentAllowed" value="false" />
```

printer parameter

This parameter specifies the printer to which all print output is sent when the print dialog box is not displayed.

This parameter applies only when the **printDialog** parameter is set to false.

You must specify a name (or part thereof) for the printer to which all print output is sent. The named printer becomes IBM Daeja ViewONE default printer for the duration that the printer dialog box is disabled. The printer's default settings are used for all printouts, for example, orientation and resolution.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printer" value="myprinter" />
```

Related reference:

“**printDialog** parameter” on page 215

This parameter determines whether the standard print dialog box is displayed.

printHeader parameter

Printouts can include custom text at the top of each page.

The following header options are available:

false The header is not printed.

any text

Custom text is printed at the top of each printed page. For example, your own copyright for the documents that are being viewed or some other informational text is printed.

formatted text

Text can include limited formatted elements as follows.

Table 1. Header text options

Option	Printed result
\$page	Prints the word "page" in the appropriate translation.
\$of	Prints the word "of" in the appropriate translation.
\$pages	Prints the word "pages" in the appropriate translation.
#	Prints the page number of the current printed page.
##	Prints the number of pages in the document.

For example, use the following code to print "page 1 of 99 © Copyright 2000":

```
<param name="printHeader" value="$page # $of ## © Copyright 2000" />
```

By default this text is set to the page number followed by the number of pages in the document: "(\$page # \$of ##)"

Important:

Beginning with IBM Daeja ViewONE Version 3.0.228, print headers can contain multiple-lines for ViewONE Print Accelerator Module users. Each line must be separated by either an LF or a CR character, or alternatively if you include <N> within your text, then this character is automatically substituted for an LF character.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printHeader" value="any text" />
```

printingColorHeader parameter

This parameter determines whether the ViewONE Print Accelerator Module always prints color headers and footers.

When this parameter is set to true, the ViewONE Print Accelerator Module always prints color headers and footers. The default value is false.

Important: Printing colored headers and footers is slower than printing monochrome headers and footers.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- ViewONE Print Accelerator Module

Example

```
<param name="printingColorHeader" value="true" />
```

printingFontBold parameter

The parameter determines whether a bold font is used to print the headers and footers with the ViewONE Print Accelerator Module.

The available fonts are listed in the *Supported fonts* topic.

This parameter is used with the **printingFontSize** and **printingFontName** parameters.

This parameter is effective only if the **printingFontSelectionEnabled** parameter is set to true.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printingFontBold" value="true" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

printingFontName parameter

The parameter specifies the font to be used to print the headers and footers with the ViewONE Print Accelerator Module.

The available fonts are listed in *Supported fonts*.

This parameter is used with the **printingFontSize** and **printingFontBold** parameters, and this parameter is effective only if the **printingFontSelectionEnabled** parameter is set to true.

The default value is Arial.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printingFontName" value="Courier" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

printingFontSelectionEnabled parameter

This parameter determines whether a font can be specified for printing the header and footer.

If this parameter is set to true, then the ViewONE Print Accelerator Module uses the font that is specified by the **printingFontName**, **printingFontSize**, and **printingFontBold** HTML parameters for the header and footer.

The default value is false.

The available fonts are listed in the *Supported fonts* topic.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printingFontSelectionEnabled" value="true" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

printingFontSize parameter

The parameter specifies the font size for printing the header and footer with the ViewONE Print Accelerator Module.

This parameter is used with the **printingFontBold** and **printingFontName** parameters, and this parameter is effective only if the **printingFontSelectionEnabled** parameter is set to true.

The default value is 10.

The available fonts are listed in the *Supported fonts* topic.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printingFontSize" value="15" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

printingNotesFontName parameter

This parameter specifies the font to use when the text from note annotations is printed onto separate pages at the end of an image print job.

The default value is *Arial*.

The available fonts are listed in the *Supported fonts* topic.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- Annotations and ViewONE Print Accelerator Module

Example

```
<param name="printingNotesFontName" value="Courier" />
```

Related reference:

“Supported fonts” on page 263

The default fonts for IBM Daeja ViewONE are listed in the following table.

printingNotesFontSize parameter

This parameter specifies the font size to use when the text from note annotations is printed onto separate pages at the end of an image print job.

The default value is 10.

The available fonts are listed in the *Supported fonts* topic.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- ViewONE Annotations Module
- ViewONE Print Accelerator Module

Example

```
<param name="printingNotesFontSize" value="20" />
```

printingNotesStandardFont parameter

This parameter determines whether the standard header font is used when the text from note annotations is printed onto separate pages at the end of an image print job. You use this parameter when you want the page headers, footers (page labels), and note annotation text to all be the same size.

When this parameter is set to true, IBM Daeja ViewONE uses the standard header font when the text from note annotations is printed onto separate pages at the end of an image print job.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- Annotations and ViewONE Print Accelerator Module

Example

```
<param name="printingNotesStandardFont" value="true" />
```

printingNotesTextPageTitle parameter

This parameter specifies the text that is printed for each page heading when the text from note annotations is printed onto separate pages at the end of an image print job.

The following text replacements apply:

Table 2. Page title text options

Option	Printed result
\$page	Prints the word "page" in the appropriate translation.
\$of	Prints the word "of" in the appropriate translation.
\$pages	Prints the word "pages" in the appropriate translation.
#	Prints the page number of the current printed page.
##	Prints the number of pages in the document.
<N>	Carriage return for multiple lines in the header

The default value is *Page: #*.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- ViewONE Annotations Module
- ViewONE Print Accelerator Module

Example

```
<param name="printingNotesTextPageTitle" value="$page # $of ##" />
```

printingNotesTextTitle parameter

This parameter specifies the title that is printed at the top of the page when the text from note annotations is printed onto separate pages at the end of an image print job.

You can include <N> within your text to print multiple lines in the title.

The default value is *Notes*.

Requirements

- IBM Daeja ViewONE Standard V3.0.332 or later
- IBM Daeja ViewONE Professional V1.0.332 or later
- ViewONE Annotations Module

Example

```
<param name="printingNotesTextTitle" value="My Title" />
```

printingNotesTitleFontSize parameter

This parameter specifies the font size to use for the title when the text from note annotations is printed onto separate pages at the end of an image print job.

The default value is twice the value that is specified by **printingNotesFontSize** parameter.

Requirements

- IBM Daeja ViewONE Standard V3.0.336 or later
- IBM Daeja ViewONE Professional V1.0.336 or later
- ViewONE Annotations Module
- ViewONE Print Accelerator Module

Example

```
<param name="printingNotesTitleFontSize" value="20" />
```

Related reference:

“**printingNotesFontSize** parameter” on page 220

This parameter specifies the font size to use when the text from note annotations is printed onto separate pages at the end of an image print job.

printJobName parameter

This parameter specifies the print job name that is sent to the printer when IBM Daeja ViewONE prints.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printJobName" value="my print job" />
```

printKeys parameter

This parameter determines whether the keyboard shortcuts for print operations are enabled.

When this parameter is set to true, the keyboard shortcuts for print operations are enabled. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printKeys" value="false" />
```

Related reference:

 [Accessibility features of IBM Daeja ViewONE](#)

printMenus parameter

This parameter determines whether the print menus are available when you click the middle or right mouse button.

When this parameter is set to true, the print menus are enabled. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printMenus" value="false" />
```

printMonoAllowed parameter

This parameter determines whether the user can select monochrome or color printing for color images.

When this parameter is set to false, all color images are printed in color.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printMonoAllowed" value="false" />
```

printMono parameter

This parameter sets the default behavior of the ViewONE Print Accelerator Module for monochrome printing.

When this parameter is set to true, documents are printed in monochrome by default and **Print document without color** is set to true by default.

If this parameter is omitted, the user is prompted when they print their first document to select color or monochrome printing.

The print menus are reset to the value that is specified each time IBM Daeja ViewONE is used.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printMono" value="true" />
```

printOriginalSizeAllowed parameter

This parameter determines whether the print original size option is available in the preferences menu.

If this parameter is set to false, then the print original size option is not displayed.

When the print original size option item is visible and selected, any printouts that are sent to the printer are printed at the displayed size rather than being scaled to fit the printer page.

If the page that is being printed is too large to fit on the printer page, and it is being printed at its original size, a message is displayed in the printed pages header with a warning. The text for message is set with the **imageTooLargePrintHeader** parameter.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printOriginalSizeAllowed" value="false" />
```

printOriginalSizeOverride parameter

This parameter determines whether print jobs are scaled to fit the printer page.

This parameter causes the menu item to remain in this state by disabling the menu.

If this parameter is used, the **Print original size** option in the **Preferences** menu is disabled. If the parameter is set to true, all print jobs are printed by using the print original size function. If the parameter is set to false, all print jobs are scaled to fit the printer page.

If this parameter is not included, the **Print original size** option in the **Preferences** menu is enabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printOriginalSizeOverride" value="false" />
```

printRangeAllowed parameter

This parameter determines whether the print range option is enabled.

When this parameter is set to true, then the print range keyboard shortcut and menu operations are enabled. That is, users can print a range of pages. The default value is true.

If this parameter is set to false, then the print range key and menu operations are disabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printRangeAllowed" value="false" />
```

printSelectedAllowed parameter

When this parameter is set to true, the print-selected key and menu operations are enabled. That is, the user can print selected pages. The default value is true.

When this parameter is set to false, the print-selected key and menu operations are disabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printSelectedAllowed" value="false" />
```

printStickyMenus parameter

When this parameter is set to true, the **Print sticky notes** and **Print sticky note text** preference menus are available. The default value is true.

When this parameter is set to false, the **Print sticky notes** and **Print sticky note text** preference menus are hidden.

Requirements

- IBM Daeja ViewONE Standard V3.0.728 or later
- IBM Daeja ViewONE Professional V1.0.728 or later
- No additional modules

Example

```
<param name="printStickyMenus" value="false" />
```

printVisibleAllowed parameter

This parameter determines whether the print visible menu operation is enabled.

When this parameter is set to true, the print visible menu operation is enabled. That is, the user can print only the visible area of the image. The default value is true.

When this parameter is set to false, the print visible menu operation is disabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="printVisibleAllowed" value="false" />
```

printWithUserSettings parameter

This parameter determines whether the user settings for rotation, flip, invert, brightness, contrast, and luminance are applied to the image before printing.

If this parameter is set to true, then the user's current settings are applied to the image before printing.

The default value is false.

Note: Monochrome images cannot be made darker when they are printed. However, the ViewONE Print Accelerator Module supports the "preserve black" feature for printing images. For more information on preserve black, see the **brightness** *parameter* topic.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="printWithUserSettings" value="true" />
```

Related reference:

"**brightness** parameter" on page 162

This parameter sets the default brightness, for all pages, of the display area.

reusePrinter parameter

This parameter determines if the default printer is used or if a previously named printer is used when the print dialog box is disabled.

This parameter is valid only when the **printDialog** parameter is set to false, and the ViewONE Print Accelerator Module is in use.

When this parameter is set to true and a printer was used previously, IBM Daeja ViewONE uses the last printer that the user selected.

When this parameter is set to true and a printer was not used previously, the default printer is used.

When this parameter is set to false, the default printer is always used.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
<param name="reusePrinter" value="false" />
```

Related reference:

“**printDialog** parameter” on page 215

This parameter determines whether the standard print dialog box is displayed.

separator<N> parameter

All the features that are available for normal documents, such as annotations, can also be used with page separators.

Separator pages are considered to be a separate document by IBM Daeja ViewONE. When IBM Daeja ViewONE prints the actual document, it interleaves pages from the separator document into the print job.

The separator pages can be in any of the formats supported by IBM Daeja ViewONE, for example, TIFF, BMP, JPG, or a simple text file (courier font is assumed). Additionally, the feature can be configured to use a single separator page (for example, one at the head of a print job) or a separator between every page that prints. You can also select which pages have separators that are associated with them. The latter might be useful when multiple documents are viewed as one large one, but where you would want a separator between the documents when printing.

You define the separator document by using the **separator<N>** HTML parameter or the **separatorList** parameter. You can use a file in any location that IBM Daeja ViewONE can access. The file location can include any server-side object that streams separator page data to IBM Daeja ViewONE.

If a selective print is requested, then IBM Daeja ViewONE prints the separator that corresponds to that page if one is defined. Examples of a selective print are print page, print range, or print selected, which can be selected with the IBM Daeja ViewONE right-click menu. If that page does not have a separator that is defined for it, then the first page from the separator document is used as a replacement, that is, **separator1**. This replacement ensures that all print jobs get a separator page if the **separator1** parameter is defined.

Because the separator pages are treated as a separate document, it is also possible to specify an annotations file to go with that document by using the **separatorAnnotationFile** HTML parameter. This parameter works in the same way as the **annotationFile** parameter.

If you use text files as separator pages, then you can include special markers within that separator file, which are replaced at run time. Valid markers are as follows:

<FILE> Inserts the URL of the page that follows the separator page.

<PAGE> Inserts the page number of the page that follows the separator page.

<DESCRIPTION>

Inserts the value that is specified by the **description** HTML parameter.

doc<*n*>

The **doc**<*n*> supports the following additional information:

<DOCINDEX>

Inserts the index number of the current document.

<DOCNUM>

Inserts the total number of documents currently specified.

<DOCREF>

Inserts the value that is specified by the appropriate **docRef**<*N*> HTML parameter.

All separator locations are relative to the codebase, just as they are for normal pages.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

The following example inserts a separator page in front of each of four pages:

```
<param name="page1" value="../docs/mixed/p1.tif" />
<param name="page2" value="../docs/mixed/p2.jpg" />
<param name="page3" value="../docs/mixed/p3.tif" />
<param name="page4" value="../docs/mixed/p4.jpg" />
<param name="separator1" value="../docs/separator/sep1.txt" />
<param name="separator2" value="../docs/separator/sep2.txt" />
<param name="separator3" value="../docs/separator/sep3.txt" />
<param name="separator4" value="../docs/separator/sep4.txt" />
```

This example results in the following order in the print job:

```
sep1.txt
p1.tif
sep2.txt
p2.jpg
sep3.txt
p3.tif
sep4.txt
p4.jpg
```

The following example uses a hyphen (-) as the **separator**<*N*> parameter value for pages 2 and 3 to print separator pages for only pages 1 and 4:

```
<param name="page1" value="../docs/mixed/p1.tif" />
<param name="page2" value="../docs/mixed/p2.jpg" />
<param name="page3" value="../docs/mixed/p3.tif" />
<param name="page4" value="../docs/mixed/p4.jpg" />
<param name="separator1" value="../docs/separator/sep1.txt" />
<param name="separator2" value="-" />
<param name="separator3" value="-" />
<param name="separator4" value="../docs/separator/sep4.txt" />
```

This example results in the following order in the print job:

```
sep1.txt
p1.tif
p2.jpg
p3.tif
sep4.txt
p4.jpg
```

The following example prints a single separator page at the beginning of the print job:

```
<param name="page1" value="../docs/mixed/p1.tif" />
<param name="page2" value="../docs/mixed/p2.jpg" />
<param name="page3" value="../docs/mixed/p3.tif" />
<param name="page4" value="../docs/mixed/p4.jpg" />
<param name="separator1" value="../docs/separator/sep1.txt" />
```

This example results in the following order in the print job:

```
sep1.txt
p1.tif
p2.jpg
p3.tif
p4.jpg
```

The following example uses the **filename** parameter in place of the **page<N>** parameter to print a single separator page at the beginning of the print job:

```
<param name="filename" value="../docs/mixed/fourpagedocument.tif" />
<param name="separator1" value="../docs/separator/sep1.txt" />
```

This example results in the following order in the print job:

```
sep1.txt
p1.tif
p2.jpg
p3.tif
p4.jpg
```

Related reference:

"separatorList parameter"

This parameter specifies a file that contains the list of separator pages to use for printing.

"annotationFile parameter" on page 79

This parameter specifies the annotations definition file that contains the list of the annotations to display and their associated parameters.

separatorList parameter

This parameter specifies a file that contains the list of separator pages to use for printing.

This parameter removes the need to specify each separator page for a document when you use the **separator<N>** parameter. This parameter is useful for large documents because it removes the need to specify each **page<N>** parameter.

This parameter also keeps the HTML constant. You change the source list instead of changing the HTML between different documents.

The specified file is a text file, and each entry in the file can be separated by a <CR>, <CR><LF>, or <LF> character. The location of the files can be absolute or relative to the codebase of the applet.

For example, the file `myseparators.lst` contains:

myseparators/sep1.txt
myseparators/sep2.txt
myseparators/sep3.txt

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="separatorList" value="mylist/list.lst" />
```

Related reference:

“**separator**<N> parameter” on page 227

All the features that are available for normal documents, such as annotations, can also be used with page separators.

Text file options

Text files are difficult to handle because there is no fixed industry standard for the format of a text document.

You use the viewer text file parameters to specify how to display and paginate text documents.

allowTextFilterDisplayAnyFile parameter

This parameter determines whether the text filter attempts to display an image that cannot be displayed by the other IBM Daeja ViewONE filters.

If this parameter is set to true, the text filter always attempts to display an image that cannot be displayed by other filters. This setting overrides the behavior of the following parameters: **textfilterMimeTypes**, **textfilterLocalFileExtensions**, and **routeDocs**.

The default for this parameter is false.

This parameter changes the handling of the text file to the behavior that existed in IBM Daeja ViewONE Standard Version 3.1.112 and IBM Daeja ViewONE Professional Version 1.1.112.

Requirements

- IBM Daeja ViewONE Standard V3.1.124 or later
- IBM Daeja ViewONE Professional V3.1.124 or later
- No additional modules

Example

```
<param name="allowTextFilterDisplayAnyFile" value="true" />
```

Related reference:

“**textfilterMimeTypes** parameter” on page 232

This parameter specifies additional MIME media types that the text filter can display if the type cannot be displayed by one of the other filters within IBM Daeja ViewONE.

“**textfilterLocalFileExtensions** parameter” on page 231

This parameter specifies the extensions of local files that the text filter displays.

“**routeDocs** parameter” on page 200

This parameter determines how unsupported image files are viewed.

ignoreTextFilterMimeTypes parameter

This parameter determines whether the viewer attempts to apply filters other than the text filter to files with text MIME types such as text/plain, text/html, or text/xml. You can use this parameter to prevent problems in displaying files that incorrectly are assigned a text MIME type.

The viewer uses the MIME type to choose the filter that is used to decode the requested document. By default, the viewer uses the text filter to display any file with a text MIME type. However, problems can occur in the display if the text MIME type is not appropriate for a file. For example, if an email is sent to the viewer with a MIME type of text/plain, the viewer displays the raw text data for the email.

To reduce the problems that are caused by incorrect text MIME types, you can set this parameter to true. This setting causes the viewer to first analyze a file with a text MIME type to determine the correct filter to apply. If all other file filters fail, the viewer finally attempts to use a text filter to display the file. If the text filter fails, the viewer returns an error message.

Important: This function cannot be used to override the display of other file types such as TIF files or PDF files.

Requirements

- IBM Daeja ViewONE Standard V3.1.148 or later
- IBM Daeja ViewONE Professional V1.1.148 or later
- No additional modules

Example

```
<param name="ignoreTextFilterMimeTypes" value="false" />
```

textEncoding parameter

This parameter determines whether a specific character encoding is used to display special characters in a text file.

Files with non-standard characters must be encoded correctly for the viewer to correctly display these characters. When a text file has the encoding embedded in the file, the viewer detects the encoding and uses it to display the characters correctly. However, in some cases the encoding is not contained within the file itself. When this parameter is set to a specific encoding type, the viewer uses that encoding type for files that do not contain the encoding data.

Requirements

- IBM Daeja ViewONE Standard V3.1.198 or later
- IBM Daeja ViewONE Professional V1.1.198 or later
- No additional modules

Example

```
<param name="textEncoding" value="UTF8" />
```

textFilterLocalFileExtensions parameter

This parameter specifies the extensions of local files that the text filter displays.

This parameter value accepts a comma-separated list of extensions. The default value is `txt,text,plain`. The text filter detects and displays any file with an extension of `.txt`, `.text`, or `.plain`.

This function does not override the display of other file types, such as TIFF or PDF.

Requirements

- IBM Daeja ViewONE Standard V3.1.124 or later
- IBM Daeja ViewONE Professional V3.1.124 or later
- No additional modules

Example

```
<param name="textFilterLocalFileExtensions" value="txt,log" />
```

textFilterMimeTypes parameter

This parameter specifies additional MIME media types that the text filter can display if the type cannot be displayed by one of the other filters within IBM Daeja ViewONE.

The text filter is used as a filter of last resort if none of the other filters within IBM Daeja ViewONE can display the file. When the file is being retrieved by using an HTTP Request, the MIME media type that is returned in the HTTP Response can help IBM Daeja ViewONE determine whether the file can be viewed.

This parameter value accepts a comma-separated list of MIME Media Types. You can use an `*` for the subtype, but partial matches on a value are not valid. For example, `application/*` and `application/octet-stream` are valid, but `application/oct*` is not valid.

Important: This function does not override the display of other file types, for example `image/tiff` or `application/pdf`.

By default, `text/*` is the only MIME media type that is used by the text filter. Any file type that has a main content type of `text` is detected by the text filter. It does not matter what the subtype is.

By default the parameter list is empty.

Requirements

- IBM Daeja ViewONE Standard V3.1.124 or later
- IBM Daeja ViewONE Professional V3.1.124 or later
- No additional modules

Example

```
<param name="textFilterMimeTypes" value="application/octet-stream" />
```

textFilterPageBreaks parameter

This parameter determines whether the viewer interprets FF (Form Feed - 12) characters in text files as a page break and paginates the document.

When this parameter is set to true and the **textFilterWrapping** parameter also is set to true, the viewer interprets FF (Form Feed - 12) characters in text files as a page break and paginates the document.

Tip: This parameter is disabled by default. Setting the **textFilterWrapping** parameter to true also sets this parameter to true.

Requirements

- IBM Daeja ViewONE Standard V3.1.142 or later
- IBM Daeja ViewONE Professional V1.1.142 or later
- No additional modules

Example

```
<param name="textFilterPageBreaks" value="true" />
```

Related reference:

“**textFilterWrapping** parameter” on page 234

This parameter determines how to display text files in the viewer. It also activates the **textFilterPageBreaks**, **textFilterPageHeight**, and **textFilterPageWidth** parameters.

textFilterPageHeight parameter

This parameter determines the number of lines to be displayed as a single page.

When this parameter is set to a value greater than 0, an automatic page break is added at line $N + 1$.

This parameter is disabled by default. The **textFilterWrapping** parameter must be set to true to activate this parameter.

Requirements

- IBM Daeja ViewONE Standard V3.1.142 or later
- IBM Daeja ViewONE Professional V1.1.142 or later
- No additional modules

Example

```
<param name="textFilterPageHeight" value="<N>" />
```

Related reference:

“**textFilterWrapping** parameter” on page 234

This parameter determines how to display text files in the viewer. It also activates the **textFilterPageBreaks**, **textFilterPageHeight**, and **textFilterPageWidth** parameters.

textFilterPageWidth parameter

This parameter determines the number of characters to be displayed on one line across the viewed page.

When this parameter is set to a value greater than 0, an automatic line break is added at character $N + 1$.

This parameter is disabled by default. The **textFilterWrapping** parameter must be set to true to activate this parameter.

Requirements

- IBM Daeja ViewONE Standard V3.1.142 or later
- IBM Daeja ViewONE Professional V1.1.142 or later
- No additional modules

Example

```
<param name="textFilterPageWidth" value="<N>" />
```

Related reference:

“**textFilterWrapping** parameter”

This parameter determines how to display text files in the viewer. It also activates the **textFilterPageBreaks**, **textFilterPageHeight**, and **textFilterPageWidth** parameters.

textFilterWrapping parameter

This parameter determines how to display text files in the viewer. It also activates the **textFilterPageBreaks**, **textFilterPageHeight**, and **textFilterPageWidth** parameters.

When this parameter is set to false, the default page break, length, and width are use. The default value is false.

Requirements

- IBM Daeja ViewONE Standard V3.1.142 or later
- IBM Daeja ViewONE Professional V1.1.142 or later
- No additional modules

Example

```
<param name="textFilterWrapping" value="true" />
```

Viewing options

Use the viewing options parameters to display buttons on the toolbar.

abortSaveOnShutdown parameter

This parameter determines whether the IBM Daeja ViewONE applet can close when a save operation is in progress.

If set to false, this parameter prevents the closure of the applet until any running save operation is finished.

If set to true, a save operation ends as soon as the applet is closed.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V3.0.54 or later
- IBM Daeja ViewONE Professional V1.0.54 or later
- No additional modules

Example

```
<param name="abortSaveOnShutdown" value="false" />
```

allButtons parameter

This parameter determines whether all buttons in the top toolbar and the status bar are visible.

