

Oracle® Documanage

Rules Reference for the Documanage Bridge

version 3.5

Part number: E14904-01

April 2009

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. Government Rights

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Trademarks

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

This documentation is in prerelease status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Publication History

First issue for version 3.0: February 2005

Revision 1 for version 3.1: September 2005

Revision 3 for version 3.1 Service Release 1: January 2006

Revision 4 for version 3.2 Service Release 1: May 2006

Revision 5 for version 3.3: September 2006

Revision 6 for version 3.4: September 2007

Revision 7 for version 3.4: February 2009

Contents

iii Publication History

Chapter 1, Introduction

- 2 Types of Rules
- 3 Attachment Variable Conventions
 - 3 Prefixes
 - 4 Document, Folder, and Project Specifications
 - 6 Standard Extended Project Properties
- 7 Rule Parameters
 - 7 RecordSet Wildcard
- 9 Server Rule Validations
- 10 Rule Descriptions

Chapter 2, Rules Reference

- 12 DmgBrcDumpFile
- 15 DmgBrcReceiveTempFile
- 17 DmgBrsAddDocumentAsRendition
- 19 DmgBrsAddNote
- 21 DmgBrsAdvanceProject
- 22 DmgBrsCacheAnnotationsFile
- 24 DmgBrsCacheContentsFile
- 27 DmgBrsCheckInDocument
- 29 DmgBrsCheckInProject
- 30 DmgBrsCheckOutDocument
- 34 DmgBrsCheckOutProject
- 35 DmgBrsCheckPausedServer
- 36 DmgBrsCopyAttachment
- 38 DmgBrsCopyDocument
- 40 DmgBrsCreateDiary
- 42 DmgBrsCreateDocument
- 47 DmgBrsCreateDocumentShortcut

49 DmgBrsCreateFolder
53 DmgBrsCreateProject
55 DmgBrsCreateProjectFromFolder
57 DmgBrsDeleteDiary
59 DmgBrsDeleteDocument
61 DmgBrsDeleteFolder
63 DmgBrsEncodeURLField
65 DmgBrsForwardProject
66 DmgBrsGetTaskQueue
70 DmgBrsGetTasks
72 DmgBrsIdle
74 DmgBrsInit
76 DmgBrsListCabinets
78 DmgBrsListCategories
79 DmgBrsListGroups
81 DmgBrsListUsers
83 DmgBrsListWorkflows
84 DmgBrsLockAnnotations
86 DmgBrsLogin
89 DmgBrsLogout
90 DmgBrsLogPropertySets
91 DmgBrsMarkNoteCurrent
93 DmgBrsMarkNoteDeleted
94 DmgBrsMoveDocument
96 DmgBrsPresentCabinetEx
99 DmgBrsPresentCategory
102 DmgBrsPresentCompoundDocument
108 DmgBrsPresentDocument
116 DmgBrsPresentDocumentPage
120 DmgBrsPresentFolder
126 DmgBrsPresentProject
133 DmgBrsPresentSubdocument
137 DmgBrsPresentWorkflow
139 DmgBrsQuery
153 DmgBrsQueryAllProjectsEx
157 DmgBrsQueryDiaries
160 DmgBrsQueryDocumentRenditions

163	DmgBrsQueryDocuments
174	DmgBrsQueryProjects
178	DmgBrsRemoveDocumentAsRendition
180	DmgBrsResumeProject
181	DmgBrsReturnAnnotations
183	DmgBrsReturnConvertedDocument
193	DmgBrsReturnDocument
195	DmgBrsReturnDocumentPage
198	DmgBrsReturnThumbnail
200	DmgBrsRunDocumangeDiagnostic
203	DmgBrsSetDocumentRetention
205	DmgBrsSetIndicators
207	DmgBrsSetupCompoundDocument
210	DmgBrsSetupFolder
212	DmgBrsSuspendProject
213	DmgBrsUndoDocumentCheckOut
215	DmgBrsUnlockAnnotations
217	DmgBrsUpdateAnnotations
219	DmgBrsUpdateDiary
221	DmgBrsUpdateDocument
225	DmgBrsUpdateDocumentCM
227	DmgBrsUpdateDocumentFile
230	DmgBrsUpdateFolder
232	DmgBrsUpdateProject
234	DmgBrsValidateSession
242	DmgBruCopyAttachmentVar
244	DmgBruLogAttachments
245	DmgBruPurgeFiles

Chapter 3, Quick Reference by Alphabet

Chapter 4, Quick Reference by Documange Object

256	Documents
258	Folders

259	Projects
260	Tasks
261	Notes
262	Diaries
263	Annotations
264	Documanage Diagnostic Tests
265	Session

Chapter 5, Exploration Rules

Chapter 6

Introduction

This topic describes each of the rules that are part of the Documanager Bridge.