If this parameter is set to false, the only buttons that are visible in the top toolbar are the **Hide Annotations Toolbar** button and any custom buttons. All other buttons are hidden.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="allButtons" value="false" />
```

allKeys parameter

This parameter determines whether the IBM Daeja ViewONE keyboard shortcuts are enabled.

Important: The parameter does not disable shortcut keys that enabled by the **magOptions** parameter or the **flipOptions** parameter. To disable these shortcut keys, set the parameter value to false.

The default for the **allKeys** parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="allKeys" value="false" />
```

allMenus parameter

This parameter determines whether the IBM Daeja ViewONE menus are available.

This parameter does not affect the display of the **Help** menu.

The default value for this parameter is true, which makes all menus available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="allMenus" value="false" />
```

autoScroll parameter

This parameter determines whether the image scrolls automatically when an area to zoom is selected and the mouse is dragged outside the image area.

When this parameter is set to true, the image is scrolled automatically. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
<param name="autoScroll" value="false" />  
<param name="autoScroll " value="true" />
```

cacheMenus parameter

This parameter determines whether the **Cache** submenu of options is displayed in the preference menu.

When this parameter is set to false, the **Cache** submenu of options is not displayed in the preference menu.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard V3.1.38 or later
- IBM Daeja ViewONE Professional V1.1.38 or later
- No additional modules

Example

```
<param name="cacheMenus" value="false" />
```

clipOptions parameter

This parameter determines whether the clipboard menu and keyboard options are enabled.

When this parameter is set to true, the clipboard menu and keyboard options are enabled. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="clipOptions" value="true" />
```

defaultLinkViewer parameter

If the **docCompare** parameter is set to true, and there is more than one viewer on the same HTML page with these settings, then setting this parameter to true means that the viewers are linked. If the viewers are linked, you do not need to activate the link with the link viewer buttons.

Any transformations in one viewer are reflected in the other.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="defaultLinkViewer" value="true" />
```

Related reference:

“**docCompare** parameter” on page 238

This parameter determines whether the **LinkViewer** button is displayed. If more than one viewer is loaded in an HTML page and this button is selected in each viewer, any transformation the user makes in one viewer is reflected in the linked viewers.

defaultZoomDoc parameter

This parameter determines whether the view adjusts the size of the magnifier window or the document when the magnifier window is visible and the user uses the zoom options.

If you set this parameter to false and the external or internal magnifier window is visible, the viewer increases and decreases the size of the magnifier window when the user zooms in or out.

If you set this parameter to true and the external or internal magnifier window is visible, the viewer increases and decreases the document zoom factor when the user zooms in or out.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
<param name="defaultZoomDoc" value="true" />
```

disableShutdownDialogs parameter

This parameter determines whether the viewer displays dialog boxes after the main browser window is closed. For example, this parameter determines whether the viewer displays the dialog box that informs the user of an annotation save failure.

If this parameter is set to true, the viewer is prevented from displaying any dialog boxes after the main browser window is closed.

The default for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard V3.0.530 or later
- IBM Daeja ViewONE Professional V1.0.530 or later
- No additional modules

Example

```
<param name="disableShutdownDialogs" value="true" />
```

displayColorProgressively parameter

This parameter determines whether the viewer displays color images as the images are processed.

The default value for this parameter is false, which hides color images as they are processed. The images are displayed after processing is completed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="displayColorProgressively" value="true" />
```

docCompare parameter

This parameter determines whether the **LinkViewer** button is displayed. If more than one viewer is loaded in an HTML page and this button is selected in each viewer, any transformation the user makes in one viewer is reflected in the linked viewers.

This feature is useful when a user wants to compare two or more documents on screen at the same time. For example, if the **LinkViewer** button enabled and the user rotates the image in the first viewer 90 degrees clockwise, the image in the second viewer is simultaneously rotated clockwise.

The user must select the **LinkViewer** button in each viewer that is to be linked.

The default value for this parameter is false, which hides the **LinkViewer** button.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="docCompare" value="true" />
```

displayDocButtons parameter

This parameter, when set to false, hides the document buttons that are displayed when the **doc<N>** parameter is used. The document buttons are automatically added when the **doc<N>** parameters are used to enable the user to change a document.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="displayDocButtons" value="false" />
```

Related reference:

“**doc<N> parameter**” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

displayMonoProgressively parameter

This parameter determines whether the viewer displays monochrome images as they are processed.

If this parameter is set to false, the viewer does not display monochrome images as they are processed. Instead, the viewer displays the images after all processing is completed.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="displayMonoProgressively" value="true" />
```

draggingEnabled parameter

This parameter determines whether users can drag images to pan the images to the left and right.

The default value for this parameter is true, which enables users to drag images.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="draggingEnabled" value="false" />
```

emptyClipboardOnClose parameter

When this parameter is defined, the user preference in the clipboard menu for emptying the clipboard when a document is closed is disabled and overridden with the value specified by this parameter.

If this parameter is not defined, the user can choose the option to take from the clipboard menu. If this parameter is defined by the owner of the system, the menu is not available to the user.

The default for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="emptyClipboardOnClose" value="true" />
```

emptyOnClose parameter

This parameter determines whether IBM Daeja ViewONE clears its cache when a document is closed and multiple documents are loaded in the viewer during the same session.

If this parameter is set to true, IBM Daeja ViewONE clears its cache when a document is closed. If the user then reopens this document, IBM Daeja ViewONE must reload the document from the server. Therefore, this parameter is useful when images are likely to be modified on the server while users are viewing them.

Important: If the HTML page is refreshed to load a new document, the viewer is shut down and restarted. Because the viewer cache is always emptied when the viewer is shut down or started up, the cache is automatically cleared and repopulated.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="emptyOnClose" value="true" />
```

enableDocButtons parameter

This parameter determines whether the document buttons are disabled in the viewer when the **doc<N>** parameter is used.

Set this parameter to false to disable the document buttons in the viewer.

The default value for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- No additional modules

Example

```
<param name="enableDocButtons" value="false" />
```

Related reference:

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

externalMagnifier parameter

This parameter determines whether the magnifier window appears to be an internal window in the display area when the user clicks the **Magnifier** button.

The default value for this parameter is false, which displays the magnifier as an internal window. If the parameter is set to true, the magnified area is displayed in a new window.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="externalMagnifier" value="true" />
```

flickScrollEnabled parameter

While you drag a page to scroll, release the mouse button before you stop the mouse to flick the page. The page keeps scrolling and slows down gradually unless you click the mouse button again.

The default value for this parameter is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="flickScrollEnabled" value="false" />  
<param name="flickScrollEnabled" value="true" />
```

focusBorder parameter

This parameter determines whether a border is displayed around a viewer that has keyboard focus.

The default value for this parameter is true, which displays the border.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="focusBorder" value="false" />
```

helpTarget parameter

This parameter determines where the web page that is specified by the **helpURL** parameter is opened.

You can specify the following values for this parameter.

Value	Description
<code>_blank</code>	The page opens in a new browser window.
<code>_self</code>	The page opens in the same window or frame as the viewer.
<code>_top</code>	The page opens in the top-level frame in the same the window as the viewer. If frames are not used, this value is the same as <code>_self</code> .
<code>_parent</code>	The page opens in the parent of the frame that contains the viewer. If frames are not used, this value is the same as <code>_self</code> .
<i>window name</i>	The page opens in the specified window. If this window does not exist, the viewer creates a top-level window with this name.

The default value for this parameter is `_blank`.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="helpTarget" value="_blank" />
```

Related reference:

“[helpURL parameter](#)”

This parameter specifies a URL to a web page that contains help documentation.

helpURL parameter

This parameter specifies a URL to a web page that contains help documentation.

If you set this parameter, the **Help** button and a **Help** menu item to access the URL become available in the viewer.

You can specify an absolute or relative URL to the web page. You can use the **helpTarget** parameter to specify how the web page is opened.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="helpURL" value="http://www.mysite.com/help.html" />
```

htmlListShortcutEnabled parameter

This parameter determines whether the user can press the Shift+T keyboard shortcut to display a list of the HTML parameters that are processed by the viewer.

The default value for this parameter is `true`, which enables the Shift+T keyboard shortcut.


Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="htmlListShortcutEnabled" value="false" />
```

Related reference:

 [Accessibility features of IBM Daeja ViewONE](#)

httpReadTimeout parameter

When IBM Daeja ViewONE Professional makes a server-side request, **httpReadTimeout** sets the amount of time that the client waits for a response from the server.

If the server does not respond within the set timeout, the request to the server is aborted.

The default value is 60 seconds.

Requirements

- IBM Daeja ViewONE Standard, V4.1.1 or later
- IBM Daeja ViewONE Professional, V4.1.1 or later

Example

```
<param name="httpReadTimeout" value="60s" />
```

initialFocus parameter

This parameter determines whether the viewer attempts to obtain focus when it is initialized.

The default value for this parameter is true, which enables the viewer to attempt to obtain focus.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="initialFocus" value="false" />
```

keepAlive parameter

This parameter determines a mechanism for ensuring that network connections remain open while the viewer is in use. The viewer can either send regular requests to a specified URL or generate a JavaScript event to a specified event handler.

If you specify a URL for the parameter value, the viewer issues an HTTP:GET request to that URL.

If you specify a JavaScript event for the parameter value, the viewer generates an event with ID 41.

To specify the time interval for the keep alive actions, use the **keepAliveTime** parameter.

Important: Some environments require regular network activity to keep network connections open.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="keepAlive" value="myurl.asp" />  
<param name="keepAlive" value="javascript::mytext" />
```

Related reference:

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

“Events IDs and descriptions” on page 316

IBM Daeja ViewONE can signal many different events to the event handler.

“Obtain additional information from an event” on page 323

Starting with IBM Daeja ViewONE, V4.0.38, the event handling is extended to allow additional information to be passed back with the event.

“**keepAliveTime** parameter”

This parameter determines the time interval at which the viewer performs the action that is specified by the **keepAlive** parameter. Specify the value of this parameter in seconds.

keepAliveTime parameter

This parameter determines the time interval at which the viewer performs the action that is specified by the **keepAlive** parameter. Specify the value of this parameter in seconds.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="keepAliveTime" value="30" />
```

Related reference:

“**keepAlive** parameter” on page 243

This parameter determines a mechanism for ensuring that network connections remain open while the viewer is in use. The viewer can either send regular requests to a specified URL or generate a JavaScript event to a specified event handler.

keepScroll parameter

This parameter determines if the viewer maintains the zoom and scroll when pages in the document are changed.

If this parameter is set to true, and providing the pages in a document do not differ by 10% in either width or height, the viewer maintains the zoom and scroll when pages in the document are changed.

The default value is false; the scroll bars center themselves for each new page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="keepScroll" value="true" />
```

magnifierFollowMouse parameter

This parameter determines if the magnifier follows the mouse cursor.

When this parameter is set to false, the user must hold the left mouse button to move the magnifier window.

When this parameter is set to true, the magnifier follows the mouse cursor. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
<param name="magnifierFollowMouse" value="false" />
```

magnifierPersistent parameter

This parameter makes magnifier use quicker for users when they are continually magnifying the same area on images in a set of documents.

When this parameter is set to true, the magnifier window is automatically reinstated in its last form (same size, position, magnitude) when a new document is opened.

The default value is false. The default mode is to hide the magnifier window when documents are changed, so the user must manually show it each time.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="magnifierPersistent" value="true" />
```

menuNewWindow parameter

This parameter determines whether the **New Window** menu is available.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="menuNewWindow" value="false" />
```

mouseRightButtonEnabled parameter

This parameter determines whether the right mouse button function for viewer buttons is enabled.

When this parameter is set to false, the right mouse button function is disabled for viewer buttons.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="mouseRightButtonEnabled" value="false" />
```

newWindow parameter

The parameter determines if a new window opens.

When this parameter is set to true, IBM Daeja ViewONE starts and then opens into new window mode as if the user clicked **New Window**.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="newWindow" value="true" />
```

nwViewMode parameter

This parameter specifies the initial view mode when a document is displayed within a **New Window**.

The following table lists the option values.

Value	Description
fullpage	The document is opened with only the first page visible. This option is the default value.
thumbsonly	The document is opened with only the thumbnails visible.
twopage	The document is opened with two pages displayed side by side.
thumbsleft	The document is opened with thumbnails on the left and the remaining area is displayed on the first page.
thumbsright	The document is opened with thumbnails on the right and the remaining area is displayed on the first page.
thumbstop	The document is opened with thumbnails at the top and the remaining area is displayed on the first page.
thumbsbottom	The document is opened with thumbnails at the bottom and the remaining area is displayed on the first page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="nwViewMode" value="fullpage" />
<param name="nwViewMode" value="thumbsonly" />
<param name="nwViewMode" value="twopage" />
<param name="nwViewMode" value="thumbsleft" />
<param name="nwViewMode" value="thumbsright" />
<param name="nwViewMode" value="thumbstop" />
<param name="nwViewMode" value="thumbsbottom" />
```

obfuscate parameter

This parameter determines whether cached image files are stored in a format that prevents users from viewing or copying the files directly.

When this parameter is set to true, IBM Daeja ViewONE uses a non-disclosed file format to store cached image files to preventing users from viewing or copying them directly.

This parameter might be useful where access to images is a sensitive issue or users are charged for image viewing or printing.

Restriction: The ViewONE Universal Viewing Module cannot read the cached file format that is used when the **obfuscate** parameter is set to true. See the **obfuscateUV** parameter on how to deal with the ViewONE Universal Viewing Module and the **obfuscate** parameter.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="obfuscate" value="true" />
```

Related reference:

“**obfuscateUV** parameter” on page 261

The ViewONE Universal Viewing Module cannot process the cached file format that is used when the **obfuscate** parameter is set to true. To get around this issue, use the **obfuscateUV** parameter.

pageButtons parameter

This parameter determines whether the toolbar includes page buttons.

When this parameter is set to true, the page buttons are displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="pageButtons" value="false" />  
<param name="pageButtons" value="true" />
```

pageKeys parameter

This parameter determines whether the keyboard shortcuts for page operations are enabled.

When this parameter is set to true, the keys are enabled. The default value is true.

When this parameter is set to false, the keys are disabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="pageKeys" value="false" />  
<param name="pageKeys" value="true" />
```

Related reference:

 Accessibility features of IBM Daeja ViewONE

pageMenus parameter

This parameter determines whether the page menus are available when you click the middle or right mouse button.

When this parameter is set to true, the page menus are available. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="pageMenus" value="false" />  
<param name="pageMenus" value="true" />
```

preferenceMenus parameter

This parameter determines whether the preference menus are available when the user clicks the middle or right mouse button.

When this parameter is set to true, the menus are enabled. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="preferenceMenus" value="false" />  
<param name="preferenceMenus" value="true" />
```

saveExtension parameter

This parameter determines the file name extension for a saved document.

When this parameter is included, all files that are saved have the specified extension.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="saveExtension" value="tif" />
```

scrollbars parameter

This parameter determines whether scroll bars are displayed when the image is scaled to a size larger than the display area.

When this parameter is set to true, the scroll bars are displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="scrollbars" value="false" />
<param name="scrollbars" value="true" />
```

selectKeys parameter

This parameter specifies whether the keyboard shortcuts for page selections are enabled.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="selectKeys" value="false" />
```

selectMenus parameter

This parameter determines whether the select menus are available when a user clicks the middle or right mouse buttons.

When this parameter is set to true, the menus are enabled. The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="selectMenus" value="false" />
```

serverEventInterest parameter

This parameter is used with the **serverEventHandler** parameter to filter which IBM Daeja ViewONE events are sent to the server event logger.

By default, when the **serverEventHandler** parameter is specified, all events are sent to the server event logger.

The list of available events to log is the same as the one defined by the client event handler. For more information, see the *IBM Daeja ViewONE JavaScript Reference Manual*.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- No additional modules

Example

```
<param name="serverEventInterest" value="5,22" />
```

Related reference:

“**serverEventHandler** parameter” on page 177

This parameter determines whether event handling is enabled for use with the JavaScript API.

serverFileMonitor parameter

This parameter allows a server object to monitor when a user saves or prints a specific file.

If you want a server script, object, or executable file to receive notification of how many times a document was saved or printed, set this parameter to the URL path that represents the object.

In the following Perl script example, IBM Daeja ViewONE tags the following text onto the end of the value that is specified:

```
save=n&print=n
```

Where *n* is a number that represents the total number of operations.

The monitor URL is called when a document is closed, including when a user changes web pages from a page that contains IBM Daeja ViewONE.

Requirements

- IBM Daeja ViewONE Standard V4.0.14 or later
- IBM Daeja ViewONE Professional V4.0.14 or later
- No additional modules

Example

```
<param name="serverFileMonitor" value="http://myserver.com/fileMonitor.pl?" />
```

synchronizeThumbs parameter

This parameter determines whether the thumbnail view scrolls to show the page that is currently displayed in the image area.

When this parameter is set to true, the thumbnail view scrolls to show the page that is displayed in the image area.

The default value is false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="synchronizeThumbs" value="true" />
```

systemInfoShortcutEnabled parameter

This parameter determines whether the system information keyboard shortcut (Shift+Y) is enabled.

When this parameter is set to false, the system information keyboard shortcut (Shift+Y) is disabled.

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="systemInfoShortcutEnabled" value="false" />
```

thumbDbClick parameter

This parameter specifies the behavior when a user double-clicks a thumbnail. By default when a user double-clicks a thumbnail, that page is displayed as a single full page.

When this parameter is set to *FullPage*, the page is displayed as a single full page when a user double-clicks a thumbnail.

When this parameter is set to *Select*, the page is selected when a user double-clicks a thumbnail.

When this parameter is set to *NewWindow*, a new window with that page is displayed when a user double-clicks a thumbnail.

The default value is *FullPage*.

Requirements

- IBM Daeja ViewONE Standard V3.1.130 or later
- IBM Daeja ViewONE Professional V1.1.130 or later
- No additional modules

Example

```
<param name="thumbDbClick" value="Select" />
```

thumbDbClickSelect parameter (Deprecated)

This parameter determines the behavior when a user double-clicks a thumbnail.

Deprecated: This parameter is deprecated as of IBM Daeja ViewONE Standard Version 3.1.130 and IBM Daeja ViewONE Professional Version 1.1.130. Use the **thumbDbClick** parameter with a value *Select* instead. However, this parameter continues to work.

When this parameter is set to true and a user double-clicks a thumbnail, that page is marked as selected instead of being displayed in full page mode.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thumbDbClickSelect" value="true" />
```

Related reference:

“thumbDbClick parameter” on page 252

This parameter specifies the behavior when a user double-clicks a thumbnail. By default when a user double-clicks a thumbnail, that page is displayed as a single full page.

thumbsHeight parameter

This parameter specifies the height of thumbnails in pixels.

The default value is 120.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thumbsHeight" value="200" />
```

thumbsWidth parameter

This parameter specifies the width of thumbnails in pixels.

The default value is 120.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="thumbsWidth" value="200" />
```

useBrowserAlertOnShutdown parameter

IBM Daeja ViewONE Professional was updated so that the browser displays the Leave this Page/Stay on this Page dialog box. The dialog box is displayed to inform the user that they have unsaved changes on the page.

Later versions of web browsers do not allow applications to display their own dialog boxes when the browser page is closing.

If this parameter is set to false, IBM Daeja ViewONE uses the behavior from before V4.1.0.2 and displays its save confirmation dialog box to users when they close the page and they have unsaved annotations.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Professional V4.1.0.2 or later
- No additional modules

Example

In the following example, the parameter is set to true, so the dialog box is shown.

```
<param name="useBrowserAlertOnShutdown" value="true" />
```

useErrorFiles parameter

IBM Daeja ViewONE V3.0.144 replaced the error files `openError.txt` and `notFound.txt` with auto-translated error messages.

However, if you want to revert to using the error files so that you can customize these messages yourself, then use this parameter with the value set to `true`.

The default value for this parameter is `false`.

Requirements

- IBM Daeja ViewONE Standard V3.0.144 or later
- IBM Daeja ViewONE Professional V1.0.144 or later
- No additional modules

Example

```
<param name="useErrorFiles" value="true" />
```

viewButtonsDetail parameter

This parameter can make any of the individual view buttons invisible.

The presence of a name in the list, for example *thumbs*, indicates that the associated button is to be made visible. An absence of a name indicates that the associated button is to be made invisible.

The view button identifiers: *page*, *thumbs*, *twopage*, *left*, *bottom*, *right*, and *top*. The *twopage* button is not supported in IBM Daeja ViewONE Virtual.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="viewButtonsDetail" value="page, thumbs" />
```

viewButtons parameter

This parameter specifies whether the toolbar includes view buttons.

The default for this parameter is `true`.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="viewButtons" value="false" />
```

viewKeys parameter

This parameter specifies whether the keyboard shortcuts for view operations are enabled.

When set to true, this parameter indicates that the keys are enabled and false indicates that they are not.

The keys are as follows:

- View page F2
- View thumbnails F3
- View two pages F4
- View thumbnails: Left F5
- View thumbnails: Bottom F6
- View thumbnails: Right F7
- View thumbnails: Top F8

The default value is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="viewKeys" value="false" />
```

viewMenus parameter

This parameter specifies whether the view menus are available by using the middle or right mouse button.

The default for this parameter is true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="viewMenus" value="false" />  
<param name="viewMenus" value="true" />
```

viewMode parameter

This parameter specifies the initial view mode when a document is displayed.

Optional values are in the following table.

Value	Description
<i>fullpage</i> (default)	The document is opened with the first page visible only.
<i>thumbsonly</i>	The document is opened with the thumbnails visible only.
<i>twopage</i>	The document is opened with two pages displayed side by side. This parameter is not supported in IBM Daeja ViewONE Virtual.

Value	Description
<i>thumbsleft</i>	The document is opened with thumbnails on the left and the remaining area displays the first page.
<i>thumbsright</i>	The document is opened with thumbnails on the right and the remaining area displays the first page.
<i>thumbstop</i>	The document is opened with thumbnails at the top and the remaining area displays the first page.
<i>thumbsbottom</i>	The document is opened with thumbnails at the bottom and the remaining area displays the first page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
<param name="viewMode" value="fullpage" />
<param name="viewMode" value="thumbsonly" />
<param name="viewMode" value="twopage" />
<param name="viewMode" value="thumbsleft" />
<param name="viewMode" value="thumbsright" />
<param name="viewMode" value="thumbstop" />
<param name="viewMode" value="thumbsbottom" />
```

zoom, xScroll, and yScroll parameters

These parameters set the zoom factor and the scroll positions of a displayed image.

The **zoom** parameter sets the zoom factor of the displayed image. 1.0 = 100% or 1 image pixel = 1 screen pixel. 2.0 = 200% or 1 image pixel = 2 screen pixels. The **xScroll** and **yScroll** set the scroll position (in screen pixels) of the image.

Important: The **zoom** parameter can be specified on its own, but **xScroll** and **yScroll** do not take effect unless all three parameters are specified.

Important: These parameters cannot be used when the **viewMode** parameter is set to **thumbsonly** or **twopage**.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
<param name="zoom" value="1.0" />
<param name="xScroll" value="0" />
<param name="yScroll" value="100" />
```

Related reference:

“viewMode parameter” on page 255

This parameter specifies the initial view mode when a document is displayed.

ViewONE Document Streaming Server Module options

Use the parameters to enable and configure the streamer and cookies.

cookieAllowedDomainList parameter

This parameter defines a comma-separated list of hosts that the client IBM Daeja ViewONE Professional is allowed to send cookies to from its containing HTML page. This parameter is required when the source document repository uses cookies to identify logged in users.

This comma-separated list contains the host names of the server that is running the ViewONE Document Streaming Server Module server-side code. This list then allows cookies that are intended for the document repository to be forwarded correctly and allow authorization for requested documents.

Requirements

- IBM Daeja ViewONE Professional V1.1.184 or later
- ViewONE Document Streaming Server Module

Example

```
<param name="cookieAllowedDomainList" value="streamerhost,filehost" />
```

documentCookies parameter

This parameter determines the cookies that are used to request the next document from the document server. You might set this parameter in an environment such as Ajax in which a user might be logged in to a resource remotely without a refresh of the current viewer HTML page.

To specify the value for this parameter, enter a list of cookies that are separated by semicolons.

Requirements

- IBM Daeja ViewONE Standard V3.1.172 or later
- IBM Daeja ViewONE Professional V1.1.172 or later
- No additional modules

Example

```
<param name="documentCookies" value="mysessionid=29;userid=abc" />
```

streamerEnabled parameter

This parameter determines whether HTTP requests for documents and images are directed to the ViewONE Document Streaming Server Module.

When this parameter is set to true, all HTTP requests for documents and images that are sent from the viewer by using either HTML parameters or JavaScript methods are directed to the URL that is specified in the **streamerURL** parameter.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.1.184 or later
- ViewONE Document Streaming Server Module

Example

```
<param name="streamerEnabled" value="true" />
```

Related reference:

“**streamerURL** parameter”

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

streamerURL parameter

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

When this parameter is set to the location of the installed streamer and the **streamerEnabled** parameter is set to true, the URL is appended to any file requests. Any authentication cookies that are supplied with the request are passed to the streamer and to the image repository.

Requirements

- IBM Daeja ViewONE Professional V1.1.184 or later
- ViewONE Document Streaming Server Module

Example

```
<param name="streamerURL" value="http://www.mywebapp.com/streamer" />
```

ViewONE Permanent Redaction Server Module options

Use the parameters to direct redacted output files, and enable and configure burn options.

annotationBurnURL parameter

This parameter specifies the URL of the burn servlet.

When this parameter is defined, the viewer automatically displays an extra **Burn** button in the annotation toolbar. The user clicks this button to start the burn process.

Important: For burning to process, you must specify both the **annotationDefaults** parameter and the **annotationBurnURL** parameter.

For performance reasons, this action does not burn the annotations into an image on the client-side. Instead, it burns the annotations into an image on the server-side and marks the annotations on the client-side as uneditable.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- IBM Daeja ViewONE Virtual, any version
- ViewONE Permanent Redaction Server Module

Example

```
<param name="annotationBurnURL" value="http://myserver/burner" />
```

Related reference:

“**annotationDefaults** parameter” on page 259

This parameter specifies the annotation types that are burnable.

annotationDefaults parameter

This parameter specifies the annotation types that are burnable.

Tip: You can also use this parameter to specify the default security settings for specified annotations. For more information, see “**annotationDefaults** parameter” on page 109.

The value for this property uses the following format:

```
value="[type {burnable=burnable_state}]"
```

If you are also using this parameter to set the default security settings, the value uses the following format:

```
value="[type {burnable=burnable_state}];  
type {init='1,1,0,1,0,0'}"
```

The *type* variable indicates the type of annotations to which the default line color applies. If the output document is a TIFF file, you can specify any of the following annotation types as burnable:

- arrow
- freehand
- highlight
- highlightPolygon
- line
- openPoly
- oval
- rectangle
- redact
- redactPolygon
- stamp
- text

If the output document is a PDF file, you can specify only the redact type or redactPolygon type as burnable.

To indicate that an annotation type is burnable, set the *burnable_state* variable to true. The default value for all annotation types is false.

You can use this parameter with the `setRedactionIsSemiTransparent()` JavaScript method to specify a set of annotation types that are to be semitransparent when the method is set to true.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="annotationDefaults" value="redact {burnable=true} text{burnable=true}  
line{burnable=true} arrow{burnable=true}">
```

Related reference:

“**annotationDefaults** parameter” on page 109

This parameter determines the default security settings for specified annotations.

burnIncrementalAnnotations parameter

This parameter specifies whether the annotation data sent from IBM Daeja ViewONE Professional to the burn servlet contains changes since the last burn operation or all the changes since opening the document.

When this parameter is set to true, only incremental changes to the data since the last burn operation are sent. The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="burnIncrementalAnnotations" value="false" />
```

burnUseGZip parameter

This parameter specifies whether the annotation data sent from IBM Daeja ViewONE Professional to the burn servlet is compressed.

When this parameter is set to true, the Content-Encoding type of the data is "gzip". When this parameter is used, the servlet must decompress the annotation data before it is passed to one of the **burnAnnotations** methods on the *jiBurner* object.

The default value is false.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="burnUseGZip" value="true" />
```

redactionHttpReadTimeout parameter

This parameter determines whether the redacted output file is stored locally or in the repository after a burn operation completes.

When this parameter is set to true and the burn operation completes, the user is prompted to open or save the file locally. The default value is true.

When this parameter is set to false, the user receives no output file. The output file is stored in the back-end repository.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="redactionHttpReadTimeout" value="false" />
```

redactionHttpReadTimeout parameter

This parameter determines the time in milliseconds that the applet waits for the redaction server to send a status update before the applet assumes that the process failed.

By default, the applet waits 60000 milliseconds (60 seconds) to send a status update.

Requirements

- IBM Daeja ViewONE Professional V4.1.2, or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="redactionHttpReadTimeout" value="120000" />
```

showAnnotationBurnPrompt parameter

This parameter specifies whether a confirmation prompt is displayed when a user clicks **burn**.

When this parameter is set to true, a confirmation prompt is displayed. The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.1.106 or later
- ViewONE Permanent Redaction Server Module

Example

```
<param name="showAnnotationBurnPrompt" value="false" />
```

ViewONE Universal Viewing Module options

IBM Daeja ViewONE Professional introduced an optional ViewONE Universal Viewing Module.

The ViewONE Universal Viewing Module comes with the following optional parameters.

obfuscateUV parameter

The ViewONE Universal Viewing Module cannot process the cached file format that is used when the **obfuscate** parameter is set to true. To get around this issue, use the **obfuscateUV** parameter.

Use this parameter if the **obfuscate** parameter is set to true or if you are using the ViewONE Document Streaming Server Module as part of your application configuration. Users can view or print the documents that are rendered by the ViewONE Universal Viewing Module, but the streaming servlet obfuscates all documents that are passed through it.

If you set this parameter to false, documents that are displayed use the ViewONE Universal Viewing Module and are cached by using their native format. This setting can be a problem where access to images is a sensitive issue or where users are charged for image viewing or printing.

The default value is true.

Requirements

- IBM Daeja ViewONE Professional V1.1.52 or later
- ViewONE Universal Viewing Module

Example

```
<param name="obfuscateUV" value="false" />
```

Related reference:

“**obfuscate** parameter” on page 247

This parameter determines whether cached image files are stored in a format that prevents users from viewing or copying the files directly.

uvTextDirectionMethod parameter

This parameter is used to ensure that documents loaded with the ViewONE Universal Viewing Module is displayed right to left where the language is a right to left language such as Arabic or Hebrew.

Valid values:

- 0, which does nothing
- 2, which uses Unicode

The default value for this parameter is 0.

Requirements

- IBM Daeja ViewONE Professional V4.0.18 or later
- ViewONE Universal Viewing Module

Example

```
<param name="uvTextDirectionMethod" value="2" />
```

IBM Daeja ViewONE color scheme text

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

The following table lists the color names that you can use in place of RGB values. These names are not case-sensitive.

Color name	RGB value
Black	0, 0, 0
Blue	0, 0, 255
Cyan	0, 255, 255
DarkGray	64, 64, 64
Gray	128, 128, 128
Green	0, 255, 0
LightGray	192, 192, 192
Magenta	255, 0, 255
Orange	255, 200, 0
Pink	255, 175, 175
Red	255, 0, 0
White	255, 255, 255
Yellow	255, 255, 0

IBM Daeja ViewONE Default Stamps

The viewer's default stamps are listed in the following table. The stamps are used on documents that need to be tagged with a predefined set of annotations.

annotationStamp parameter value	annotationStampProperties parameter value
<date>	<menu=Today's date>
RECEIVED	<menu=Received>
PAID	<menu=Paid>
APPROVED	<menu=Approved>
* REJECTED *	<menu=Rejected>
* VOID *	<menu=Void>

Supported fonts

The default fonts for IBM Daeja ViewONE are listed in the following table.

Name	Plain	Bold
Courier	x	x
Arial	x	x
Serif	x	
SansSerif	x	
Tahoma	x	

WANG Compliant Annotation Types

The following table contains the WANG compliant annotation types.

Compliant	Not compliant
Line	Arrow
Text	Note
Solid Text	Hyperlink
Highlight	Polygon
Rectangle	Open Polygon
Square	Redaction Polygon
Redaction	Oval
Line	Circle
Text Stamp	
Image Stamp	
Ruler	
Angle	

Hyperlinks on annotations are not supported for WANG annotations.

The Ruler and Angle IBM Daeja ViewONE annotations do not display as a ruler and angle in a WANG annotation viewer, for example, Kodak Imaging. Instead, the Ruler annotation is represented as a line and the Angle annotation is represented as two lines.

Configuring the viewer by using JavaScript API methods

The IBM Daeja ViewONE JavaScript API offers an alternative mechanism to configure and control IBM Daeja ViewONE Professional and IBM Daeja ViewONE Virtual.

About this task

In many cases, you do not need to use the JavaScript API. Instead, you can implement most configuration parameters by using simple HTML. However, where a more dynamic operation is required, then the JavaScript API is ideal. For example, you might use the JavaScript API to configure parameters during the operation of IBM Daeja ViewONE or to change a document without reloading a web page or to rotate a page.

Additionally, traditional web-viewing applications habitually reload a web page to view or change the currently viewed image or document. To avoid refreshing the web page, you can use the JavaScript operations to open and close documents or images. This feature can result in significant performance improvements because delays that are caused by refreshing the page and reloading and restarting the viewer are avoided.

You set the server properties in the parameter configuration file, *servlet-name.properties*. The name of the parameter configuration file is specified in the *web.xml* file. This file is deployed on the web server with the application itself, for example, in the WEB-INF folder.

Use the following example format for adding a parameter to a parameter configuration file:

```
ViewONE.name(int, other)
```

The examples might be used within functions of a JavaScript program or directly as event handlers to such things as buttons and hyperlinks.

Procedure

To use the JavaScript methods:

1. Replace name with the method name and int, other with any required value.

Important: Except for file names and hyperlink addresses, all parameters are not case-sensitive. However, some browsers are case-sensitive. Therefore, you must enter method names as shown to ensure that the methods are called correctly.

2. Use the internet URL address format to specify file names and hyperlink addresses, for example, <http://mysite/myimage.tif>. If any part of the address before *myimage.tif* is not included, then the viewer assumes a base address that is the same as the viewer location (the codebase).

Results

Important: Some JavaScript methods are disabled by default and can be enabled only by setting the **JavaScriptExtensions** or **annotationJavaScriptExtensions** HTML parameter to true as follows:

```
<param name="JavaScriptExtensions" value="true">
<param name="annotationJavaScriptExtensions" value="true">
```


These methods are disabled to prevent unauthorized users from attempting to manipulate IBM Daeja ViewONE through JavaScript methods to obtain access to secure information, documents, and annotations.

The methods that are restricted in this manner are clearly marked within this document.

Related tasks:

“Configuring the viewer by using HTML parameters” on page 70
You can use HTML parameters to configure and customize the viewer.

Related information:

 A guide to performance analysis when using ViewONE professional
See this white paper for information about performance tuning, including web-page refreshing and viewer reloading.

Adding selected items to the clipboard

You can use the following APIs to decide what is selected and added to the clipboard.

clearSelections method

Clears all page selections in the document.

This method is for use only with multiple page documents.

Syntax

```
ViewONE.clearSelections();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

copyPageToClipboard method

Puts a copy of the selected page in the Windows clipboard for pasting into other applications.

Syntax

```
ViewONE.copyPageToClipboard();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

getSelection method

Returns a comma-delimited string that contains the pages that the user selected.

Syntax

```
ViewONE.getSelection()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var selection = ViewONE.getSelection();
```

selectPage method

In a multiple page document, this method toggles the select property on the page in a document that is indicated by either the **pageNumber** parameter or `getPage()` method.

Syntax

This method is available only for multiple page documents.

```
ViewONE.selectPage(int pageNumber)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.selectPage(4);
```

Related reference:

“`getPage` method” on page 301

Returns the number of the current page as an integer.

startCopyAreaToClipboard method

Puts the mouse in selection mode, which allows the user to select an area of the current page to copy to the Windows clipboard. Replicates the behavior of the **Copy selected area to clipboard** option in the viewer **Clipboard** menu.

Syntax

```
ViewONE.startCopyAreaToClipboard();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, V4.0.34 or later
- IBM Daeja ViewONE Professional, V4.0.34 or later
- No additional modules

Related reference:

“**javascriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

Annotations

You can use the annotation methods determine whether annotations are displayed and to configure annotations settings.

Specify an Annotations File

You can use the following methods to set an annotation file and to determine whether or not annotations are displayed.

isShowAnnotations method:

Returns true if annotations are allowed to be displayed or false if annotations are not allowed to be displayed.

Syntax

```
ViewONE.isShowAnnotations()
```

Requirements

- IBM Daeja ViewONE Standard, V3.0.598 or later
- IBM Daeja ViewONE Professional, V1.0.598 or later
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
if (ViewONE.isShowAnnotations()) ...
```

setAnnotationFile method:

Sets the annotation file. You must call this method before you call any of the open methods.

Syntax

```
ViewONE.setAnnotationFile(String filename)
```

You must set the **annotationJavaScriptExtensions** HTML parameter to true to use the `setAnnotationFile` method.

If the current document is closed after you call this method, the annotation file is reset. Certain methods, such as the `initializePageArray` method, close the current document as a side effect, resetting also the annotation file. Therefore, close the current document before you call the `setAnnotationFile(filename)` method to avoid resetting the annotation file unintentionally.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

The following example sets annotation files.

```
ViewONE.setAnnotationFile("http://mysite/myannotations.ant");

viewONE.setAnnotationFile("http://...annotations.ant");
viewONE.openFile("http://...mydocument.tif", 1);

viewONE.closeDocument();
viewONE.setAnnotationFile("http://...annotations.ant");
viewONE.initializePageArray(2);
viewONE.setPageArray("http://...page1.tiff", 0);
viewONE.setPageArray("http://...page2.tif", 1);
viewONE.openPageArray(1);
```

setShowAnnotations method:

Sets whether IBM Daeja ViewONE displays annotations.

Syntax

```
ViewONE.setShowAnnotations(Boolean option)
```

Set to true to show annotations. Set to false to hide annotations.

If this method is set to false, IBM Daeja ViewONE hides only those annotations for which the user has edit privileges.

Requirements

- IBM Daeja ViewONE Standard, V3.0.598 or later
- IBM Daeja ViewONE Professional, V1.0.598 or later
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setShowAnnotations(false);
```

Open and save annotations

Use APIs to open, save, and reload annotations on a document.

reloadAnnotations method:

Forces IBM Daeja ViewONE to reload the annotations from the source and to redraw the image on screen.

Syntax

```
ViewONE.reloadAnnotations();
```

Use this method if you want to call the `setAnnotationFile(path)` method without having to close the document first.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationFile("http://www.mysite.com/myscript.pl?userID=12&docID=8791");
```

Related reference:

“setAnnotationFile method”

Specifies the location of the annotations definition file that is required by IBM Daeja ViewONE.

“**annotationJavascriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

saveAnnotations method:

Saves the annotations by using the server-side annotations save object that was specified by using the relevant HTML parameters or JavaScript methods.

Syntax

```
ViewONE.saveAnnotations();
```

Important: This method is asynchronous so the viewer does not wait for the save process to complete before proceeding. If your code requires the save process to be complete before it continues, monitor both of the following events:

- Event 24, which indicates that the save process was successful
- Event 25, which indicates that the save process failed

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.0.2 or later
- ViewONE Annotations Module

Related reference:

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

setAnnotationFile method:

Specifies the location of the annotations definition file that is required by IBM Daeja ViewONE.

Syntax

This file contains a list of the annotations that are to be displayed along with the parameters for these annotations.

```
ViewONE.setAnnotationFile(String path)
```

The **path** parameter can specify the file location relative to the viewer codebase. Alternatively, the **path** parameter can specify a server-side object such as CGI, EXE, or ASP that streams an annotations file to the viewer.

You can add parameters to the URL that is specified in the **path** parameter. For example, you might include a user ID to send user-specific annotations or a document ID to send document-specific annotations. If you specify a server-side object for this parameter, that object must handle any parameters that you include in the **path**.

The annotations file must conform to the IBM Daeja ViewONE annotations file specification.

If you are unsure whether IBM Daeja ViewONE is processing the file correctly, set the **TraceAnnotation** parameter to true to enable debug mode. In debug mode, more information is written in the log files.

Important: You must call this method before the document is opened. Alternatively, you can call the `reloadAnnotations()` method after you call the `setAnnotationFile(path)` method.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationFile("../docs/p1.ant");  
ViewONE.setAnnotationFile("http://www.mysite.com/myscript.pl?userID=12&docID=8791");
```

Related reference:

“`reloadAnnotations` method” on page 268

Forces IBM Daeja ViewONE to reload the annotations from the source and to redraw the image on screen.

Related information:

 [Creating an annotation template](#)

setAnnotationPostPrefix method:

Specifies the parameters that are to be added to the POST data for the `annotationSavePost` method.

Syntax

The POST object parses the data that is posted to extract all the parameters.

```
ViewONE.setAnnotationPostPrefix(String parameters)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

This example adds “`p1=value1`” and “`p2=value2`” as prefixes to the POST data.

```
ViewONE.setAnnotationPostPrefix("p1=value1&p2=value2");
```

setAnnotationSave method:

Specifies the location of a server-side object such as CGI, EXE, ASP, or JSP that saves annotations that were created by the viewer through multiple HTTP:GET calls.

Syntax

Whoever implements the system must supply an object that conforms to the specification for IBM Daeja ViewONE annotations save GET objects.

```
ViewONE.setAnnotationSave(String path)
```

If the **path** parameter specifies a relative address, the location is relative to the codebase of the viewer.

You can add parameters to the URL that is specified in the **path** parameter. For example, you might include a user ID to send user-specific annotations or a document ID to send document-specific annotations.

IBM Daeja ViewONE then adds its own parameters to the URL. Therefore, you must end the URL value correctly to ensure that it conforms to a standard URL format. For example, end the last parameter value with an ampersand (&). If the URL contains no parameters, end the value with the question mark (?).

If you are unsure what IBM Daeja ViewONE is sending to the servlet, set the **TraceAnnotation** parameter to true to enable debug mode. In debug mode, more information is written in the log files.

The server-side object is called only if the document that is displayed was retrieved through a web server. For example, if a document is retrieved through the `file:///` protocol, the servlet is not called when the user presses the annotations save button.

Important: You must call this method before the document is opened.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationSave("http://www.mysite.com/myscript.pl?  
userID=12&docID=8791&");
```

setAnnotationSavePost method:

Specifies the location of a server-side object such as CGI, EXE, ASP, or JSP that saves annotations that were created by the viewer.

Syntax

Whoever implements the system must supply an object that conforms to the specification for IBM Daeja ViewONE annotations save POST objects.

```
ViewONE.setAnnotationSavePost(String path)
```

If the **path** parameter specifies a relative address, the location is relative to the codebase of the viewer.

If you are unsure what IBM Daeja ViewONE is sending to the servlet, set the **TraceAnnotation** parameter to true to enable debug mode. In debug mode, more information is written in the log files.

The server-side object is called only if the document that is displayed was retrieved through a web server. For example, if a document is retrieved through the `file:///` protocol, the servlet is not called when the user presses the annotations save button.

Important: You must call this method before the document is opened.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationSavePost("http://www.mysite.com/annotationsave.dll");
```

Related information:

 [Configuring the object with HTTP:POST](#)

setAnnotationSaveServlet method:

Specifies the location of a servlet that handles the saving of annotations that are created by the viewer.

Syntax

Whoever implements the system must supply a servlet that conforms to the specification for IBM Daeja ViewONE annotations save servlets.

```
ViewONE.setAnnotationSaveServlet(String path)
```

If the **path** parameter specifies a relative address, the location is relative to the codebase of the viewer.

You can add parameters to the URL that is specified in the **path** parameter. For example, you might include a user ID to send user-specific annotations or a document ID to send document-specific annotations.

If you are unsure what IBM Daeja ViewONE is sending to the servlet, set the **TraceAnnotation** parameter to true to enable debug mode. In debug mode, more information is written in the log files.

The servlet is called only if the document that is displayed was retrieved through a web server. For example, if a document is retrieved through the `file:///` protocol, the servlet is not called when the user presses the annotations save button.

Important: You must call this method before the document is opened.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- ViewONE Annotations Module

Example


```
ViewONE.setAnnotationSaveServlet  
("http://www.mysite.com/servlet/annotationsave.class?userID=12&docID=8791");
```

Related reference:

“**annotationFile** parameter” on page 79

This parameter specifies the annotations definition file that contains the list of the annotations to display and their associated parameters.

Related information:

 Configuring the object as a servlet

Edit and find annotations

Use APIs to edit and find annotations.

addAnnotation method:

Adds an annotation with the specified properties.

The **annotationProperties** parameter must specify all the properties that are required by the annotation type.

Syntax

```
ViewONE.addAnnotation(String annotationProperties)
```

Use a delimited string for the **annotationProperties** parameter value. You can call the `getDelimiter()` method to retrieve the delimiter to be used.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.addAnnotation("[HIGHLIGHT]<P>PAGE=0<P>X=300<P>Y=350<P>WIDTH=400  
<P>HEIGHT=500<P>LABEL=myHighlight1<P>HYPERLINK=<page><3><P>");
```

Related reference:

“`getDelimiter` method” on page 281

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

Related information:

Creating an annotation template

addAnnotationStamp method:

Adds the specified image stamp or text stamp to the annotation stamp menu. If this method was called previously, the method replaces the previous menu item instead of creating a new menu item.

Syntax

```
viewONE.addAnnotationStamp(String TEXTorURL, String displayText)
```

To specify an image stamp, you must specify the **TEXTorURL** parameter as `image:` and followed by the URL of the image file. The URL can point to any image file that can be decoded by IBM Daeja ViewONE.

You can use the `#` modifier to specify a page that the method is to extract from the image file. For example, `mystamp.tif#2` extracts page 2 from the `mystamp.tif` file for use as the signature. If you do not specify the `#` modifier for a multiple page image, the method uses the first page in the file.

To specify a text stamp, set the **TEXTorURL** parameter to the text that is to be displayed in the text stamp annotation.

Use the **displayText** parameter to specify the string value that is to be displayed in the annotation stamp menu for the stamp.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.addAnnotationStamp("image:http://mysite/mystamp.tif#2", "Sign John Smith");  
ViewONE.addAnnotationStamp("Updated", "Updated");
```

addAnnotationStampWithProperties method:

Adds the specified image stamp or text stamp with the specified properties applied to the annotation stamp menu. If this method was called previously, the method replaces the previous menu item instead of creating a new menu item.

Syntax

```
viewONE.addAnnotationStampWithProperties(String TEXTorURL,  
    String displayText, String properties)
```

To specify an image stamp, you must specify the **TEXTorURL** parameter as `image:` and followed by the URL of the image file. The URL can point to any image file that can be decoded by IBM Daeja ViewONE.

You can use the `#` modifier to specify a page that the method is to extract from the image file. For example, `mystamp.tif#2` extracts page 2 from the `mystamp.tif` file for use as the signature. If you do not specify the `#` modifier for a multiple page image, the method uses the first page in the file.

To specify a text stamp, set the **TEXTorURL** parameter to the text that is to be displayed in the text stamp annotation.

Use the **displayText** parameter to specify the string value that is to be displayed in the annotation stamp menu for the stamp.

The format of the properties string is as follows:

```
<property1=value1><property2=value2>...
```

Example strings:

```
<color=blue>, or <color=blue><rotation=45>
```

The following table contains the full list of properties.

Property name	Description and value options
Color	Color of text or line (or transparency color for image stamps)
FillColor	Background Color of text or fill color of shape
Border	<ul style="list-style-type: none"> • 1 = The annotation has a border • 0 = The annotation does not have a border
Menu	Menu text to display (for stamp menus)
Rotation	Rotation of annotation (0, 45, 90, 180 or 270)
Label	Text for the annotation label
HelpText	Tooltip text
Owner	Owner (user name) text
FontHeight	Height in pixels for text or stamp annotations
Print	<ul style="list-style-type: none"> • True = users can print annotation • False = users cannot print annotation
Read	<ul style="list-style-type: none"> • True = users can read annotation • False = users cannot read annotation
Modify	<ul style="list-style-type: none"> • True = users can modify annotation • False = users cannot modify annotation
Delete	<ul style="list-style-type: none"> • True = users can delete annotation • False = users cannot delete annotation
Scale	Scale factor for image stamp annotations (for example, 1.5)
ModifySecurity	<ul style="list-style-type: none"> • True = users can modify annotation security • False = users cannot modify annotation security

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
viewONE.addAnnotationStampWithProperties("image:http://mysite/mystamp.tif#2",  
    "Sign John Smith", "<color=blue>");  
viewONE.addAnnotationStampWithProperties("Updated", "Updated",  
    "<color=blue><rotation=45>");
```

burnAnnotations method:

Starts the burn process with the server-side permanent redaction object that is defined. This method performs the same action as clicking **Burn**.

Syntax

```
viewONE.burnAnnotations(boolean prompt)
```

The **prompt** parameter defines whether the user is prompted to accept or reject the burn before it is started. Set this parameter to true to prompt the user.

Requirements

- IBM Daeja ViewONE Professional, V1.1.96 or later
- ViewONE Permanent Redaction Server Module

Example

```
viewONE.burnAnnotations(false)
```

clearAnnotationStamps method:

Clears the contents of the annotation stamp menu.

Syntax

```
viewONE.clearAnnotationStamps();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

deleteAllAnnotations method:

Deletes the annotation of the specified type from the specified page.

Syntax

```
viewONE.deleteAllAnnotations(String type, Int page)
```

Specify one of the following values for the **type** parameter:

- any
- arrow
- freehand
- highlight
- highlightPoly
- line
- note
- openPoly

- oval
- poly
- rectangle
- redact
- redactPoly
- stamp
- text

If you set the **type** parameter to any, the list includes all types of annotation.

Notes:

- A circle annotation is an oval polygon with fixed aspect ratio.
- A square is a rectangle annotation with fixed aspect ratio.
- A ruler is a line annotation with different display properties.
- An angle is an openPoly annotation with different display properties.

Set the **page** parameter to -1 to include annotation labels from all pages in the document.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
viewONE.deleteAllAnnotations("highlight", 1);
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

deleteAnnotation method:

Deletes the annotation that has the specified label.

Syntax

```
viewONE.deleteAnnotation(String label)
```

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.deleteAnnotation("myHighlight1");
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

endAnnotation method:

Ends the editing of the current annotation and stores any changes internally.

You can also use this method to end the automatic reselection of the current annotation type when the editing of the current new annotation is finished.

Syntax

```
ViewONE.endAnnotation();
```

Requirements

- IBM Daeja ViewONE Standard, V3.1.118 or later
- IBM Daeja ViewONE Professional, V1.1.118 or later
- ViewONE Annotations Module

endModifyAnnotations method:

Ends the modification of multiple annotations that was started by a call to the `startModifyAnnotations` method.

Syntax

```
ViewONE.endModifyAnnotations();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Related reference:

“`startModifyAnnotations` method” on page 297

Starts the modification of multiple annotations.

getActiveAnnotation method:

Returns the label of the active annotation.

Syntax

```
ViewONE.getActiveAnnotation()
```

The active annotation is the currently selected annotation. If no annotation is selected, this method returns `NONE`.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var annotationLabel = ViewONE.getActiveAnnotation();
```

getAnnotation method:

Returns a delimited list that contains the properties of the annotation with the specified label on the current page.

Syntax

```
ViewONE.getAnnotation(String label)
```

To retrieve the delimiter that is used in the list, call the `getDelimiter` method.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var strAnnotationProperties = ViewONE.getAnnotation("myHighlight1");
```

Related reference:

“`getDelimiter` method” on page 281

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

getAnnotationLabels method:

Returns a delimited list that contains the labels of the annotations of the specified type that are on the specified page.

Syntax

```
ViewONE.getAnnotationLabels(String type, Int page, Int order)
```

Specify one of the following values for the **type** parameter:

- any
- arrow
- freehand
- highlight
- highlightPoly
- line
- note
- openPoly
- oval
- poly
- rectangle

- redact
- redactPoly
- stamp
- text

If you set the **type** parameter to any, the list includes all types of annotation.

Notes:

- A circle annotation is an oval polygon with fixed aspect ratio.
- A square is a rectangle annotation with fixed aspect ratio.
- A ruler is a line annotation with different display properties.
- An angle is an openPoly annotation with different display properties.

Set the **page** parameter to -1 to include annotation labels from all pages in the document.

Use the **order** parameter to specify the order of the returned list. Set this parameter to 0 to list the annotation labels in the order in which they were added to the document. Set this parameter to 1 to list the annotation labels in the order they are shown on the page according to their position on the X/Y axis.

To retrieve the delimiter that is used in the list, call the `getDelimiter` method.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var strAnnotationLabels = ViewONE.getAnnotationLabels("highlight", 1, 0);
```

Related reference:

“getDelimiter method” on page 281

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

“annotationJavaScriptExtensions parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

getAnnotationOnPage method:

Returns a delimited list that contains the properties of the annotation with the specified label that is on the specified page.

Syntax

```
ViewONE.getAnnotationOnPage(String label, int page)
```

To retrieve the delimiter that is used in the list, call the `getDelimiter` method.

If there is no annotation with the specified label or the JavaScript is not ready, this method returns NONE. If two or more annotations on the page have the same label, the results of this method are undefined.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var strAnnotationProperties = ViewONE.getAnnotationOnPage("myHighlight1",1);
```

Related reference:

“getDelimiter method”

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

getDelimiter method:

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

Syntax

```
ViewONE.getDelimiter()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var strDelimiter = ViewONE.getDelimiter();
```

getNumAnnotations method:

Returns the number of annotations of the specified type that are on the specified page.

Syntax

```
ViewONE.getNumAnnotations(String type, int page)
```

Specify one of the following values for the **type** parameter:

- any
- arrow
- freehand
- highlight
- highlightPoly

- line
- note
- openPoly
- oval
- poly
- rectangle
- redact
- redactPoly
- stamp
- text

If you set the **type** parameter to any, the list includes all types of annotation.

Notes:

- A circle annotation is an oval polygon with fixed aspect ratio.
- A square is a rectangle annotation with fixed aspect ratio.
- A ruler is a line annotation with different display properties.
- An angle is an openPoly annotation with different display properties.

Set the **page** parameter to -1 to include annotation labels from all pages in the document.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var intNumAnnotations = ViewONE.getNumAnnotations("highlight", 1);
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

insertAnnotationStamp method:

Adds the specified text stamp to the annotation stamp menu.

Syntax

```
ViewONE.insertAnnotationStamp(String text, int beforeIndex, String properties)
```

The new text stamp is inserted before the stamp at the index indicated by the **beforeIndex** parameter. The index starts at 0 for the first item in the menu. If the **beforeIndex** parameter is greater than the number of stamps, this method adds the new stamp to the end.

Use the **properties** parameter to specify any properties for the new menu item. For the format to use for this parameter, see the

startAnnotationWithProperties(type, properties) method. Setting this parameter to null sets the parameters for the menu item to the default values.

This method returns true if the stamp is inserted successfully. Otherwise, this method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

In the following example, 0 is the first item in the menu, and null sets the parameters for the menu item to the default values.

```
var ok = ViewONE.insertAnnotationStamp("my stamp", 0, null);
```

Related reference:

“startAnnotationWithProperties method” on page 295

Gets IBM Daeja ViewONE to behave as if one of the annotations toolbar buttons is selected.

isAnnotationsSemiTransparent method:

Returns a Boolean value that indicates whether the specified annotation types are set to semitransparent.

Syntax

```
ViewONE.isAnnotationsSemiTransparent(String type)
```

Use one of the following values for the **type** parameter.

Value	Description
all	Returns true if all annotation types are set to semitransparent.
redacttypes	Returns true if all annotations of the redaction type are set to semitransparent.
burnable	Returns true if all annotations whose type can be permanently redacted are semitransparent. The burnable property is set by using the annotationDefaults HTML parameter.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var isRedactSemi = ViewONE.isAnnotationsSemiTransparent("redacttypes");
```

Related reference:

“**annotationDefaults** parameter” on page 259

This parameter specifies the annotation types that are burnable.

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

isAnnotationToolbar method:

Returns a Boolean value of true if the main annotations toolbar is displayed.

Syntax

Otherwise, this method returns false.

```
var boolAnnotationToolbar = ViewONE.isAnnotationToolbar();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

isAnnotationsUpdated method:

Returns a Boolean value of false if annotations did not change since they were last saved in IBM Daeja ViewONE.

Syntax

This method returns true if the annotations changed since they were last saved.

```
var boolAnnotateUpdated = ViewONE.isAnnotationsUpdated();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

modifyAnnotation method:

Modifies the properties of the annotation with the specified label.

Syntax

The **annotationProperties** parameter needs to specify only the properties that are to be changed or added.

```
ViewONE.modifyAnnotation(String label, String annotationProperties)
```

You can call the `getDelimiter()` method to retrieve the delimiter that is used in the **annotationProperties** parameter.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.modifyAnnotation("myHighlight1", "X=500<P>Y=350<P>WIDTH=400  
<P>HEIGHT=600<P>");
```

Related reference:

“getDelimiter method” on page 281

Returns the string that IBM Daeja ViewONE is using as a delimiter in returned Strings that contain multiple values.

“**annotationJavascriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

Related information:

 [Creating an annotation template](#)

parseProperty method:

Returns the value of the specified property from the specified annotation property list.

Syntax

```
ViewONE.parseProperty(String property, String annotationProperties)
```

Use this method with the `getAnnotation(label)` method to avoid parsing the annotation properties in JavaScript.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Examples

```
var strWidth = ViewONE.parseProperty("width", "[HIGHLIGHT]<P>PAGE=0<P>X=300  
<P>Y=350<P>WIDTH=400<P>HEIGHT=500<P>LABEL=myHighlight1  
<P>HYPERLINK=<page><3><P>");
```

```
var strWidth = ViewONE.parseProperty("width", getAnnotation("highlight1"));
```

Related reference:

“getAnnotation method” on page 279

Returns a delimited list that contains the properties of the annotation with the specified label on the current page.

removeAnnotationStamp method:

Removes the annotation stamp from the specified index in the annotation stamps menu.

Syntax

```
ViewONE.removeAnnotationStamp(int index)
```

The items in the menu are indexed starting at 0.

This method returns true if the stamp is removed successfully. Otherwise, this method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var ok = ViewONE.removeAnnotationStamp(0);
```

setAnnotateEdit method:

Determines whether the user can modify editable annotations and add new annotations by using the annotations toolbar.

Syntax

```
ViewONE.setAnnotateEdit(Boolean option)
```

When the method is called with false, annotations cannot be edited (or added - only viewed), and the annotations toolbar is removed from the interface.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotateEdit(true);
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

setAnnotationHideButtons method:

Hides the specified buttons on the annotations toolbar.

Syntax

Hiding a button disables only the ability to create an annotation through its user interface.

```
ViewONE.setAnnotationHideButtons(String)
```

The **string** parameter is a comma-delimited string. Use the following values to identify the buttons that are to be hidden:

Value	Button name
<i>restore</i>	Restore annotations
<i>save</i>	Save annotation changes
<i>select</i>	Select mode
<i>line</i>	Draw line

Value	Button name
<i>arrow</i>	Draw arrow
<i>text</i>	Draw text
<i>solidText</i>	Draw solid text
<i>note</i>	Draw sticky note
<i>highlight</i>	Draw highlight
<i>Transparent highlight</i>	Draw hyperlink
<i>highlightPoly</i>	Draw polygon highlight
<i>rectangle</i>	Draw rectangle
<i>Square</i>	Draw square
<i>redact</i>	Draw redaction
<i>redactPoly</i>	Draw polygon redaction
<i>poly</i>	Draw polygon
<i>openPoly</i>	Draw open polygon
<i>oval</i>	Draw oval
<i>circle</i>	Draw circle
<i>freehand</i>	Draw freehand line
<i>stamp</i>	Draw stamp
<i>ruler</i>	Draw ruler
<i>angle</i>	Draw angle
<i>show</i>	Show annotations

To clear the list of hidden buttons, call the `setAnnotationHideButtons` method with an empty string.

Security note: This method is disabled by default unless the `annotationJavaScriptExtensions` parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationHideButtons("note,text,select,arrow");
ViewONE.setAnnotationHideButtons(""); // to clear the list of hidden buttons
```

Related reference:

“`annotationJavaScriptExtensions` parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

`setAnnotationHideContextButtons` method:



















Specifies the buttons that are to be hidden on the context-sensitive toolbar.



Syntax

Hiding a button disables the ability to modify an annotation's default properties through the user interface. The default is to display all buttons.

```
ViewONE.setAnnotationHideContextButtons(String buttons)
```

Use the following values in a comma-delimited string to identify the buttons that are to be hidden. These values are not case-sensitive. If you disable all the buttons, the context-sensitive toolbar is hidden.

Button	Value	Description
	save	Fix button
	security	Edit security button
	text	Text editor button
	changeFont	Annotation font editor button
	increaseline	Increase line width
	decreaseline	Decrease line width
	increasefont	Increase text font size
	decreasefont	Decrease text font size
	increasearrowhead	Increase arrowhead size
	decreasearrowhead	Decrease arrowhead size
	linecolor	Line color chooser
	fillcolor	Fill color chooser
	strikethrough	Strike through text (IBM Daeja ViewONE Standard, V3.0.332 or later, IBM Daeja ViewONE Professional, V1.0.332 or later)
	transparent	Make text semi-transparent
	hyperlink	Hyperlink dialog
	rotater	Rotate clockwise
	rotatel	Rotate counterclockwise
	angleflip	Flip angle (changes between obtuse & acute) (V3-only)

Button	Value	Description
	behind	Move behind button
	delete	Delete button (The delete key is still enabled.)

Requirements

- IBM Daeja ViewONE Standard, V4.0.6 or later
- IBM Daeja ViewONE Professional, V4.0.6 or later
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationHideContextButtons("freehand, hyperlink, text");
```

Related reference:

“setAnnotationHideContextButtonsIds method”

This method specifies the annotation types for which the buttons that are specified by the setAnnotationHideContextButtons method or the **annotationHideContextButtons** parameter are to be hidden.

setAnnotationHideContextButtonsIds method:

This method specifies the annotation types for which the buttons that are specified by the setAnnotationHideContextButtons method or the **annotationHideContextButtons** parameter are to be hidden.

Syntax

```
ViewONE.setAnnotationHideContextButtonsIds(String AnnotationTypes)
```

For example, to disable the context-sensitive hyperlink dialog button for only the note annotation type, you make the following calls:

```
ViewONE.setAnnotationHideContextButtons("hyperlink");
ViewONE.setAnnotationHideContextButtonsIds("note");
```

Use the following values in a comma-delimited string to specify the annotation types:

- arrow
- freehand
- highlight
- highlightpoly
- line
- note
- openpoly
- oval
- poly
- rectangle
- redact
- redactpoly
- stamp
- text

The annotation types are not case-sensitive.

The default is to apply the **annotationHideContextButtons** values to all annotation types.

Requirements

- IBM Daeja ViewONE Standard, V4.0.6 or later
- IBM Daeja ViewONE Professional, V4.0.6 or later
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationHideContextButtonsIds("note, text");
```

Related reference:

“setAnnotationHideContextButtons method” on page 287

Specifies the buttons that are to be hidden on the context-sensitive toolbar.

“**annotationHideContextButtons** parameter” on page 134

This parameter determines which buttons are hidden on the annotation context-sensitive toolbar.

setAnnotationsSemiTransparent method:

Sets whether the annotations are semitransparent.

Syntax

```
ViewONE.setAnnotationsSemiTransparent(Boolean option, String type)
```

Call this method with a value of true to make annotations semitransparent so that the user can see the image behind the annotations and position the annotations.

Use one of the following values for the **type** parameter.

Value	Description
all	Sets all editable annotations to semitransparent.
redacttypes	Sets all redact and redactpoly annotation types to semitransparent. Tip: The setAnnotationsSemiTransparent(Boolean, type) method with this parameter replaces the deprecated setRedactionIsSemiTransparent(Boolean) method.
burnable	Sets all annotations whose type can be permanently redacted to semitransparent. The burnable property is set by using the annotationDefaults HTML parameter. See the <i>IBM Daeja ViewONE Parameters Reference Manual</i> for details of the burnable property.

Call this method with a value of false to disable the semitransparent mode.

To change the color that is used for the semitransparent redaction annotations, use the HTML parameter **transparentRedactionColor**.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setAnnotationsSemiTransparent(true, "all");
```

Related reference:

“setRedactionIsSemiTransparent method” on page 292

Sets the redaction mode in IBM Daeja ViewONE to semitransparent.

“transparentRedactionColor parameter” on page 103

This parameter specifies the fill color that is used for redaction annotations in semi-transparent mode.

“annotationJavascriptExtensions parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

setAnnotationStampText method:

Sets the text for an annotation stamp menu item to the string specified by the **text** parameter.

Syntax

```
ViewONE.setAnnotationStampText(String text, int index)
```

Use the **index** parameter to identify the menu item with 0 indicating the first item in the menu.

This method returns true if the stamp is changed. Otherwise, this method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
var ok = ViewONE.setAnnotationStampText("my stamp", 0);
```

setDefaultSelectAnnotation method:

Determines whether the annotation selection button is always in the on (pressed) state.

Syntax

```
ViewONE.setDefaultSelectAnnotation(Boolean option)
```

Call this method with a value of true to keep the button in the on state.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- ViewONE Annotations Module

Example

```
ViewONE.setDefaultSelectAnnotation(true);
```

setDelimiter method:

Sets the string that IBM Daeja ViewONE is to use as a delimiter in returned Strings that contain multiple values.

Syntax

```
ViewONE.setDelimiter(String delimiter)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setDelimiter("||");
```

setRedactionIsSemiTransparent method:

Sets the redaction mode in IBM Daeja ViewONE to semitransparent.

Syntax

```
ViewONE.setRedactionIsSemiTransparent(Boolean option)
```

Call this method with a value of true to make redaction annotations semitransparent so that the user can see the redacted text and the position of the annotation.

Call this method with a value of false to return to normal redaction mode.

To change the color that is used for the semitransparent redaction annotations, use the HTML parameter **transparentRedactionColor**.

Deprecated: The `setRedactionIsSemiTransparent(Boolean)` method is deprecated as of Version 4.0.10. Use the `setAnnotationsSemiTransparent(Boolean, type)` method instead.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setRedactionIsSemiTransparent(true);
```

Related reference:

“`setAnnotationsSemiTransparent` method” on page 290

Sets whether the annotations are semitransparent.

“**transparentRedactionColor** parameter” on page 103

This parameter specifies the fill color that is used for redaction annotations in

semi-transparent mode.

setRulerScale method:

Sets the scale for all the ruler annotations that are later added.

Syntax

```
ViewONE.setRulerScale(int double)
```

Requirements

- IBM Daeja ViewONE Standard, V3.0.794 or later
- IBM Daeja ViewONE Professional, V1.0.794 or later
- ViewONE Annotations Module

Example

```
ViewONE.setRulerScale(1.0);
```

setRulerUnits method:

Sets the units that the ruler annotation displays.

Syntax

```
ViewONE.setRulerUnits(String text)
```

Use the following values for the **text** parameter:

- *cm centimeters*
- *mm millimeters*
- *inches inches*
- *inchesandcm inches and centimeters*

Requirements

- IBM Daeja ViewONE Standard, V3.0.794 or later
- IBM Daeja ViewONE Professional, V1.0.794 or later
- ViewONE Annotations Module

Example

```
ViewONE.setRulerScale("cm");
```

setStickyAnnotations method:

Determines whether the user can reuse a selected annotation without selecting that annotation again from the annotation toolbar.

Syntax

```
ViewONE.setStickyAnnotations(Boolean option)
```

Call this method with a value of `true` to enable the user to reuse a selected annotation. Call this method with a value of `false` to force the user to reselect the annotation before the annotation is reused.

You must call this method before the user selects an annotation. You need to call this method only once in a session.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.setStickyAnnotations(true);
```

showAnnotationToolBar method:

Specifies whether to show or hide the main annotations toolbar.

Syntax

```
ViewONE.showAnnotationToolBar(Boolean option)
```

Call this method with a value of true to show the toolbar.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.showAnnotationToolBar(false);
```

startAnnotation method:

Gets IBM Daeja ViewONE to behave as if one of the annotations toolbar buttons is selected.

Syntax

```
ViewONE.startAnnotation(String type)
```

Typically, this method puts the mouse cursor into the mode where the cursor is ready to place an annotation. The type of annotation is determined by the **type** parameter. You can use the following values for this parameter:

- line
- arrow
- text
- textSolid
- note
- highlight
- highlightPoly
- hyperlink
- rectangle
- square
- redact
- redactPoly
- poly
- openPoly

- oval
- circle
- freehand
- ruler
- angle
- anglereversed
- stamp
- stampMenu<N>

If you specify stampMenu<N> for the **type** parameter, *N* represents the position of the stamp item in the stamp menu list. The first item in the menu is "1", the next is "2," and so on.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.startAnnotation("text");
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

startAnnotationWithProperties method:

Gets IBM Daeja ViewONE to behave as if one of the annotations toolbar buttons is selected.

Syntax

```
ViewONE.startAnnotationWithProperties(String type, String properties)
```

Typically, this method puts the mouse cursor into the mode where the cursor is ready to place an annotation. The type of annotation is determined by the **type** parameter. You can use the following values for this parameter:

- line
- arrow
- text
- textSolid
- note
- highlight
- highlightPoly
- hyperlink
- rectangle
- square
- redact

- redactPoly
- poly
- openPoly
- oval
- circle
- freehand
- ruler
- angle
- angleReversed
- stamp
- stampMenu<N>

If you specify stampMenu<N> for the **type** parameter, N represents the position of the stamp item in the stamp menu list. The first item in the menu is "1", the next is "2," and so on.

You can use the **properties** parameter to override the default settings for specific annotations. The format of the **properties** parameter is "<property1=value1><property2=value2>", for example, "<color=blue>" or "<color=blue><rotation=45>".

The following table lists the optional properties that you can specify:

Property name	Description and value options
Color	Color of text or line (or transparency color for image stamps)
FillColor	Background Color of text or fill color of shape
Menu	Menu text to display (for stamp menus)
Rotation	Rotation of annotation (0, 45, 90, 135, 180, 225, 270, or 315)
Label	Text for the annotation label
HelpText	Tooltip text
Owner	Owner (user name) text
FontHeight	Height in pixels for text/stamp annotations
Print	True = other users can print annotation False = other users cannot print annotation
Read	True = other users can view annotation False = other users cannot view annotation
Modify	True = other users can modify annotation False = other users cannot modify annotation
Delete	True = other users can delete annotation False = other users cannot delete annotation

Property name	Description and value options
ModifySecurity	True = other users can modify annotation security False = other users cannot modify annotation security

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.startAnnotationWithProperties("text", "<color=blue>");
```

Related reference:

“**annotationJavaScriptExtensions** parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

startModifyAnnotations method:

Starts the modification of multiple annotations.

Syntax

Use this method with the `endModifyAnnotations()` method to encapsulate multiple calls to the `modifyAnnotation(label, annotationProperties)` method.

```
ViewONE.startModifyAnnotations();
```

Typically, the viewer repaints after you call the `modifyAnnotation(label, annotationProperties)` method. Making multiple calls to this method at a time, can cause flicker and slow performance. To prevent this problem, encapsulate the calls to the `modifyAnnotation(label, annotationProperties)` method in the `startModifyAnnotations()` method and `endModifyAnnotations()` method.

Security note: This method is disabled by default unless the **annotationJavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Annotations Module

Example

```
ViewONE.startModifyAnnotations();
ViewONE.modifyAnnotation("line1", <annotation properties>);
ViewONE.modifyAnnotation("line2", <annotation properties>);
ViewONE.endModifyAnnotations();
```

Related reference:

“endModifyAnnotations method” on page 278

Ends the modification of multiple annotations that was started by a call to the startModifyAnnotations method.

“modifyAnnotation method” on page 284

Modifies the properties of the annotation with the specified label.

“annotationJavascriptExtensions parameter” on page 111

This parameter determines whether the viewer accepts calls for certain JavaScript methods to provide an extra level of security.

Document hyperlinks

Use the clearHyperlink() method and setHyperlink(url, dblClick) method to manipulate document hyperlinks.

clearHyperlink method

Clears a hyperlink if one was previously defined by using the setHyperlink(url, dblClick) method.

Syntax

```
ViewONE.clearHyperlink();
```

Requirements

- IBM Daeja ViewONE, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“setHyperlink method”

Specifies a hyperlink that is activated when the user clicks the image area.

setHyperlink method

Specifies a hyperlink that is activated when the user clicks the image area.

Syntax

```
ViewONE.setHyperlink(String url, Boolean dblClick)
```

If the **dblClick** parameter is true, then the hyperlink is activated only after the user double-clicks the image area, otherwise it requires only a single click.

The hyperlink can specify either a file name relative to the code base or a full URL.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setHyperlink("newpage.html", false);  
ViewONE.setHyperlink("http://mysite/newpage.html", false);
```

Documents and images

You can use this API to perform basic operations with documents and or retrieve basic information.

closeDocument method

Closes an open document. When this method closes an open document, the method also resets annotations, annotation templates, and background images.

Syntax

```
ViewONE.closeDocument();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

firstDoc method

Opens the first document in the series of multiple documents that are loaded into the viewer.

Syntax

```
ViewONE.firstDoc()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray`, `setDocumentArray`, and `openDocumentArray` method group JavaScript equivalent. If this method is called when on the first document in the series, the method has no effect.

Calling this method is the JavaScript equivalent of clicking the **First Document** button on the toolbar when multiple documents are loaded in the viewer.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“`initializeDocumentArray`, `openDocumentArray`, and `setDocumentArray` method group” on page 302

Specifies the number of documents to open in a multiple document session. The `setDocumentArray` method then specifies the parameter file for each document where each file represents a successive document. The `openDocumentArray` method then opens the assembled document list at the provided document index.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

getDocIndex method

Returns the index value of the currently viewed document in the series of multiple documents that are loaded into the viewer.

Syntax

```
ViewONE.getDocIndex()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray`, `setDocumentArray`, and `openDocumentArray` method group JavaScript equivalent.

The index starts at 1, which represents the first document in the list.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
var currentdoc = ViewONE.getDocIndex()
```

Related reference:

“`initializeDocumentArray`, `openDocumentArray`, and `setDocumentArray` method group” on page 302

Specifies the number of documents to open in a multiple document session. The `setDocumentArray` method then specifies the parameter file for each document where each file represents a successive document. The `openDocumentArray` method then opens the assembled document list at the provided document index.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

getDocReference method

Gets the document reference for the document.

Syntax

```
ViewONE.getDocReference()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var ref = ViewONE.getDocReference();
```

Related reference:

“**docRef<N>** parameter” on page 178

This parameter determines the reference number that is to be used for the specified document. You might use this document reference number in error retrieval files and print separator pages.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

getNumDocs method

Returns an integer that represents the total number of documents that are loaded into the viewer.

Syntax

```
ViewONE.getNumDocs()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray`, `setDocumentArray`, and `openDocumentArray` method group JavaScript equivalent.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var numdocs = ViewONE.getNumDocs()
```

Related reference:

“`initializeDocumentArray`, `openDocumentArray`, and `setDocumentArray` method group” on page 302

Specifies the number of documents to open in a multiple document session. The `setDocumentArray` method then specifies the parameter file for each document where each file represents a successive document. The `openDocumentArray` method then opens the assembled document list at the provided document index.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

getNumPages method

Returns the number of pages in the current document as an integer.

Syntax

```
ViewONE.getNumPages()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
var numPages = ViewONE.getNumPages();
```

getPage method

Returns the number of the current page as an integer.

Syntax

```
ViewONE.getPage()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
var page = ViewONE.getPage();
```

initializeDocumentArray, openDocumentArray, and setDocumentArray method group

Specifies the number of documents to open in a multiple document session. The setDocumentArray method then specifies the parameter file for each document where each file represents a successive document. The openDocumentArray method then opens the assembled document list at the provided document index.

Syntax

```
ViewONE.initializeDocumentArray(int numDocs)
ViewONE.openDocumentArray()
ViewONE.setDocumentArray(String filename, int index)
```

The document array begins at array element zero.

Initializing the document array causes a soft close to be performed. That is, the current document is closed and the annotation source, templates, and background images are reset if a document is open. If no document is open, then the close and reset is not performed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Examples

```
ViewONE.initializeDocumentArray(3);
ViewONE.setDocumentArray("doc1parameters.txt", 0);
ViewONE.setDocumentArray ("doc2parameters.txt", 1);
ViewONE.setDocumentArray ("doc3parameters.txt", 2);
ViewONE.openDocumentArray(1);
```

Related reference:

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

initializePageAndThumbsArray, openPageArray, setPageArray, and setThumbsArray method group

Specifies the number of files (pages) in a list, then specifies each file where each file represents a successive page of the document. Specifies each file where each file represents a thumbnail for that page in the document, and then opens the assembled document at the provided page.

Syntax

```
ViewONE.initializePageAndThumbsArray(int numPages)
ViewONE.openPageArray(int page)
ViewONE.setPageArray(String filename, int page)
ViewONE.setThumbsArray(String filename, int page)
```

The final method opens the assembled document at the provided page. The page array begins at array element zero.

Because smaller files are quicker to download and view, this method can make browsing thumbnails quicker by specifying different files for the thumbnails.

Initializing the page and thumb arrays causes a soft close to be performed. That is, the current document is closed and the annotation source, templates, and background images are reset if a document is open. If no document is open, then the close and reset is not performed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.initializePageAndThumbsArray(3);
ViewONE.setPageArray("page1.tif", 0);
ViewONE.setThumbsArray("page1-t.tif", 0);
ViewONE.setPageArray("page2.tif", 1);
ViewONE.setThumbsArray("page2-t.tif", 1);
ViewONE.setPageArray("page3.tif", 2);
ViewONE.setThumbsArray("page3-t.tif", 2);
ViewONE.openPageArray(1);
```

initializePageArray, openPageArray, and setPageArray method group

Specifies the number of files (pages) in a list, then specifies each file where each file represents a successive page of the document, and then opens the assembled document at the provided page.

Syntax

```
ViewONE.initializePageArray(int numPages)
ViewONE.openPageArray(int page)
ViewONE.setPageArray(String filename, int page)
```

The page array begins at array element zero.

Initializing the page array causes a soft close to be performed. That is, the current document is closed and the annotation source, templates, and background images are reset if a document is open. If no document is open, then the close and reset is not performed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.initializePageArray(3);
ViewONE.setPageArray("page1.tif", 0);
ViewONE.setPageArray("page2.tif", 1);
ViewONE.setPageArray("page3.tif", 2);
ViewONE.openPageArray(1);
```

LastDoc method

Opens the last document in the series of multiple documents that are loaded into the viewer.

Syntax

```
ViewONE.lastDoc()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray(numDocs)`, `setDocumentArray(filename, index)`, and `openDocumentArray()` method group JavaScript equivalent. If this method is called when on the last document in the series, the method has no effect.

Calling this method is the JavaScript equivalent of clicking the **Last Document** button on the toolbar when multiple documents are loaded in the viewer.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“initializeDocumentArray, openDocumentArray, and setDocumentArray method group” on page 302

Specifies the number of documents to open in a multiple document session. The `setDocumentArray` method then specifies the parameter file for each document where each file represents a successive document. The `openDocumentArray` method then opens the assembled document list at the provided document index.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

nextDoc method

Opens the next document in the series of multiple documents that are loaded into the viewer.

Syntax

```
ViewONE.nextDoc()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray(numDocs)`, `setDocumentArray(filename, index)`, and `openDocumentArray()` method group JavaScript equivalent. If this method is called when on the last document in the series, the method has no effect.

Calling this method is the JavaScript equivalent of clicking the **Next Document** button on the toolbar when multiple documents are loaded in the viewer.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“initializeDocumentArray, openDocumentArray, and setDocumentArray method group” on page 302

Specifies the number of documents to open in a multiple document session. The setDocumentArray method then specifies the parameter file for each document where each file represents a successive document. The openDocumentArray method then opens the assembled document list at the provided document index.

“doc<N> parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

nextPage method

Displays the next page; that is, the current page + 1.

Syntax

```
ViewONE.nextPage();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

openDirectory method

Searches the specified path, including subdirectories, and opens the files in that path.

Syntax

The files are opened as one document with each file assumed to be a single-page image file.

```
ViewONE.openDirectory(String pathname)
```

The **pathname** parameter can be used to specify a local drive and directory. You must always use forward-slashes as the path separator.

This method is useful when the images are stored on hard disks, accessible locally or by using a mapped drive, and the user wants to view all files in a directory.

Requirements

- IBM Daeja ViewONE Standard, V3.1.112 or later
- IBM Daeja ViewONE Professional, V1.1.112 or later
- No additional modules

Example

```
ViewONE.openDirectory("c:/images");
```

openDoc method

Opens the document that is specified by the index parameter.

Syntax

```
ViewONE.openDoc(int index)
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray(numDocs)`, `setDocumentArray(filename, index)`, and `openDocumentArray()` method group JavaScript equivalent.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

The value of the index parameter represents the index of the chosen document within the ones that are listed with the **doc<N>**. Therefore, the following example causes IBM Daeja ViewONE to open the second document in the list that is specified by the **doc<N>** HTML parameter.

```
ViewONE.openDoc(2)
```

Related reference:

“initializeDocumentArray, openDocumentArray, and setDocumentArray method group” on page 302

Specifies the number of documents to open in a multiple document session. The `setDocumentArray` method then specifies the parameter file for each document where each file represents a successive document. The `openDocumentArray` method then opens the assembled document list at the provided document index.

“**doc<N>** parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

openFile method

Specifies the file name and initial page of the document to be viewed.

Syntax

```
ViewONE.openFile(String filename, int page)
```

The **filename** parameter can specify the file name relative to the code base or the full URL for the file. The code base is specified in the HTML code for the viewer.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.openFile("myimages.tif", 1);  
ViewONE.openFile("http://mysite/myimages.tif", 1);
```

openList(listFile, page) method

Opens the list of files that are specified by the **listFile** parameter. The **listFile** parameter allows a file to be supplied which contains a list of pages.

Syntax

```
ViewONE.openList(String listFile, int page)
```

This method provides an alternative to the `initializePageArray(numPages)`, `setPageArray(filename, page)`, `openPageArray(page)` method group methods.

This method is useful for very large documents because it removes the need to deal with an array in JavaScript.

This method can also be used to keep the HTML constant by changing the source list of documents instead of changing the HTML between different documents.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.openList("mylist/list.txt", 1);
```

Related reference:

“initializePageArray, openPageArray, and setPageArray method group” on page 303

Specifies the number of files (pages) in a list, then specifies each file where each file represents a successive page of the document, and then opens the assembled document at the provided page.

prevDoc method

Opens the previous document in the series of multiple documents that are loaded into the viewer.

Syntax

```
ViewONE.prevDoc()
```

This method applies only when multiple documents are loaded into the viewer by using the **doc<N>** HTML parameter or the `initializeDocumentArray(numDocs)`, `setDocumentArray(filename, index)`, and `openDocumentArray()` method group JavaScript equivalent. If this method is called when on the first document in the series, the method has no effect.

For information about the **doc<N>** parameter, see the *IBM Daeja ViewONE Parameters Reference Manual*.

Calling this method is the JavaScript equivalent of clicking the **Previous Document** button on the toolbar when multiple documents are loaded in the viewer.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“initializeDocumentArray, openDocumentArray, and setDocumentArray method group” on page 302

Specifies the number of documents to open in a multiple document session. The setDocumentArray method then specifies the parameter file for each document where each file represents a successive document. The openDocumentArray method then opens the assembled document list at the provided document index.

“doc<N> parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

previousPage method

Displays the previous page; that is, the current page - 1.

Syntax

```
ViewONE.previousPage();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

reloadList method

Forces the list that is used with the openList(listFile, page) method to be reloaded and the document to be reopened.

Syntax

```
ViewONE.reloadList()
```

This method reloads the list file from source, that is, the web server, each time. Therefore, any changes in the source file are picked up.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“openList(listFile, page) method” on page 307

Opens the list of files that are specified by the listFile parameter. The listFile parameter allows a file to be supplied which contains a list of pages.

setBackgroundImageEnabled method

Sets whether the background image is visible.

Syntax

```
ViewONE.setBackgroundImageEnabled(Boolean option)
```

If set to true, the background image is visible. If set to false, the background image is removed from the viewer.

This method is applicable only when used with COLD documents and where a background image is specified.

Requirements

- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setBackgroundImageEnabled(true);
```

setDocumentCookies method

Sets the cookies that are to be used for the next document request from the document server.

Syntax

This method is useful when the viewer is used in an Ajax style environment in which a user needs to log in to a resource remotely without refreshing the current viewer HTML page.

```
ViewONE.setDocumentCookies(String cookieslist)
```

Requirements

- IBM Daeja ViewONE Standard, V3.1.172 or later
- IBM Daeja ViewONE Professional, V1.1.172 or later
- No additional modules

Example

The cookies must be supplied in a semi-colon separated list, for example:

```
ViewONE.setDocumentCookies("mysessionid=29;userid=abc");
```

The format of the document.cookie JavaScript property is also supported:

```
ViewONE.setDocumentCookies(document.cookie);
```

setDocumentIndexListFile method

Updates the text-based-index-list of the viewer.

Syntax

```
ViewONE.setDocumentIndexListFile(String indexList)
```

You can use this method to dynamically specify a different file that contains the replacement text. The file that you specify must be a simple text file. Use a new line for each page index:

```
index1=Click for Page 1  
index2=Click for Page 2
```

Requirements

- IBM Daeja ViewONE Standard, V3.0.490 or later
- IBM Daeja ViewONE Professional, V1.0.490 or later
- No additional modules

Example

```
ViewONE.setDocumentIndexListFile("mylist.lst");
```

getDescription and setDescription method group

Gets or sets the description for the document.

Syntax

The description that is displayed in the status bar when a document is opened in IBM Daeja ViewONE. For example, the description might display "Opening docXYZ..." or "Opening docXYZ, page 2..." in the status bar.

```
ViewONE.getDescription() ViewONE.setDescription(String description)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var desc = ViewONE.getDescription();  
ViewONE.setDescription("myDoc");
```

setPage method

Sets the current page to the integer that is specified by the **page** parameter.

Syntax

```
ViewONE.setPage(int page)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setPage(2);
```

showExternalEmailImages method

Redisplays the email with the external images visible (true) or blocked (false).

Syntax

```
ViewONE.showExternalEmailImages(Boolean option)
```

This method requires that the **allowExternalEmailImages** HTML parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“**allowExternalEmailImages** parameter” on page 187

This parameter determines whether external email images can be loaded into a displayed email message.

isMultipageTif method

Checks whether the current document is a multiple page TIFF file.

Syntax

```
var ismulti = ViewONE.isMultipageTif();
```

You can call this method before you call any of the Save methods.

The method returns true if the file is a multiple page TIFF file. Otherwise, the method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

saveAsFormat method

Saves the current document, with the specified format.

Syntax

```
ViewONE.saveAsFormat(String filename, String format)
```

If the **filename** parameter is null, either the save dialog box is displayed or, if a server side save component was specified, the image is sent to the server-side component.

If the **format** parameter is null and the document is a single file, multiple page document, the source file is copied to the destination.

If the **format** parameter `tiff` and the document are not a single file, multiple page TIFF, then a new TIFF file is created.

Requirements

- IBM Daeja ViewONE Standard, V3.1.210 or later
- IBM Daeja ViewONE Professional, V1.1.210 or later
- No additional modules

Example

```
ViewONE.saveAsFormat("c:/temp/page.tif", "tiff");
```

save method

Saves either the current document or the document at the current index for multiple file documents (loaded by using the **doc<N>** parameter for instance) to the specified file name.

Syntax

```
ViewONE.save(String filename)
```

If file name is null, either the save dialog box is displayed or, if a server-side save component was specified, the image is sent to the server-side component.

The save method does not re-create the source file. Instead, the method copies the source file to the specified destination and renames it to the specified file name.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.save("c:/temp/image.tif");
```

savePage method

Saves the current page to the specified file name.

Syntax

```
ViewONE.savePage(String filename)
```

If file name is null, either the save dialog is displayed or, if a server side save component was specified, the image is sent to the server-side component.

If the document is a multiple page TIFF file, this method extracts the current page from the source TIFF file. The method then creates a new TIFF file that contains only that page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.savePage("c:/temp/page.tif");
```

saveSelected method

Saves the current selected pages of a multiple page TIFF document.

Syntax

```
ViewONE.saveSelected(String filename)
```

If the **filename** parameter is null, either the save dialog box is displayed or, if a server-side save component was specified, the image is sent to the server-side component.

Use this method only for multiple page TIFF documents. You cannot use this method to save selected pages from a multiple file document. Instead, you must call the `savePage(filename)` method for each page to be saved in the multiple file document.

To ensure that a call to the `saveSelected(filename)` method is valid, first make the following calls:

- Call the `isMultipageTif()` method to ensure that the document is a multiple page TIFF file.
- Call the `getSelection()` method to verify that the user has selected pages to be saved.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.saveSelected("c:/temp/selected.tif");
```

Related reference:

“savePage method” on page 312

Saves the current page to the specified file name.

“isMultipageTif method” on page 311

Checks whether the current document is a multiple page TIFF file.

“getSelection method” on page 266

Returns a comma-delimited string that contains the pages that the user selected.

setDefaultSaveFilename(filename) method

Specifies the file name that is to appear in the file name field of the save dialog box.

Syntax

This method provides a document-specific parameter and must be specified for each instance of IBM Daeja ViewONE.

```
ViewONE.setDefaultSaveFilename(String filename)
```

Requirements

- IBM Daeja ViewONE Standard, V3.1.96 or later
- IBM Daeja ViewONE Professional, V1.1.96 or later
- No additional modules

Example

```
ViewONE.setDefaultSaveFilename("myFile.tif");
```

setImageSaveURL method

Sets the URL that is to be used to save images.

The **saveURL** parameter specifies the URL.

Syntax

```
ViewONE.setImageSaveURL(String saveURL, Int type, Boolean useButton)
```

The **type** parameter indicates one of the following mechanisms for saving the images:

Type	Description
0	Save by using GET method. This type is deprecated.
1	Save by using the POST method.
2	Save by using the SERVLET method.

The **useButton** parameter determines whether the **Save** button in the GUI uses the specified URL. If this parameter is set to false, only embedded annotations are saved by using the specified URL.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setImageSaveURL("http://myServer/imageSave", 2, true);
```

Document text search

Document text search allows JavaScript control over the text-finding capability in IBM Daeja ViewONE Professional.

Tip: Currently, supported by the ViewONE PDF Module only. Other file formats cannot be searched. Trying to run any of the JavaScript methods that are listed in the following section when other file formats are loaded results in nothing.

cancelFind method

Cancels any find process that is running, including a find process that was started by the user from the IBM Daeja ViewONE user interface.

Syntax

```
ViewONE.cancelFind();
```

Requirements

- IBM Daeja ViewONE Professional, V1.1.70 or later
- ViewONE PDF Module

clearFindResults method

Clears both the search results that are shown in the thumbs tool pane and the search result highlights that are shown in the document-viewing area.

Syntax

```
ViewONE.clearFindResults();
```

Requirements

- IBM Daeja ViewONE Professional, V1.1.70 or later
- ViewONE PDF Module

find method

Starts a document find. The **text** parameter specifies the term to find.

Syntax

```
Viewone.find(String text, Boolean caseSensitive, Boolean wholeWord)
```

If the **caseSensitive** parameter is set to true, the search is case-sensitive. If the **wholeWord** parameter is set to true, then partial word matches are not returned.

If you call this method while another find is running, the existing find process is canceled. This action applies to a find that was started from the IBM Daeja

ViewONE user interface or from a JavaScript call.

Requirements

- IBM Daeja ViewONE Professional, V1.1.70 or later
- ViewONE PDF Module

Example

The following example returns the word *view*, but not the word *ViewONE*.

```
ViewONE.find("view", false, true);
```

nextFindResult method

Highlights the next find result in the search view of the viewer thumbnail pane and in the document-viewing area.

Syntax

```
ViewONE.nextFindResult();
```

If no find result is selected, the first find result is highlighted. Highlighting a find result also includes changing to the page that the find result is on. This method is equivalent to clicking the **Next result** button in the IBM Daeja ViewONE user interface.

Requirements

- IBM Daeja ViewONE Professional, V1.1.70 or later
- ViewONE PDF Module

prevFindResult method

Highlights the previous find result.

Syntax

Highlighting a find result also includes changing to the page that the find result is on. This method is equivalent to clicking the **Previous result** button in the IBM Daeja ViewONE user interface.

```
ViewONE.prevFindResult();
```

Requirements

- IBM Daeja ViewONE Professional, V1.1.70 or later
- ViewONE PDF Module

Event handlers and event handling

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

The IBM Daeja ViewONE viewer is an asynchronous threaded application. That is, the viewer does not necessarily wait for one thread to finish before starting another thread. Therefore, you might get unwanted results when you call one method directly after another. To avoid this problem, you can use the viewer's event handler to actively control the flow of events.

For example, suppose you want to use the `setZoom()` method to zoom to a particular zoom level when a document is opened. If you load the viewer and call the `setZoom()` method immediately, the method might not work depending on

how long it takes to render the image. Because the `setZoom()` method requires the rendered image to zoom in on, the method does not work if it is called before the image finishes rendering.

If you do not code defensively by using the event handler, users might experience intermittent faults in the code because of the timing of events. For example, if you do not use the event handler, the `setZoom()` method works for small documents that load quickly because the image is rendered before the method is called. However, for larger documents, the zoom is not set because the image does not finish rendering before the `setZoom()` method is called.

To avoid the timing problem with the `setZoom()` method, use the event handler monitor for event 38 to determine when the viewer is started and when the page is rendered. Then, call the `setZoom()` method after event 38 is fired.

You enable event handling by using the **eventHandler** HTML parameter and **eventInterest** HTML parameter. The **eventHandler** parameter sets the name of the JavaScript method that gets called. The **eventInterest** parameter specifies the identifiers of the events to be monitored.

When the viewer catches the specified events, it calls the method that is specified by the **eventHandler** parameter. The viewer passes to the event handler the following parameters:

id An integer that specifies the ID of the event that is to be monitored.

text

A string that provides information about the event.

The following code provides an example of a JavaScript event handler:

```
function myEventHandler(id, text)
{
    alert("Event received, id="+id+", text="+text);
}
```

You can design the event handler to perform any tasks that are needed. For example, the event handler might test the value of the **id** parameter and **text** parameter and then take further actions as necessary.

Important: You must include the **MayScript** parameter in the viewer definition for the event handler to work.

Related reference:

“Events IDs and descriptions”

IBM Daeja ViewONE can signal many different events to the event handler.

“**eventHandler** parameter” on page 176

This parameter determines the JavaScript method that handles events for the viewer.

“**eventInterest** parameter” on page 176

This parameter determine IBM Daeja ViewONE events are sent to the event logger.

Related information:

 [Configuring the client viewer for redactions](#)

Events IDs and descriptions

IBM Daeja ViewONE can signal many different events to the event handler.

The following table lists the events that the event handler can receive. The Virtual column indicates whether the event is available in IBM Daeja ViewONE Virtual. The JSON column indicates whether the event can pass a JSON object back to the handler.

ID	Virtual	JSON	Event text	Event trigger
1			<i>transform_action</i>	The user performed a transformation action on the document, such as flipping, mirroring, rotating, or inverting the document.
3	✓		printpage: page <i>n</i> of <i>n</i>	The user printed a page.
4			printvisible: page <i>n</i> of <i>n</i>	The user printed the visible part of a page.
5	✓		opened: <i>url</i>	The user opened a document.
6			saved: page <i>n</i> of <i>n</i>	The user saved a page.
7		✓	click: page <i>n</i> of <i>n</i>	The user clicked a page. The JSON object is supported in V4.0.32 or later. The object contains the following elements: <pre>{"pageX": value, "pageY": value, "page": {"number": value, "view": {"resolutionX": value, "resolutionY": value }}}</pre>
8		✓	dblclick: page <i>n</i> of <i>n</i>	The user double-clicked a page. The JSON object is supported in V4.0.32 or later. The object contains the following elements: <pre>{"pageX": value, "pageY": value, "page": {"number": value, "view": {"resolutionX": value, "resolutionY": value }}}</pre>
9	✓		page: page <i>n</i> of <i>n</i>	The user changed the page.
10			timeout: page <i>n</i> of <i>n</i>	The viewer timed out. For more information, see the <code>setTimeout(seconds)</code> method and the timeout HTML parameter.
11				Reserved.
12			select page: page <i>n</i> of <i>n</i>	The user selected a page.
13			unselect page: page <i>n</i> of <i>n</i>	The user cleared a page.
14		✓	mouse down: page <i>n</i> of <i>n</i>	The user pressed a mouse button. The JSON object is supported in V4.0.46 or later. The object contains the following elements: <pre>{"pageX": value, "pageY": value, "page": {"number": value, "view": {"resolutionX": value, "resolutionY": value }}}</pre>
15		✓	mouse up: page <i>n</i> of <i>n</i>	The user released a mouse button. The JSON object is supported in V4.0.46 or later. The object contains the following elements: <pre>{"pageX": value, "pageY": value, "page": {"number": value, "view": {"resolutionX": value, "resolutionY": value }}}</pre>
16				Reserved.

ID	Virtual	JSON	Event text	Event trigger
17				Reserved.
18				Reserved.
19				Reserved.
20	✓		set document: doc <i>n</i> of <i>n</i>	The user selected the next document in the list. This event is triggered only if the doc<N> HTML parameter is used.
21			end tab	The user pressed the tab key to change focus from another viewer component. The viewer assigns focus back to the first component in the list. You can use this event to override the viewer behavior and switch focus to an alternative component on the web page.
22	✓		ready	The viewer started and is ready to accept JavaScript calls.
23			annotation hyperlink	The user activated a JavaScript annotation hyperlink. For more information about hyperlinks, see the <i>Hyperlinks</i> topic.
24	✓		annotations save ok	The annotations were saved.
25	✓		annotations save failed	The annotation save operation failed.
26	✓		print canceled	A print dialog or print job was canceled.
27	✓		print ended	A print job was successfully sent to the printer.
28				Reserved.
29				Reserved.
30			annotation created	The user added an annotation through the user interface or with a JavaScript call.
31			SaveDocument Failed(<i>n</i>)	If the <i>n</i> value is 1, IBM Daeja ViewONE was unable to create the saved files in cache before the files were sent to the destination. For example, this event can occur if the disk is full. If the <i>n</i> value is 2, IBM Daeja ViewONE was unable to copy the saved files to the specified destination.
32			annotations updated: page <i>n</i> of <i>n</i>	The document annotations set was modified for the first time. This event is not fired again until the annotation data is saved and then modified again. Therefore, you can use this event to determine whether a document's annotations require saving.
33			annotations restored: page <i>n</i> of <i>n</i>	The document annotations set was restored. Therefore, you can use this event to determine whether a document's annotations do not require saving.
34	✓		Page rendered	The image for the current page was rendered or updated. This event occurs whenever the page is redrawn or refreshed. Events 38 and 39 provide filtered versions of this event.
35			Area Selected	The user selected an area of the current page by using the zoom area tool.
36				Reserved

ID	Virtual	JSON	Event text	Event trigger
37			Key pressed	<p>A key was pressed.</p> <p>Tip: To prevent IBM Daeja ViewONE from processing keys and generating this event, set the ProcessKeys HTML parameter to false.</p> <p>Tip: Keyboards can differ in the text value of keys and according to the setup of the system. Therefore, run a test to display the text of this event to verify the values.</p> <p>The following keys do not generate events:</p> <ul style="list-style-type: none"> • Keys that are pressed when the user edits a text annotation. • Shift, Ctrl, and Alt keys. Instead, the event includes a marker to indicate when these keys are pressed with another key. • Menu key <p>The following keys generate events and are also handled by the IBM Daeja ViewONE event if the ProcessKeys parameter is set to false:</p> <ul style="list-style-type: none"> • Scroll bar keys: Page Up, Page Down, Home, End, and arrow keys • Windows menu key <p>Modifier and cursor keys are defined by using key names as shown in the following examples:</p> <p>"A" The user pressed the A key.</p> <p>Control A The user pressed Ctrl+A.</p> <p>Shift A The user pressed Shift+A.</p> <p>Alt A The user pressed Alt+A.</p> <p>Alt Shift Control A The user pressed Alt+Shift_Ctrl+A.</p> <p>Space The user pressed the Spacebar key.</p> <p>Ctrl Space The user pressed Ctrl+Spacebar.</p> <p>Caps Lock The user pressed the Caps Lock key.</p> <p>Up The user pressed the Up Arrow key.</p> <p>Numpad-9 The user pressed the 9 key on the number pad. Tip: Some number pad keys use a hyphen as a separator rather than a space.</p>
38	✓		Full page rendered	The user viewed a page in the full-page pane. This event is fired only the first time that the user views the page.
39			All full pages rendered	The user viewed all document pages in the full-page pane. This event is fired only 1 time for a document.
40				Reserved
41			Keep Alive	This event is generated at regular intervals as specified by the KeepAliveTime HTML parameter. The text that accompanies the event is specified by the KeepAlive HTML parameter.
42			Save Complete	A document was saved successfully.

ID	Virtual	JSON	Event text	Event trigger
43			Cache Access Denied	All attempts to write to an IBM Daeja ViewONE cache failed.
44			Request alternative URL	The viewer is preparing to retrieve a document or image. The event supplies the URL that the viewer is using to open the document or image. The event handler can return a different URL by calling the <code>setEventHandlerResponse(text)</code> method with the text parameter set by using the format " <code>url:url</code> ". In this case, the viewer uses the specified URL to retrieve the document or image.
45			Request alternative URL upon error	The viewer attempted to retrieve a document or image and encountered a <code>FileNotFoundException</code> condition. The event supplies the URL that the viewer is using to open the document or image. The event handler can return a different URL by calling the <code>setEventHandlerResponse(text)</code> method with the text parameter set by using the format " <code>url:url</code> ". In this case, the viewer uses the specified URL to retrieve the document or image.
46	✓		Magnifier opened/closed	The magnifier was opened or closed.
47			Save dialog cancelled	The user canceled the Save dialog box for saving a document or page.
48		✓	Annotations burnt	Annotation burning on the server finished and the client was notified. The JSON object is supported in V4.0.38 or later. The object contains the following element: <code>customServerResponse</code> Tip: You must modify your server code to populate the text.
49				Reserved.
50				Reserved.
51			Annotation burn canceled by user (Version 3.1.106 onwards)	The user canceled the annotation burning.
52		✓	Annotation burn failed (Version 3.1.106 onwards)	The client was notified that the annotation burning failed. The JSON object is supported in V4.0.38 or later. The object contains the following element: <code>customServerResponse</code> Tip: You must modify your server code to populate the text.
57	✓		Print start (version 3.1.158 or later)	A printing operation is about to start. This event is fired only 1 time per print operation. For example, if the user prints 3 pages, only a single event is fired. If a user clicks the print button but cancels the print from the print dialog box before printing starts, this event is not fired.
58			Start page (version 3.1.172 or later)	The user selected a page for display. If the page needs to be downloaded, this event is fired first.

ID	Virtual	JSON	Event text	Event trigger
59	✔		Annotations loaded OK (version 3.1.198 or later)	The annotations were loaded into the viewer with no errors. If annotation data is streamed, this event is fired on a per page basis and the page number is supplied with the event.
60	✔		Annotations load failed (version 3.1.198 or later)	Retrieval of annotation data from the server-side component failed.
61	✔		Document load failed (version 4.0.4 or later)	A document failed to load. This problem can occur if the document to be displayed cannot be found, the annotations failed to load, or the document cannot be displayed.
62	✔		Page load failed (version 4.0.4 or later)	One page in a multiple page document cannot be displayed. This problem can occur if the page to be displayed cannot be found or another problem prevented the page from being displayed.
63	✔		Thumbnail load failed (version 4.0.4 or later)	An image to be displayed in a thumbnail cannot be displayed. This problem can occur if the thumbnail to be displayed cannot be found or another problem prevented the thumbnail from being displayed.
64				Reserved.
65	✔		<i>annotation_label</i>	The user activated an annotation by right-clicking the annotation. This event is also fired when a user finishes drawing the outline of a text box to create a text annotation. The new annotation is activated to allow text entry.
66	✔		<i>annotation_label</i>	The user clicked the document outside the selected annotation and caused the annotation to be deactivated.
67	✔	✔	Annotation Deleted (version 4.0.38 or later)	The user deleted an annotation by using the user interface or a JavaScript call. The event returns a JSON object that contains an array with the labels of all deleted annotations. If only 1 annotation was removed, the returned array contains only 1 element. This event is not fired if the user removed annotations by using the Restore annotations button. The JSON object contains the following elements: { "annotationLabels": [array of values] }
68	✔	✔	Mouse move	The user moved the mouse on a page. The JSON object is supported in V4.0.42 or later. The object contains the following elements: { "pageX": <i>value</i> , "pageY": <i>value</i> , "page": { "number": <i>value</i> , "view": { "resolutionX": "200.0dpi", "resolutionY": <i>value</i> } } }
69		✔	Drag box start	The user started the dragging action to create the rectangle. The event contains the origin of the rectangle relative to the currently viewed page. The following code provides an example of the JSON object that is returned for this event: { "pageX": "0px", "pageY": "0px", "page": { "number": 1, "view": { "resolutionX": "200.0dpi", "resolutionY": "200.0dpi" } }, "dragRectangle": { "x": "0px", "y": "0px" } }

ID	Virtual	JSON	Event text	Event trigger
70		✔	Drag box update	<p>The user updated the rectangle by moving the mouse. The event contains the origin, width, and height (which might be negative) relative to the viewed page.</p> <p>The following code provides an example of the JSON object that is returned for this event:</p> <pre>{ "pageX": "0px", "pageY": "0px", "page": { "number": 1, "view": { "resolutionX": "200.0dpi", + "resolutionY": "200.0dpi" } }, "dragRectangle": { "x": "0px", "y": "0px", "width": "0.0px", + "height": "0.0px" } }</pre>
71		✔	Drag box end	<p>The user completed the dragging action. The event contains a status, the origin, width, and height (which might be negative) relative to the viewed page.</p> <p>The following code provides an example of the JSON object that is returned for this event:</p> <pre>{ "pageX": "0px", "pageY": "0px", "page": { "number": 1, "view": { "resolutionX": "200.0dpi", "resolutionY": "200.0dpi" } }, "dragRectangle": { "x": "0px", + "y": "0px", "width": "0.0px", "height": "0.0px" }, "status": 2 }</pre> <p>The status argument indicates how the end event completed:</p> <ol style="list-style-type: none"> (complete) The drag rectangle was completed successfully. Any processing that requires the coordinates of the completed rectangle can use them. (user abort) The user aborted the drag, for example, by pressing escape. (system abort) The system aborted the drag, for example, by shutdown. (error) The drag was not completed.
72			Return image area success	The grabImage(..) method successfully completed. The text element that is returned by the event contains the base64 encoded string that represents the image data requested.
73			Return image area failure	The grabImage(..) method failed to complete. The text element that is returned by the event contains the reason the method failed.

Related tasks:

“Hyperlinks” on page 88

You can use the hyperlink parameters determine the default base location for web hyperlinks and specifies a list of optional targets for web hyperlinks to use.

Related reference:

“Event handlers and event handling” on page 315

You can use a JavaScript event handler to monitor user activity and other selected actions that are performed by IBM Daeja ViewONE.

“timeOut parameter” on page 205

This parameter sets and starts a usage timer.

“setTimeout method” on page 363

Sets and starts a usage timer.

“**doc**<N> parameter” on page 191

This parameter determines the parameters file that is associated with a document. The parameters file specifies the document and the parameters that are used to display the document.

Obtain additional information from an event

Starting with IBM Daeja ViewONE, V4.0.38, the event handling is extended to allow additional information to be passed back with the event.

This data is contained in a JSON object that can be parsed to retrieve further information.

To enable the event handler to pass back additional information, set the **eventHandlerJSON** HTML parameter to true.

The following code provides an example of an event handler when the **eventHandlerJSON** parameter is enabled:

```
function myEventHandler(id, text,json)
{
    alert("Event received, id="+id+", text="+text+ "JSON="+json);
}
```

Only certain events support the JSON object for including additional information.

Related reference:

“Events IDs and descriptions” on page 316

IBM Daeja ViewONE can signal many different events to the event handler.

“**eventHandlerJson** parameter” on page 176

This parameter determines that JavaScript method that is to handle events. You can use this parameter to extend the standard **eventHandler** parameter to allow a server-side process to return a **JSON** object. This **JSON** object contains a map of parameters that can be parsed. You can specify extra actions to be taken depending on your requirements.

Image options

You can use the image methods to manipulate image properties.

applyRotationArray, initializeRotationArray, and setRotationArray method group

The `initializeRotationArray` method initializes the rotation array, which is used for setting the rotation of each page of the current document, to the specified size. The `setRotationArray` method sets the rotation for given page to the specified angle. The `applyRotationArray` method applies the rotation array that was set up by using the `initializeRotationArray` method and `setRotationArray` method to the current document.

Syntax

```
ViewONE.applyRotationArray();
viewONE.initializeRotationArray(int size)
viewONE.setRotationsArray(int angle, int page)
```

Use the `initializeRotationArray(int size)` method to set the size of the rotation array before you call the `setRotationArray(int angle, int page)` method.

In the `setRotationArray(int angle, int page)` method, the specified angle, which is in degrees, must be set to 0, 90, 180 or 270. The pages are indexed from 0, so the first page is 0, and the second page is 1.

The `setRotationArray(int angle, int page)` method returns true if the page rotation was successfully set. The method returns false if the page rotation was not set. For example, the page rotation is not set if the specified page is an invalid index in the rotation array that was initialized by using the `initializeRotationArray()` method.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Examples

```
ViewONE.initializeRotationArray(4);  
var ok = ViewONE.setRotationArray(90, 0);
```

endUserDragMode method

Ends the current drag mode operation without waiting for the user or another event to end it.

Syntax

```
ViewONE.endUserDragMode(var status)
```

The *status* variable affects the status that is passed to the drag box end event. The valid status codes are in the following table:

- complete
- user abort
- system abort
- error

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- No additional modules

Example

```
ViewONE.endUserDragMode("complete")
```

Related reference:

“startUserDragMode method” on page 341

Puts the viewer in drag mode to enable the user to draw a drag box. The user clicks in the document and drags the mouse to draw a box from the origin to the current mouse position.

“isUserDragMode method” on page 333

Returns a Boolean value of true if drag mode is active.

getBrightness method

Returns the current brightness setting as a percentage 0 - 100.

Syntax

```
ViewONE.getBrightness()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var percent = ViewONE.getBrightness();
```

getContrast method

Returns the current contrast setting as a percentage 0 - 100.

Syntax

```
ViewONE.getContrast()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var percent = ViewONE.getContrast();
```

getEnhanceMode method

Returns the current enhance state.

Syntax

```
ViewONE.getEnhanceMode()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var state = ViewONE.getEnhanceMode();
```

getFlip method

Returns the flip mode as an integer.

Syntax

```
ViewONE.getFlip()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var angle = ViewONE.getFlip();
```

getImageHeight method

Returns an integer value that represents the height of image that is displayed.

Syntax

```
ViewONE.getImageHeight()
```

The height is specified in image pixels.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var height = ViewONE.getImageHeight();
```

getImageWidth method

Returns an integer that represents the width of the image that is displayed.

Syntax

```
ViewONE.getImageWidth()
```

The value is specified in image pixels.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var width = ViewONE.getImageWidth();
```

getLuminance method

Returns the current luminance setting as a percentage 0 - 100.

Syntax

```
ViewONE.getLuminance()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var percent = ViewONE.getLuminance();
```

getRotation method

Returns the angle of rotation of the current page as an integer.

Syntax

```
var angle = ViewONE.getRotation();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getScale method

Returns the scale value as an integer.

Syntax

```
ViewONE.getScale()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.1 or later
- No additional modules

Example

```
var scale = ViewONE.getScale();
```

getStates and setStates method group

The `getStates` method returns a coded string. It is used with the `setStates` method, which sets states to the values specified by the coded string.

Syntax

```
var states = ViewONE.getStates();  
ViewONE.setStates(String states);
```

Call the `setStates(String states)` method before you open a document. This method sets the following states:

- Zoom level
- Scroll
- View type, such as single page
- Scale
- Flip mode
- Rotation
- Whether the page is inverted
- Brightness
- Contrast
- Luminance

When the `getStates()` method is called while a document is open, it returns a string that contains the previous information.

The `getStates()` method and `setStates()` methods together permit the viewing states to be restored when a document is closed and reopened.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getXResolution method

Returns an integer value that represents the x-axis resolution of the image that is displayed.

Syntax

```
var xRes = ViewONE.getXResolution();
```

The returned value is specified in image dots per inch.

The value is obtained from the image header information. If the header information is missing or corrupted, the returned value is invalid.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getYResolution method

Returns an integer value that represents the y-axis resolution of the image that is displayed.

Syntax

```
var yRes = ViewONE.getYResolution();
```

The returned value is specified in image dots per inch.

The value is obtained from the image header information. If the header information is missing or corrupted, the returned value is invalid.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getXScroll method

Returns the position of the horizontal scroll bar as a percentage of the entire scroll bar.

Syntax

```
var xScroll = ViewONE.getXScroll();
```

The leftmost position is 0 and the rightmost position is 100.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getYScroll method

Returns the position of the vertical scroll bar as a percentage of the entire scroll bar.

Syntax

```
var yScroll = ViewONE.getYScroll();
```

The top position is 0 and the bottom position is 100.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getZoom method

Returns the current zoom level as a double value.

Syntax

```
var zoom = ViewONE.getZoom();
```

For example, a zoom level of 1.0 equals 100%, in which case 1 image pixel equals 1 screen pixel. A zoom level of 2.0 equals 200%, in which case one image pixel equals two screen pixels.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

grabImage method

Returns a base64 string that represents the data for the specified area of an image that is in the viewer. You can use this data to apply basic transformations such as rotation and scaling to the area. Use this method to select an area that you can clip, extract, or save. This image then can be displayed in the viewer.

Syntax

```
ViewONE.grabImage(var mimeType,var mimeTypeProperties, var encoding, var clipArea,  
var currentState)
```

When the user is drawing a drag box and releases the mouse, the grabImage method is called with all of the information, including start and end coordinates. The grabImage method gathers the following information, and to obtain that information, you must enable an event handler for JavaScript event 72.

mimeType

The MIME type of the output. The supported MIME type values are image/png, image/tiff, and image/jpeg.

mimeTypeProperties

This variable is not currently used; therefore, the value is ignored.

encoding

The encoding type to be used to represent the image data. You must set this parameter to base64.

clipArea

An instance of the JSObject class that defines the area to clip from the original image. You must include the following numeric attributes in the object: x-axis starting point, y-axis starting point, width, and height.

currentState

An instance of the JSObject class that defines the state of the area to clip. You must include the following numeric attributes in the object: scale and rotate.

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- No additional modules

Example

The following script shows how you might use the grabImage method to paste a selected area of an image outside of the viewer. This script assumes that you created a custom button that the user clicks to select the area.

<script>

```
// Evaluate the status of the custom button (up or down)
function ev()
{
    var status = ViewONE.isUserDragMode();
    if (status)
        return 7;
    else
        return 5;
}

// This function is fired by the custom button
function toggle()
{
    if (ViewONE.isUserDragMode())
    {
        ViewONE.endUserDragMode("user abort");
    }
    else
    {
        ViewONE.startUserDragMode("style");
    }
}

// The event handler
function func(id,text,jRect)
{
    if (id == 71) // Drag MOUSE END
    {
        tellViewerToGrabTheImage(jRect)
    }
    else if (id == 72) // GRAB IMAGE SUCCESS
    {
        pasteTheImageDataOnScreen(text);
    }
    else if (id == 73) //GRAB IMAGE FAIL
    {
        alert("ERROR : "+ text); // The reason for the failure.
    }
}

function tellViewerToGrabTheImage(jRect)
{
    var jShape = eval('(' + jRect + ')');

    if (jShape.status != 2)
    {
```



```

// Clean the values

// The width or height might be negative
// (in case the mouse moved, for instance, from right to left)
var realX = cleanVal(jShape.dragRectangle.x);
var realY = cleanVal(jShape.dragRectangle.y)
var realWidth = cleanVal(jShape.dragRectangle.width);
var realHeight = cleanVal(jShape.dragRectangle.height);

if (realWidth < 0)
{
    realWidth = Math.abs(realWidth);
    realX = realX - realWidth;
}
if (realHeight < 0)
{
    realHeight = Math.abs(realHeight);
    realY = realY - realHeight;
}
// Build a shape with the selected area of the screen
var shape =
{
    x : realX,
    y : realY,
    width : realWidth,
    height : realHeight
};

// Build a mimeTypeProperty object
var mimeTypeProp =
{
}
// Specify any transformation to be applied to the output image
var stateObj =
{
    rotate : "0deg", // no rotation applied
    scale : "1"      // no zoom applied
}

// call grabImage to generate a string representing the selected area
try
{
    ViewONE.grabImage("image/png",mimeTypeProp,"base64",shape, stateObj);
    // when the viewer finishes processing this function, event 72 gets
    // called (or 73 in case of failure)
}
catch(e)
{
    alert("Javascript Exception caught when calling ViewONE.grabImage(..);
        The reason is: "+e.message);
}
}
}

function pasteTheImageDataOnScreen(text)
{
    var image = text;
    // the text coming from the event handler contains the image data

    var img = document.createElement('img');
    img.src = 'data:image/png;base64,'+image;
    document.getElementById("gotImage").appendChild(img);
}

function cleanVal(string)
{

```

```

var value;
if (string.indexOf(".") > 1)
{
    var elements = string.split(".");
    value = parseInt(elements[0]);
}
else if (string.indexOf("px") > -1)
{
    var elements = string.split("p");
    value = parseInt(elements[0]);
}
return value;
}

function cleanDiv()
{
    document.getElementById("gotImage").innerHTML = "";
}

</script>

```

Related reference:

“Events IDs and descriptions” on page 316

IBM Daeja ViewONE can signal many different events to the event handler.

invert method

Inverts the display colors, for example, black changes to white and white changes to black.

Syntax

```
ViewONE.invert();
```

This method is also effective on images with more than two colors. A second call to this method re-establishes the original display colors.

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isDraggingEnabled method

Returns a Boolean value of true if the user can drag the image, false otherwise.

Syntax

```
var draggingEnabled = ViewONE.isDraggingEnabled();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isEnhance method

Returns a Boolean true if enhance is on. Otherwise, this method returns false.

Syntax

Enhance provides anti-aliasing to images to smooth the outline and edges to make text more readable.

```
ViewONE.isEnhance(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
var enhance = ViewONE.isEnhance(true);
```

isInverted method

Returns a Boolean true if the colors are inverted.

Syntax

Otherwise, this method returns false.

```
var inverted = ViewONE.isInverted();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isUserDragMode method

Returns a Boolean value of true if drag mode is active.

Syntax

```
var dragModeEnabled = ViewONE.isUserDragMode()
```

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- No additional modules

Related reference:

“startUserDragMode method” on page 341

Puts the viewer in drag mode to enable the user to draw a drag box. The user clicks in the document and drags the mouse to draw a box from the origin to the current mouse position.

“endUserDragMode method” on page 324

Ends the current drag mode operation without waiting for the user or another event to end it.

resetBrightness method

Resets the brightness level to 50%, the default value.

Syntax

```
ViewONE.resetBrightness();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

resetContrast method

Resets the contrast level to 50%, the default value.

Syntax

```
ViewONE.resetContrast();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

resetLuminance method

Resets the luminance level to 50%, which is the default value.

Syntax

```
ViewONE.resetLuminance();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

rotateClockwise method

Increases the rotation by 90 degrees.

Syntax

```
ViewONE.rotateClockwise();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version

- No additional modules

rotateCounterclockwise method

Decreases the rotation by 90 degrees.

Syntax

```
ViewONE.rotateCounterclockwise();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

rotate180 method

Rotates the page 180 degrees.

Syntax

```
ViewONE.rotate180();
```

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

setBrightness method

Sets the brightness of the display.

Syntax

```
ViewONE.setBrightness(int percent)
```

The value represents a percentage 0 - 100, with 50 being the default value. The value 0 indicates the minimum brightness (dark) and the value 100 indicates the maximum brightness (light).

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setBrightness(60);
```

setContrast method

Sets the contrast of the image that is displayed.

Syntax

```
ViewONE.setContrast(int percent)
```

The value represents a percentage 0-100, with 50 being the default value. The value 0 indicates the minimum contrast (flat) and the value 100 indicates the maximum contrast (not flat).

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setContrast(40);
```

setDraggingEnabled method

Specifies whether the user can use the mouse to drag the image.

Syntax

Dragging the image to the right pans the image to the right. Dragging the image to the left pans the image to the left.

```
ViewONE.setDraggingEnabled(Boolean option)
```

A value of true (default) indicates that the user can drag the image.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setDraggingEnabled(true);
```

setEnhanceMode method

Specifies the anti-aliasing algorithm that is to be used.

Syntax

```
ViewONE.setEnhanceMode(int mode)
```

The following table lists the valid values for the **mode** parameter.

| mode | Description |
|-------------|-----------------------------------------------|
| 0 | Enhance off |
| 1 | Enhance level (typical images) |
| 2 | Enhance level (drawings) |
| 3 | Enhance level (very high-resolution drawings) |

You can use this method to turn anti-aliasing on or off. You do not need to use both this method and the `setEnhance` method together.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setEnhanceMode(1);
```

setEnhance method

Specifies whether a monochrome image is displayed with anti-aliasing on or off.

Syntax

```
ViewONE.setEnhance(boolean option)
```

If set to true, anti-aliasing is turned on.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setEnhance(true);
```

setFlip method

Specifies the flip mode of displayed pages.

Syntax

```
ViewONE.setFlip(int mode)
```

The following table lists the valid values for the **mode** parameter. The default value is 0.

| mode | Description |
|-------------|------------------------------|
| 0 | None |
| 1 | Horizontal or mirror |
| 2 | Vertical |
| 3 | Both horizontal and vertical |

Flip buttons and menus are not visible to the user by default. To enable these options for the user, you must use the **flipOptions** HTML parameter when the applet is loaded.

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setFlip(1);
```

Related reference:

“**flipOptions** parameter” on page 165

This parameter determines whether the buttons and menu options are available to enable users to flip the visible page horizontally and vertically.

setInverted method

Sets the invert of the display colors, for example, black changes to white and white changes to black.

Syntax

This method is also effective on images with more than two colors

```
ViewONE.setInverted(Boolean option)
```

If the document is closed after this method is called, the default for all pages is modified. If the document is open after this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setInverted(true);
```

setLuminance method

Sets the luminance of the image that is displayed.

Syntax

```
ViewONE.setLuminance(int percent)
```

The value represents a percentage 0 - 100, with 50 being the default value. The value 0 indicates the minimum luminance (dull) and the value 100 indicates the maximum luminance (bright).

If the document is closed when this method is called, the default for all pages is modified. If the document is open when this method is called, only the individual page is modified.

Brightness increases the brightness of all colors (dark and light) uniformly. Luminance causes already bright areas of the image to increase in brightness further and darker areas to increase in brightness also, but by a lesser amount. Thus, luminance simulates a light source that is shining on the image, which can make color images clearer to read.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setLuminance(70);
```

setRotation method

Specifies the angle at which pages are displayed.

Syntax

```
ViewONE.setRotation(int angle)
```

Values of 90, 180, or 270 are accepted. The default is 0.

If the document is closed after this method is called, the default for all pages is modified. If the document is open after this method is called, only the individual page is modified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setRotation(90);
```

setScale method

Specifies the scale mode to be used to display a page.

Syntax

```
ViewONE.setScale(int scale)
```

The following table lists the valid values for the **scale** parameter.

| scale | Description |
|--------------|----------------------------------------------------------------------------------------------------|
| 0 | Best fit

The page is scaled to fit into the window area so that the entire page is visible. |

| scale | Description |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | <p>Fit-to-window-width</p> <p>The page is scaled so that the width of the page matches the width of the window area.</p> <p>This setting can result in the visible page height that exceeds the available window height. In this situation, a vertical scroll bar appears automatically.</p> |
| 2 | <p>Fit-to-window-height</p> <p>The page is scaled so that the height of the page matches the height of the window area.</p> <p>This setting can result in the visible page width that exceeds the available window width. In this situation, a horizontal scroll bar appears</p> |

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setScale(0);
```

setXYScroll method

Sets the scroll bars to the specified positions.

Syntax

```
ViewONE.setXYScroll(int x, int y)
```

The **x** and **y** parameters are specified in screen pixels.

Tip: Problems can occur if you use this method with the `setZoom(zoom)` method. Instead, call the `setZoomAndXYScroll(zoom, x, y)` method.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setXYScroll(20,20);
```

Related reference:

“setZoom method” on page 341
Zooms to the specified value.

“setZoomAndXYScroll method”
Sets the zoom and the scroll-bars position to the specified value.

setZoomAndXYScroll method

Sets the zoom and the scroll-bars position to the specified value.

Syntax

```
ViewONE.setZoomAndXYScroll(int zoom, int x, int y)
```

The **zoom** parameter is specified as a double value. For example, 1.0 equals 100%, in which case 1 image pixel equals one screen pixel. 2.0 equals 200%, in which case 1 image pixel equals two screen pixels.

The **x** and **y** parameters are specified in screen pixels.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setZoomAndXYScroll(2.0, 100, 100);
```

setZoom method

Zooms to the specified value.

Syntax

```
ViewONE.setZoom(int zoom)
```

The **zoom** parameter is specified as a double value. For example, 1.0 equals 100%, in which case 1 image pixel equals one screen pixel. 2.0 equals 200%, in which case 1 image pixel equals two screen pixels.

Tip: Problems can occur if you use this method with the `setXYScroll(x, y)` method. Instead, call the `setZoomAndXYScroll(zoom, x, y)` method.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setZoom(2.0);
```

Related reference:

“setXYScroll method” on page 340

Sets the scroll bars to the specified positions.

“setZoomAndXYScroll method” on page 340

Sets the zoom and the scroll-bars position to the specified value.

startUserDragMode method

Puts the viewer in drag mode to enable the user to draw a drag box. The user clicks in the document and drags the mouse to draw a box from the origin to the current mouse position.

Syntax

```
ViewONE.startUserDragMode(var boxStyle)
```

The style feature is not implemented, so the *boxStyle* value is ignored.

If the user moves the mouse outside of the current scroll region, the viewer automatically scrolls to keep the current position in the view window. While the user is in drag mode, JavaScript events are fired to notify custom scripts of progress.

Returns true if the call is successful and events are fired. This method returns false if there a problem occurs when drag mode is initializing.

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- No additional modules

Related reference:

“endUserDragMode method” on page 324

Ends the current drag mode operation without waiting for the user or another event to end it.

“isUserDragMode method” on page 333

Returns a Boolean value of true if drag mode is active.

zoomArea method

Zooms to a specified area.

Syntax

If the specified area is a different aspect ratio from the display area, the viewer attempts to fit the zoom as best it can.

```
ViewONE.zoomArea(int x, int y, int width, int height, Boolean highlight,  
int seconds)
```

x An integer that specifies the starting point of the zoom area on the x axis in image pixels.

y An integer that specifies the starting point of the zoom area on the y axis in image pixels.

width

An integer that specifies the width of the zoom area in image pixels.

height

An integer that specifies the height of the zoom area in image pixels.

highlight

A Boolean value that if set to true causes the zoom area to be highlighted.

seconds

An integer value that when the **highlight** parameter is set to true determines how long the zoom area is highlighted. If the **seconds** parameter is greater than 0, the zoom area is highlighted for the specified number of seconds. If the parameter is less than or equal to 0, the zoom area is highlighted until the user clicks or forces a refresh by scrolling or rotating the page.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.zoomArea(400, 500, 100, 150, true, 2);
```

zoomIn method

Applies a 25% increase in zoom.

Syntax

```
ViewONE.zoomIn();
```

IBM Daeja ViewONE first attempts to use the scale modes such as fit-to-width, fit-to-height, or best-fit if they are more appropriate. The `zoomIn()` method increases the zoom factor only after the scale modes are no longer suitable.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

zoomOut method

Reverses the effect of `zoomIn()` method.

Syntax

```
ViewONE.zoomOut();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Related reference:

“`zoomIn` method”

Applies a 25% increase in zoom.

zoom100 method

Zooms image to 100% (full resolution).

Syntax

```
ViewONE.zoom100();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Menus and keyboard shortcuts

The following APIs determine whether menus and keyboard shortcuts are enabled.

isAllKeys method

Returns a Boolean value of true if all keyboard shortcuts are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isAllKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isAllMenus() method

Returns a Boolean value of true if all menus are enabled.

Syntax

```
var menusEnabled = ViewONE.isAllMenus();
```

Otherwise, this method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isCacheMenus method

Returns a Boolean value of true if the cache menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menusEnabled = ViewONE.isCacheMenus();
```

Requirements

- IBM Daeja ViewONE Professional, V1.0.808 or later
- No additional modules

isFileKeys method

Returns a Boolean value of true if the file keys are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isFileKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isFileMenus method

Returns a Boolean value of true if the file menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menuEnabled = ViewONE.isFileMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isImageKeys method

Returns a Boolean value of true if the keyboard shortcuts for the image operations are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isImageKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isImageMenus method

Returns a Boolean value of true if the menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menuEnabled = ViewONE.isImageMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isPageKeys method

Returns a Boolean value of true if the keyboard shortcuts for page operations are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isPageKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isPageMenus method

Returns a Boolean value of true if the menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menuEnabled = ViewONE.isPageMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isPreferenceMenus method

Returns a Boolean value of true if the preference menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menuEnabled = ViewONE.isPreferenceMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isPrintKeys method

Returns a Boolean value of true if the keyboard shortcuts for print operations are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isPrintKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isPrintMenus method

Returns a Boolean value of true if the menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menuEnabled = ViewONE.isPrintMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isSelectKeys method

Returns a Boolean value of true if the keyboard shortcuts for select operations are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isSelectKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isSelectMenus method

Returns a Boolean value of true if the select menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menusEnabled = ViewONE.isSelectMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isViewKeys method

Returns a Boolean value of true if the keyboard shortcuts for view operations are enabled.

Syntax

Otherwise, this method returns false.

```
var keysEnabled = ViewONE.isViewKeys();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isViewMenus method

Returns a Boolean value of true if the view menus are enabled.

Syntax

Otherwise, this method returns false.

```
var menusEnabled = ViewONE.isViewMenus();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

setAllKeys method

Specifies whether all keyboard shortcuts are enabled.

Syntax

```
ViewONE.setAllKeys(Boolean option)
```

Call this method with a value of true to enable the keyboard shortcuts. By default, the keyboard shortcuts are enabled.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setAllKeys(true);
```

Related reference:

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setAllMenus method

Specifies whether the pop-up menus are available.

Syntax

```
ViewONE.setAllMenus(Boolean option)
```

Call this method with a value of true to make all menus available. By default, the menus are available.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setAllMenus(true);
```

Related reference:

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setCacheMenus(true/false) method

Specifies whether the cache pop-up menus are available.

Syntax

```
ViewONE.setCacheMenus(Boolean option)
```

Call this method with a value of true to make the menus available. By default, the menus are available.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Professional, V1.0.808 or later
- No additional modules

Example

```
ViewONE.setCacheMenus(true);
```

Related reference:

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setFileKeys method

Specifies whether the keyboard shortcuts for file operations are enabled.

Syntax

```
ViewONE.setFileKeys(Boolean option)
```

Call this method with a value of true to enable the keyboard shortcuts. By default, the keyboard shortcuts are enabled.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setFileKeys(true);
```

Related reference:

“**fileKeys** parameter” on page 183

This parameter determines whether the keyboard shortcuts for file operations are enabled.

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setFileMenus method

Specifies whether the file pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setFileMenus(Boolean option)
```

Call this method with a value of true to make the menus available. By default, the menus are available.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, V3.1.184 or later
- IBM Daeja ViewONE Professional, V1.1.184 or later
- No additional modules

Example

```
ViewONE.setFileMenus(true);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setImageKeys method

Specifies whether the keyboard shortcuts for image operations are enabled.

Syntax

```
ViewONE.setImageKeys(Boolean option)
```

Call this method with a value of true to enable the keyboard shortcuts. By default, the keyboard shortcuts are enabled.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setImageKeys(true);
```

Related reference:

“**imageKeys** parameter” on page 167

This parameter determines whether the keyboard shortcuts for image operations are enabled.

setImageMenus method

Specifies whether the image pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setImageMenus(Boolean option)
```

Call this method with a value of true to make the menus available. By default, the menus are available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setImageMenus(true);
```

setPageKeys method

Specifies whether the keyboard shortcuts for page operations are enabled.

Syntax

```
ViewONE.setPageKeys(Boolean option)
```

Call this method with a value of true to enable keyboard shortcuts. Keyboard shortcuts are enabled by default.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setPageKeys(true);
```

Related reference:

“**pageKeys** parameter” on page 248

This parameter determines whether the keyboard shortcuts for page operations are enabled.

setPageMenus method

Specifies whether the page pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setPageMenus(Boolean option)
```

Call this method with a value of true to make the page menus available. By default, the page menus are available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setPageMenus(true);
```

setPreferenceMenus method

Specifies whether the preference pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setPreferenceMenus(Boolean option)
```

Call this method with a value of true to make the preference menus available. By default, the preference menus are available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setPreferenceMenus(true);
```

setPrintKeys method

Specifies whether the keyboard shortcuts for print operations are enabled.

Syntax

```
ViewONE.setPrintKeys(Boolean option)
```

Call this method with a value of true to enable the keyboard shortcuts. By default, the keyboard shortcuts are enabled.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setPrintKeys(true);
```

Related reference:

“**printKeys** parameter” on page 222

This parameter determines whether the keyboard shortcuts for print operations are enabled.

“**javascriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setPrintMenus method

Specifies whether the print pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setPrintMenus(Boolean option)
```

Call this method with a value of true to make the print menus available. By default, the print menus are available.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setPrintMenus(true);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setSelectKeys method

Specifies whether the keyboard shortcuts for select operations are enabled.

Syntax

```
ViewONE.setSelectKeys(Boolean option)
```

Set this method to true to enable the keyboard shortcuts.

Call this method with a value of true to enable keyboard shortcuts. Keyboard shortcuts are enabled by default.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setSelectKeys(true);
```

Related reference:

“**selectKeys** parameter” on page 250

This parameter specifies whether the keyboard shortcuts for page selections are enabled.

setSelectMenus method

Specifies whether the select pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setSelectMenus(Boolean option)
```

Call this method with a value of true to make the select menus available. By default, the select menus are available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
ViewONE.setSelectMenus(true);
```

setViewKeys method

Specifies whether the keyboard shortcuts for view operations are enabled.

Syntax

```
ViewONE.setViewKeys(Boolean option)
```

Call this method with a value of true to enable keyboard shortcuts. Keyboard shortcuts are enabled by default.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setViewKeys(true);
```

Related reference:

“**viewKeys** parameter” on page 254

This parameter specifies whether the keyboard shortcuts for view operations are enabled.

setViewMenus method

Specifies whether the view pop-up menus are available when the user clicks the middle or right mouse button.

Syntax

```
ViewONE.setViewMenus(Boolean option)
```

Call this method with a value of true to make the menus available. By default, the menus are available.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setViewMenus(true);
```

Print options

You can use the following APIs to modify and interact with the printing options.

isPrintDialog method

Returns a Boolean value of true if the print dialog box is enabled, false otherwise.

Syntax

```
var dialog = ViewONE.isPrintDialog();
```


Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

printDocument method

Displays a print dialog box from which the user can print the current document.

Syntax

This method is available only for multiple page documents.

```
ViewONE.printDocument();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

printPage method

Displays a print dialog box from which the user can print the current page.

Syntax

```
ViewONE.printPage();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

printRange method

Displays a range dialog box from which the user can print a range of pages in a document.

Syntax

This method is available only for multiple page documents.

```
ViewONE.printRange();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

printSelected method

Produces a print dialog box from which the user can print pages that were selected by using the page-select menu.

Syntax

This method is available only for multiple page documents.

```
ViewONE.printSelected();
```

This method can be used with the `selectPage(int pageNumber)` method and `clearSelections()` method to print any page or group of pages within a document.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“`selectPage` method” on page 266

In a multiple page document, this method toggles the select property on the page in a document that is indicated by either the **pageNumber** parameter or `getPage()` method.

“`clearSelections` method” on page 265

Clears all page selections in the document.

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

printTransformed method

Generates a print with the rotate, invert, and flip modes that are applied to the image.

Syntax

```
ViewONE.printTransformed();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

printVisible method

Displays a print dialog box from which the user can print the visible area of the currently displayed image.

Syntax

```
ViewONE.printVisible();
```

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setPrintAutoRotate method

Sets whether IBM Daeja ViewONE rotates images before printing.

Syntax

```
ViewONE.setPrintAutoRotate(Boolean option)
```

If this method is set to true, IBM Daeja ViewONE attempts to rotate images automatically before printing. Rotating the page allows as many images as possible to be printed vertically on the page. By default, the viewer does not attempt to auto-rotate images.

This method is available only if the **multiPrintNum** parameter is set to a value greater than 1.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrintAutoRotate(true)
```

Related reference:

“**multiPrintNum** parameter” on page 211

This parameter allows multiple images to be printed on a single printed page of a multipage document. This feature is useful where images are wide and narrow (for example, checks or similar images).

setPrintCopies method

Sets the number of copies that are printed when the user prints a single page, multiple pages, or an entire document.

Syntax

This method is available only when the print dialog box is disabled

```
ViewONE.setPrintCopies(integer)
```

You can disable the print dialog box by calling the `setPrintDialog(false)` method with a value of `false` or by setting the equivalent HTML parameter **printDialog** to `false`.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrintCopies(2);
```

Related reference:

“**setPrintDialog** method”

If set to `false`, prints without displaying the standard print dialog box.

“**printDialog** parameter” on page 215

This parameter determines whether the standard print dialog box is displayed.

setPrintDialog method

If set to `false`, prints without displaying the standard print dialog box.

Syntax

```
ViewONE.setPrintDialog(Boolean option)
```

If the user did not previously print with the ViewONE Print Accelerator Module, the user's default printer is used. Otherwise, the last printer that was used with the ViewONE Print Accelerator Module is used.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrintDialog(false);
```

setPrinter method

Sets the default printer for IBM Daeja ViewONE to the specified printer.

Syntax

```
ViewONE.setPrinter(String)
```

The parameter must be the printer name or the unique part of the printer name that is specified in the user's printer settings. The default settings for the printer, such as orientation and rotation, are used.

This method is available only when the print dialog box is disabled.

You can disable the print dialog box by calling the `setPrintDialog(true/false)` method with a value `false` or by setting the equivalent HTML parameter `printDialog` to `false`.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrinter("myprinter");
```

Related reference:

"setPrintDialog method" on page 358

If set to `false`, prints without displaying the standard print dialog box.

"printDialog parameter" on page 215

This parameter determines whether the standard print dialog box is displayed.

setPrintHeader method

Sets the custom text that is to appear at the top of each printed page.

Syntax

```
ViewONE.setPrintHeader(String headerString)
```

The following table lists the values that you can use for the `headerString` parameter.

| Value | Description |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| false | Disables the print headers. |
| any text | Specifies custom text to appear at the top of each printed page. For example, you might specify your copyright statement for the documents that are being viewed. |

| Value | Description |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| formatted text | <p>A text string that includes the following formatting elements in the header:</p> <p>\$page Inserts the word <i>page</i>, which is translated as needed.</p> <p>\$of Inserts the word <i>of</i>, which is translated as needed.</p> <p>\$pages Inserts the word <i>pages</i>, which is translated as needed.</p> <p># Inserts the number of the page that is being printed.</p> <p>## Inserts the total number of pages of the document.</p> <p>For example, you might specify the following text:</p> <p>\$page # \$of ## © Copyright My Company 2014"</p> |

By default, the header text is set to "(\$page # \$of ##)".

Print headers are available for internet Explorer 4.01 or later, Netscape 4.06 or later, and the Java plug-in 1.3. (The Java plug-in 1.2.2 does not support print headers.)

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrintHeader("$page # $of ##")
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setPrintJobName method

Sets the name of the print job that is to be sent to the printer.

Syntax

```
ViewONE.setPrintJobName(String)
```

Note: This method is available only if you are using either the ViewONE Print Accelerator Module or the Oracle Java plug-in, v1.2 or later.

Requirements

- IBM Daeja ViewONE Standard, V3.1.144 or later
- IBM Daeja ViewONE Professional, V1.1.144 or later
- ViewONE Print Accelerator Module

Example

```
ViewONE.setPrintJobName("my print job")
```

Streamer methods

Use the streamer methods to enable, disable, or set the location of the streamer.

setStreamerEnabled method

Enables and disables the streamer dynamically.

Syntax

```
ViewONE.setStreamerEnabled(Boolean option)
```

The `setStreamerEnabled(true/false)` method is the JavaScript equivalent of the **StreamerEnabled** HTML parameter.

Requirements

- IBM Daeja ViewONE Professional, V1.1.184 or later
- ViewONE Document Streaming Server Module

Example

```
ViewONE.setStreamerEnabled(true);
```

Related reference:

“**streamerEnabled** parameter” on page 257

This parameter determines whether HTTP requests for documents and images are directed to the ViewONE Document Streaming Server Module.

setStreamerURL method

Sets the streamer location dynamically to the specified URL.

Syntax

```
ViewONE.setStreamerURL(String url)
```

The `setStreamerURL` method is the JavaScript equivalent of the **StreamerURL** HTML parameter.

Important: When you open a document by using JavaScript, you must call the `setStreamerURL` method before each call to open a document.

Requirements

- IBM Daeja ViewONE Professional, V1.1.184 or later
- ViewONE Document Streaming Server Module

Example

```
ViewONE.setStreamerURL("http://myStreamerHost/Streamer");
```

Related reference:

“**streamerURL** parameter” on page 258

This parameter specifies the URL for the installed ViewONE Document Streaming Server Module.

Timeout and the user idle control

Use the following methods to configure the timeout and idle values.

isTimedOut method

Returns a value of true if the applet timed out because the user did not use the applet in the time that is specified. The time is specified by the `setTimeout(seconds)` method or the **timeout** HTML parameter.

Syntax

Otherwise, this method returns false.

```
var timedout = ViewONE.isTimedOut();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“setTimeout method” on page 363

Sets and starts a usage timer.

“timeOut parameter” on page 205

This parameter sets and starts a usage timer.

getTimeLeft method

Returns the remaining time in seconds before the applet times out if the timeout was initially set by using the setTimeout method or the **timeOut** parameter.

Syntax

```
var timeleft = ViewONE.getTimeLeft();
```

If the timeout was not set in this way, the method returns 0.

The remaining time is automatically reset each time that the user performs an action such as scrolling, changing pages, or zooming.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

getTimeout method

Returns the timeout value that was set by using either the setTimeout(seconds) method or the **timeOut** HTML parameter.

Syntax

```
var seconds = ViewONE.getTimeout();
```

The returned value is an integer that represents the timeout in seconds.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“setTimeout method” on page 363

Sets and starts a usage timer.

“timeOut parameter” on page 205

This parameter sets and starts a usage timer.

setServerEventHandler method

Sets the server event handler URL to the URL that is specified by the `url` parameter.

Syntax

```
ViewONE.setServerEventHandler(String url)
```

For more information, see the **ServerEventHandler** parameter in the *IBM Daeja ViewONE Parameters Reference Manual*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setServerEventHandler("http://www.Mysite.com/MyServerLogging.aspx");
```

Related reference:

“**serverEventHandler** parameter” on page 177

This parameter determines whether event handling is enabled for use with the JavaScript API.

setEventHandlerResponse method

Provides the user with an alternative URL for the current document or page when event 44 or event 45 occurs.

Syntax

For information about the event handler and events 44 and 45, see *Events IDs and descriptions*.

```
ViewONE.setEventHandlerResponse(String text)
```

Requirements

- IBM Daeja ViewONE Standard, V3.0.610 or later
- IBM Daeja ViewONE Professional, V1.0.610 or later
- No additional modules

Example

```
ViewONE.setEventHandlerResponse("url:http://www.daeja.com/test.tif");
```

Related reference:

“Events IDs and descriptions” on page 316

IBM Daeja ViewONE can signal many different events to the event handler.

setTimeout method

Sets and starts a usage timer.

Syntax

```
ViewONE.setTimeout(int seconds)
```

If the user does not use the applet for the specified number of seconds, the applet is automatically disabled.

The applet can be re-enabled in the following ways:

- By calling one of the timeout JavaScript methods
- By calling a JavaScript open method to open a document
- By revisiting the page that contains the applet in Netscape
- By reloading the page in internet Explorer

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setTimeout(30);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

stopTimeout method

Disables the timer that was set by using either the `setTimeout(seconds)` method or the **timeout** HTML parameter.

Syntax

```
ViewONE.stopTimeout();
```

If the applet is timed out, this method re-enables or wakes the applet.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“`setTimeout` method” on page 363

Sets and starts a usage timer.

“**timeOut** parameter” on page 205

This parameter sets and starts a usage timer.

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

wakeup() method

Wakes the applet after the applet times out.

Syntax

The user then is able to use the applet as normal.

```
ViewONE.wakeup();
```

For information about setting the timeout value, see the `setTimeout(seconds)` method.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Related reference:

“`setTimeout` method” on page 363
Sets and starts a usage timer.

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

Toolbars and buttons

Use the following APIs to display toolbars and buttons.

isAdjustToolVisible method

Returns a Boolean value of true if the adjust tool for brightness, contrast, and luminance is visible.

Syntax

```
var onOff = ViewONE.isAdjustToolVisible();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isAllButtons method

Returns a Boolean value of true if the file, print, image, new-window, and view buttons are visible.

Syntax

```
var buttonsVisible = ViewONE.isAllButtons();
```

Otherwise, this method returns false.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isDocButtons method

This method returns true if the **Doc<N>** set of buttons are visible; false otherwise.

Syntax

```
ViewONE.isDocButtons();
```

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- IBM Daeja ViewONE Virtual, V4.1.0 or later

isFileButtons method

Returns a Boolean value of true if the file buttons are visible.

Syntax

Otherwise, this method returns false.

```
var buttonsVisible = ViewONE.isFileButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isImageButtons method

Returns a Boolean value of true if the image buttons are visible.

Syntax

Otherwise, this method returns false.

```
var buttonsVisible = ViewONE.isImageButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isInvertButtons method

Returns a Boolean value of true if the invert button is visible.

Syntax

Otherwise, this method returns false.

```
var buttonVisible = ViewONE.isInvertButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isNewWindowButtons method

Returns a Boolean value of true if the **New Window** button is visible.

Syntax

Otherwise, this method returns false.

```
var buttonVisible = ViewONE.isNewWindowButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isPageButtons method

Returns a Boolean value of true if the page buttons are visible.

Syntax

Otherwise, this method returns false.

```
var buttonsVisible = ViewONE.isPageButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isPrintButtons method

Returns a Boolean value of true if the print button is visible.

Syntax

Otherwise, this method returns false.

```
var buttonVisible = ViewONE.isPrintButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isScrollbars method

Returns a Boolean value of true if scrollbars are enabled.

Syntax

Otherwise, this method returns false.

```
var scrollbars = ViewONE.isScrollbars();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isStatusBar method

Returns a Boolean value of true if the status bar is visible.

Syntax

Otherwise, this method returns false.

```
var statusBarVisible = ViewONE.isStatusBar();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isViewButtons method

Returns a Boolean value of true if the view buttons are visible.

Syntax

Otherwise, this method returns false.

```
var buttonsVisible = ViewONE.isViewButtons();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

isZoom100Button method

This method returns true if the **zoom100** button is visible; false otherwise.

Syntax

```
ViewONE.isZoom100Button();
```

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- IBM Daeja ViewONE Virtual, V4.1.0 or later

setAdjustToolVisible method

Sets the visibility of the adjust tool for brightness, contrast, and luminance.

Syntax

```
ViewONE.setAdjustToolVisible(Boolean OnOff)
```

A value of true sets the tool to be visible. By default, the tool is visible.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setAdjustToolVisible(true);
```

setAllButtons method

Specifies whether all buttons in the top and bottom toolbars are visible.

Syntax

These buttons include the file, print, zoom, transformation, and new window buttons in the top toolbar and the view button in the lower toolbar.

```
ViewONE.setAllButtons(Boolean option)
```

A value of true (default) sets the buttons to be visible.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setAllButtons(true);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setAppletEnabled method

Enables or disables the viewer user interface.

Syntax

```
ViewONE.setAppletEnabled(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, V3.1.172 or later
- IBM Daeja ViewONE Professional, V1.1.172 or later
- No additional modules

Example

```
ViewONE.setAppletEnabled(false);
```

setDocButtons method

Calling this method with a value of false hides the **doc<N>** set of buttons that you use to browse between documents.

Syntax

```
ViewONE.setDocButtons(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- IBM Daeja ViewONE Virtual, V4.1.0 or later

Example

```
ViewONE.setDocButtons(false);
```

setFileButtons method

Specifies whether the toolbar includes file buttons.

Syntax

```
ViewONE.setFileButtons(Boolean option)
```

If this method is set to true, the following buttons are included in the toolbar:

- **Open document**
- **Close document**
- **Save document**

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setFileButtons(true);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setImageButtons method

Specifies whether the toolbar includes image buttons.

Syntax

```
ViewONE.setImageButtons(Boolean option)
```

If this method is set to true, the following buttons are included in the toolbar:

- **Zoom area**
- **Zoom in**

- Zoom out
- Fit to width
- Fit to height
- Best fit
- Rotate clockwise
- Rotate counterclockwise
- Rotate 180 degrees

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setImageButtons(true);
```

Related reference:

“**JavaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setImageToolBarCollapsed method

Puts the image toolbar into an expanded or collapsed state.

Syntax

```
ViewONE.setImageToolBarCollapsed(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setImageToolBarCollapsed(false);
```

setInvertButtons method

Specifies whether the toolbar includes an invert button.

Syntax

```
ViewONE.setInvertButtons(Boolean option)
```

A value of true (default) includes the button in the toolbar.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version

- No additional modules

Example

```
ViewONE.setInvertButtons(true);
```

setNewWindowButtons method

Specifies whether the toolbar includes a new-window button.

Syntax

```
ViewONE.setNewWindowButtons(Boolean option)
```

A value of true (default) includes the button in the toolbar.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setNewWindowButtons(true);
```

setPageButtons method

Specifies whether the toolbar includes page navigation buttons.

Syntax

```
ViewONE.setPageButtons(Boolean option)
```

If this method is set to true (default), the toolbar includes the following buttons:

- First Page
- Previous Page
- Next Page
- Last Page

The toolbar also includes another page selection option:

- A scroll bar for page selection (IBM Daeja ViewONE Standard and IBM Daeja ViewONE Professional)
- A text box for page selection (IBM Daeja ViewONE Virtual)

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setPageButtons(true);
```

setPrintButtons method

Specifies whether the toolbar includes a print button.

Syntax

```
ViewONE.setPrintButtons(Boolean option)
```

A value of true includes the button in the toolbar.

Security note: This method is disabled by default unless the **JavaScriptExtensions** parameter is set to true.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setPrintButtons(true);
```

Related reference:

“**javaScriptExtensions** parameter” on page 175

This parameter enables certain IBM Daeja ViewONE methods. These methods are disabled by default to prevent unauthorized users from using the methods to access secure information, documents, or annotations in IBM Daeja ViewONE

setScrollbars method

Specifies whether scrollbars are displayed when the image is scaled to a size larger than the display area.

Syntax

```
ViewONE.setScrollbars(Boolean option)
```

A value of true (default) indicates that the scrollbars are to be displayed. A change in this setting is visible after the next refresh of the display, for example, when a page is zoomed or a page is changed.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setScrollbars(true);
```

setStatusBar method

Specifies whether the status bar is visible.

Syntax

```
ViewONE.setStatusBar(Boolean option)
```

A value of true (default) makes the status bar visible.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
ViewONE.setStatusBar(true);
```

setViewButtons method

Specifies whether the toolbar includes view buttons.

Syntax

```
ViewONE.setViewButtons(Boolean option)
```

A value of true (default) includes the following buttons in the toolbar.

- Fullpage
- Thumbnails
- Two-page
-

Note: Two-page is not available in IBM Daeja ViewONE Virtual

- Thumbs-left
- Thumbs-bottom
- Thumbs-right
- Thumbs-top

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setViewButtons(true);
```

setViewToolBarCollapsed method

Puts the view toolbar into an expanded or collapsed (folds) state.

Syntax

```
ViewONE.setViewToolBarCollapsed(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, V3.1.184 or later
- IBM Daeja ViewONE Professional, V1.1.184 or later
- No additional modules

Example

```
ViewONE.setViewToolBarCollapsed(false);
```

setZoom100Button method

Calling this method with a value of false hides the **zoom100** button that lets the user display the image at 100% of its resolution. This parameter applies to all documents.

Syntax

```
ViewONE.setZoom100Button(Boolean option)
```

Requirements

- IBM Daeja ViewONE Standard, V4.1.2 or later
- IBM Daeja ViewONE Professional, V4.1.2 or later
- IBM Daeja ViewONE Virtual, V4.1.0 or later

Example

```
ViewONE.setZoom100Button(false);
```

toggleAdjustTool method

Toggles the visibility of the adjust tool for brightness, contrast, and luminance.

Syntax

```
ViewONE.toggleAdjustTool();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Viewing options

You can use the following methods to configure viewing tools and perform viewing operations.

getMagFactor method

Returns an integer that represents the current magnification factor for the magnifier window.

Syntax

```
ViewONE.getMagFactor()
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
var factor = ViewONE.getMagFactor();
```

getView method

Returns an integer value that represents the view mode.

Syntax

```
var scale = ViewONE.getView();
```

The following table lists the values for the view mode.

| Value | Description |
|-------|--------------------------------------------------------------------------|
| 0 | Fullpage (default): A single view of the current page is visible. |

| Value | Description |
|-------|--------------------------------------------------------------------------------------------|
| 1 | Twopage: Two pages are visible at the same time. |
| 2 | Thumbsonly: Only a view of the thumbnails is visible. |
| 3 | Thumbsleft: A view of the current page with thumbnails on the left of the page. |
| 4 | Thumbsright: A view of the current page with thumbnails on the right of the page. |
| 5 | Thumbstop: A view of the current page with thumbnails at the top of the page. |
| 6 | Thumbsbottom: A view of the current page with thumbnails at the bottom of the page. |

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.1 or later
- No additional modules

getViewerWindow method

Gets the viewer window in which a command that is associated with a custom JavaScript button is to be run.

Syntax

```
ViewONE.getViewerWindow(String windowInstanceId)
```

The **windowInstanceId** is a string that represents the identifier of the current viewer window in which the user clicked the custom. You can obtain this identifier by calling the `getViewerWindowInstanceId()` method.

Requirements

- IBM Daeja ViewONE Standard, V4.1.1 or later
- IBM Daeja ViewONE Professional, V4.1.1 or later
- No additional modules

Example

The following example shows how the `getViewerWindow(windowInstanceId)` method and `getViewerWindowInstanceId()` method are used for a custom button that runs the `nextPage` command:

```
function customButton() {
    var v1Instance = ViewONE.getViewerWindowInstanceId();
    ViewONE.getViewerWindow(v1Instance).nextPage();
}
```

Related reference:

“`getViewerWindowInstanceId` method”

Returns the identifier of the current viewer window instance as a string.

getViewerWindowInstanceId method

Returns the identifier of the current viewer window instance as a string.

Syntax

```
var v1Instance = ViewONE.getViewerWindowInstanceId();
```

You can call this method to identify the window in which a user clicked a custom JavaScript button. You then call the `getViewerWindow()` method on the JavaScript reference to run the associated command in that viewer window.

Requirements

- IBM Daeja ViewONE Standard, V4.1.1 or later
- IBM Daeja ViewONE Professional, V4.1.1 or later
- No additional modules

Related reference:

“`getViewerWindow` method” on page 376

Gets the viewer window in which a command that is associated with a custom JavaScript button is to be run.

initializeLabels, setLabel, useLabels, and clearLabels method group

The label method group configures and display labels.

Syntax

```
ViewONE.initalizeLabels(int numLabels)
ViewONE.setLabel(String pageLabel, String pageLabelColor, String thumbLabel,
    String thumbLabelColor, int LabelNum)
ViewONE.useLabels()
ViewONE.clearLabels()
```

The `initializeLabels` method specifies the number of files that are to be labeled, where each file represents a successive page in a document. The `setLabel` method is then called to specify the label that is to be displayed for each file in the full-page area and on the corresponding thumbnail. Finally, the `useLabels` method is called to display the labels. The labels remain visible until the document is closed or the `clearLabels` method is called.

You can specify the label color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*. Where no color is specified (that is, null), the default color is white.

The label array begins at array element zero.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Label method examples

If you do not want to define a label for either the full-page area or a thumbnail, specify the label as a null string, for example:

```
ViewONE.setLabel(null, null, "thumb 1", null, 0);
```

This example sets up a label for the thumbnail for the first page.

```
ViewONE.initializeLabels(3);
ViewONE.setLabel("Page label 1", null, "thumb 1", null, 0);
ViewONE.setLabel("Page label 2", null, "thumb 2", null, 1);
ViewONE.setLabel("Page label 3", "223,223,255", "thumb 3", "255,223,223", 2);
ViewONE.useLabels();
ViewONE.openFile("mydocument.tif", 1);
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

isAreaZoom method

Returns a Boolean value that indicates whether the zoom-area mode is turned on.

Syntax

```
var areaZoom = ViewONE.isAreaZoom();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

isMagnifier method

Returns a Boolean value that indicates whether the magnifier is visible.

Syntax

```
var magVisible = ViewONE.isMagnifier();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

isNewWindowVisible method

Returns a Boolean value of true if the new IBM Daeja ViewONE window is visible.

Syntax

```
var visible = ViewONE.isNewWindowVisible();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

setAreaZoom method

Sets whether the zoom-area mode is initiated.

Syntax

```
ViewONE.setAreaZoom(Boolean option)
```

If the method is called with its parameter's value set to true, the zoom-area mode is initiated. The mouse pointer changes to a cross. The user can right-click and drag the pointer to select the area for zooming. When the user releases the mouse

button, the area that is selected is zoomed as large as possible while the image aspect is maintained within the available window area.

If the selected area is not greater than the zoom trigger size, 20 by 20 pixels, zooming does not occur. Thus, the user can release the mouse button if the mode was initiated accidentally.

If this method is called with its parameter's value set to false, the mouse returns to drag mode to pan the image.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setAreaZoom(true);
```

setImageBackColor method

Sets the default color that is used for the background of monochrome images.

Syntax

```
ViewONE.setImageBackColor( RGB color)
```

The **color** parameter is a string that defines the color. You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setImageBackColor("255, 255, 255")
```

Related reference:

“IBM Daeja ViewONE color scheme text” on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

setImageForeColor method

Sets the default color that is used for the foreground (text) in monochrome images.

Syntax

```
ViewONE.setImageForeColor( RGB color)
```

The **color** parameter is a string that defines the color. You can specify the color by using a comma-delimited RGB value or by specifying one of the values from *IBM Daeja ViewONE color scheme text*.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- No additional modules

Example

```
ViewONE.setImageForeColor("0, 0, 0")
```

Related reference:

"IBM Daeja ViewONE color scheme text" on page 262

For parameters that require a color value, you can use either a comma-delimited RGB value or a color name as defined by the IBM Daeja ViewONE color scheme.

setMagBounds method

Sets the window position and size of the external magnifier.

Syntax

```
ViewONE.setMagBounds(int x, int y, int width, int height)
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setMagBounds(10, 10, 300, 300);
```

setMagFactor method

Sets the magnification factor for the magnifier window.

Syntax

```
ViewONE.setMagFactor(int factor)
```

The **factor** parameter is an integer value 2 - 25.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
ViewONE.setMagFactor(5);
```

setMagnifierinternal method

Sets whether to display a magnifier window that is internal to the display area.

Syntax

```
ViewONE.setMagnifierinternal(Boolean option)
```

Set this method to true to display the magnifier window. Set this method to false to hide the magnifier window.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version

- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Example

```
ViewONE.setMagnifierinternal(true);
```

setMagnifier method

Sets whether to display a magnifier window that is external to the display area.

Syntax

```
ViewONE.setMagnifier(Boolean option)
```

Call this method with a value of true to display the magnifier window. Call this method with a value of false to hide the magnifier window.

If the parameter of this method is set to true, a rectangle is visible around the mouse pointer to highlight the area that is being magnified.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Example

```
ViewONE.setMagnifier(true);
```

setNewWindowVisible method

Specifies whether to make a new IBM Daeja ViewONE window visible.

Syntax

```
ViewOne.setNewWindowVisible(Boolean option)
```

Calling this method with a value of true makes the new window visible; false hides the new window.

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

Examples

```
ViewONE.setNewWindowVisible(true);
```

```
ViewONE.setNewWindowVisible(!ViewONE.isNewWindowVisible());
```

setView method

Specifies the view mode that is to be used to display the pages of a document.

Syntax

This method is effective only while a document is open.

```
ViewONE.setView(int view)
```

The following table lists the values for the **view** parameter.

| Value | Description |
|-------|------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | Fullpage (default): A single view of the current page is visible. |
| 1 | Twopage : Two pages are visible at the same time.

Note : This option is not available in IBM Daeja ViewONE Virtual. |
| 2 | Thumbsonly : Only a view of the thumbnails is visible. |
| 3 | Thumbsleft : A view of the current page with thumbnails on the left of the page. |
| 4 | Thumbsright : A view of the current page with thumbnails on the right of the page. |
| 5 | Thumbstop : A view of the current page with thumbnails at the top of the page. |
| 6 | Thumbsbottom : A view of the current page with thumbnails at the bottom of the page. |

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, any version
- No additional modules

Example

```
ViewONE.setView(0);
```

showImageBackColorDialog method

Displays a dialog box in which the user can change the default color that is used for the background of monochrome images.

Syntax

```
ViewONE.showImageBackColorDialog();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

showImageForeColorDialog method

Displays a dialog box in which the user can change the default color that is used for the foreground (text) of monochrome images.

Syntax

```
ViewONE.showImageForeColorDialog();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

toggleAreaZoom method

Toggles the zoom-area mode on and off.

When the zoom-area mode is on, the cursor, which changes to a cross hair, is used to select an area to zoom. When the zoom-area mode is off, the cursor is used to drag or scroll the image.

Syntax

```
ViewONE.toggleAreaZoom();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- No additional modules

toggleMagnifier method

Toggles the magnifier on and off.

Syntax

```
ViewONE.toggleMagnifier();
```

Requirements

- IBM Daeja ViewONE Standard, any version
- IBM Daeja ViewONE Professional, any version
- IBM Daeja ViewONE Virtual, V4.1.2 or later
- No additional modules

Annotating and redacting documents

Depending on your site configuration, you can add text or image annotations and redact documents.

Annotating documents

You can add text and image annotations to a document, such as a sticky note, a comment, or a received date. The available annotation types depend on your site configuration. Many of the annotations support optional hyperlinks (hotspots), and you can customize the line size, color, and fill color.

Depending on your site configuration and security, the annotations can be viewed and edited by other users. The annotation data is stored separately from the document data.

You might need appropriate permissions to add or edit annotations.

Restriction: The user interface might allow the use of tabbed characters in annotations, for example, sticky notes, but tabbed characters are not supported.

Redacting documents

Depending on your site configuration, you can redact documents. For example, you can draw a dark rectangle to cover sensitive or confidential information in a document to prevent others from reading the content.

Your site administrator determines whether the redactions can be permanently added to the underlying document by burning the annotation into the file. Burning the annotation creates a new file with the redaction that you can safely distribute to others. For example, you can email the file to someone who does not have access to IBM Daeja ViewONE.

Important: If annotation burning is not enabled, then the redaction information is stored separately from the underlying document. Depending on your site's security model, another user might be able to view or print the document without the redaction.

Editing annotations

You might need appropriate permissions to edit annotations.

Procedure

To edit an annotation:

1. Click the **Select** icon in the Annotation toolbar and click the annotation. The annotation is highlighted and the Edit Annotation toolbar is displayed above the annotation.
2. Use the icons to edit the annotation properties. For example, change the line width or add a hyperlink.
3. If needed, drag the highlighted rectangles at the edges of the annotation to resize the annotation.


4. Click **Fix Annotation** in the Edit Annotation toolbar to save the changes.
5. Click **Save annotation changes** on the annotation toolbar to save your annotations.

Tips for using the viewer


To manage documents in the viewer, you can use the buttons or the pop-up menu. The specific actions that are available to you depend on the configuration of the viewer for your application. You can also right-click some buttons to access more actions.

Opening documents

When you use the viewer on a website, you typically do not need to select a file because a document is loaded automatically when the web page is loaded. If you


need to open a file, click the **Open document** button  to display a dialog box from which you can select an image or document file. Alternatively, right-click the **Open document** button to enter the URL to an image file or document file.

Saving documents


If the current document is a single file, click the **Save document** button  to save the file to your local disk. If the document is a multipage TIFF file, right-click the **Save document** button to save the current page as a TIFF file on your local disk.

You cannot save a multipage document in which there is a separate file for each page to your local disk.

Enlarging the view

Click the **magnifier** button  to use a movable magnifier to enlarge part of the viewer. To display the magnifier in a separate window, right-click the **magnifier** button.

If you are using the viewer in a website that cannot resize the display area or the

display area is too small, click the **New window** button . This button opens the viewer in a separate window that includes all the typical buttons and menus.

Rotating documents

Click the following buttons to rotate the current page:

- **Rotate clockwise** 

- Rotate counterclockwise






- Rotate 180 degrees



Right-click these buttons to rotate all the pages in a multipage document.

Viewing thumbnails

Click the following buttons to display or hide thumbnails for the current document:

| Button | Name |
|-------------------------------------------------------------------------------------|-------------------------|
|  | View page |
|  | View thumbnails |
|  | View two pages |
|  | View thumbnails: Left |
|  | View thumbnails: Bottom |
|  | View thumbnails: Right |
|  | View thumbnails: Top |

You can right-click any of these buttons except **View page** to display only the pages that were selected by using the **Selection** menu or by clicking the S key.

Troubleshooting and support

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the IBM Support representative know where to start to find the cause of the problem. This step includes asking yourself basic questions:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, which can then lead you to a problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is “What is the problem?” This question might seem straightforward; however, you can break it down into several focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to consider when you are investigating problems.

The following questions help you to focus on where the problem occurs to isolate the problem layer:

- Is the problem specific to one platform or operating system, or is it common across multiple platforms or operating systems?
- Is the current environment and configuration supported?

If one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many problems can be

traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily develop a timeline by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log.

To develop a detailed timeline of events, answer these questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can give you a frame of reference in which to investigate the problem.

Under which conditions does the problem occur?

Knowing which systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

Can the problem be reproduced?

From a troubleshooting standpoint, the ideal problem is one that can be reproduced. Typically, when a problem can be reproduced you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve. However, problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur. If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

- Can the problem be re-created on a test system?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, or a particular application?

Getting fixes from Fix Central

For products that are available on premise, you can use Fix Central to find the fixes that are recommended by IBM Support for a variety of products. With Fix Central, you can search, select, order, and download fixes for your system with a choice of delivery options.

Procedure

To find and install fixes:

1. Go to Fix Central (www.ibm.com/support/fixcentral) to download the installation program for product fixes. Fix Central provides download, installation, and configuration instructions for the installation program.
2. From the **Select product** tab, select Enterprise Content Management as your product group, then select your product, version and platform, and click **Continue**.
3. Browse or search for fixes based on criteria such as (Authorized Program Analysis Report) APAR number or fix ID, or just request a list of recommended fixes.
4. If you have already registered for My support, sign in and skip to the next step. If you have not registered, click **Register now**. Complete the registration form using your email address as your IBM ID and click **Submit**.
5. Select and download the fix by using your preferred download method. When downloading the file, ensure that the name of the maintenance file is not changed. This change might be intentional, or it might be an inadvertent change that is caused by certain web browsers or download utilities.
6. Apply the fix.
7. Optional: Subscribe to receive weekly email notifications about fixes and other IBM Support updates.

Contacting IBM Software Support

IBM Software Support provides assistance with product defects, answering FAQs, and performing rediscovery.

About this task

For products that are available on premise, see the IBM Support Portal (www.ibm.com/support) to find fixes, product documentation, support contacts, and more.

Also, stay current with fixes, flashes, and other updates by getting daily or weekly announcements through email, custom web pages, and RSS feeds. Sign up on the My Notifications page (www.ibm.com/software/support/einfo.html).

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Software Support. Before you contact IBM Software Support, your company must have an active IBM software subscription and support contract, and you must be authorized to submit problems to IBM. The type of software subscription and support contract that you need depends on the type of product you have. For information about the types of available support, see the Support portfolio topic in the Software Support Handbook.

Procedure

Complete the following steps to contact IBM Software Support with a problem:

1. Define the problem, gather background information, and determine the severity of the problem. To determine the severity level, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

| Severity | Business impact |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution. |
| 2 | Significant business impact: The program is usable but is severely limited. |
| 3 | Some business impact: The program is usable with less significant features (not critical to operations) unavailable. |
| 4 | Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented. |

For more information, including examples of each severity level, see the "Getting IBM support" topic in the *IBM Software Support Handbook*.

2. Gather diagnostic information. In the IBM Support site, search for the keyword `mustgather`, along with your product and version name, to see whether there are specific files or other diagnostic information to gather, for example:
 - What software versions were you running when the problem occurred?
 - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
 - Can the problem be re-created? If so, what steps led to the failure?
 - Have any changes been made to the system (for example, hardware, operating system, networking software, and so on)?
 - Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.
3. Submit the problem to IBM Support.

IBM Support Assistant (ISA)

Go to the Software support site to use the IBM Support Assistant (www.ibm.com/software/support/isa/).

Online

Go to the IBM Support Portal (www.ibm.com/software/support/). You can open, update, and view all your Service Requests from the Service Request portlet on the Service Request page.

Phone For the phone number to call in your country, see the Directory of worldwide contacts web page (www.ibm.com/planetwide/).

Results

If the problem that you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily so that other users who experience the same problem can benefit

from the same resolution.

Notices

This information was developed for products and services offered in the U.S.A.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (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 Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: <http://www.ibm.com/legal/copytrade.shtml>

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

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.

Other company, product, and service names may be trademarks or service marks of others.

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 IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.

Glossary

This glossary includes terms and definitions for IBM Daeja ViewONE.

The following cross-references are used in this glossary:

- See refers you from a term to a preferred synonym, or from an acronym or abbreviation to the defined full form.
- See also refers you to a related or contrasting term.

To view glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology (opens in new window).

"A" "D" "E" F "I" "L" on page 400 M"N" on page 400 "P" on page 400 "R" on page 400S "T" on page 400 "V" on page 400 "W" on page 400 "Z" on page 401

A

angle

An annotation type that is composed of two lines and the calculated angle between them.

annotation

An added descriptive comment, note, line, or shape.

annotation data

A representation of an annotation in a form suitable for communication, interpretation, or processing by human or automatic means. The annotation data is stored separately from the document data.

annotation security

One or more settings that define how users can interact with annotations.

applet A program that performs a specific task and is typically portable between operating systems. Often written in Java, applets can be downloaded from the Internet and run in a web browser.

D

document

A machine-readable collection of one or more objects that represent a composition, a work, or a collection of data.

E

Enterprise Content Management

The strategies, methods, and tools that are used to capture, manage, store, preserve, and deliver content and documents that are related to organizational processes. Enterprise Content Manager tools and strategies allow the management of an organization's unstructured information, wherever that information exists.

event An occurrence of significance to a task or system. Events can include completion or failure of an operation, a user action, or the change in state of a process.

event handler

Software that initiates an event response that is configured by the administrator.

F

find result

The results in the search view of the viewer thumbnail pane and in the document-viewing area.

I

image An electronic representation of an original document or picture that is produced by a scanning device or created from software.

image data

A pattern of bits with 0 and 1 values that define the pixels in an image. A 1 bit is a toned pixel.

image stamp

An overlaid image annotation.

K

key A column or an ordered collection of columns that is identified in the description of a table, index, or referential constraint. The same column can be part of more than one key.

L

luminance A simulated light source that shines on an image to make color images clearer to read.

M

module A piece of software that is provided to enable the viewing of a particular document type or to provide a specific piece of functionality; for example, redaction.

N

note annotation A text annotation type.

P

pan To move a display, such as a diagram or map, within a GUI window or pane. The purpose of panning is to control which portion of the diagram or map is visible.

parameter A value or reference that is passed to a function, command, or program that serves as input or controls actions. The value is supplied by a user or by another program or process.

property A characteristic of an object that describes the object. A property can be changed. Properties describe an object name, type, value, or behavior, among other things.

R

redact To block the viewing of the data that is displayed in the area under a redaction that is drawn on a document.

redaction An annotation type that is an opaque mask that is applied to a page so that a selected portion of the page is not visible.

rotation array Sets the rotation of each page of the current document, to the specified angle.

ruler A line annotation that displays a calculated length base on the image properties.

S

scale (noun) The number and progression of ticks along a vertical or horizontal axis.

scale (verb) To enlarge an image or marker.

streamer A module that provides individual pages of a document as required in response to user page selection in the document viewer.

T

token A syntactic element, such as a phrase, a word, or a set of one or more characters, that is used for analyzing and processing text.

V

value The content of a variable, parameter, special register, or field.

view A representation of a document, image, or annotation.

viewer A piece of client software that is used to produce a graphical display of document or image content that is stored in a file.

W

web application server The runtime environment for dynamic web applications. A Java EE web application server implements the services of the Java EE standard.

Z

zoom To progressively increase or decrease the size of a part of an image on a display or window.



Product Number: 5725-Q02
5725-Q03
5725-Q04

GC27-8105-01

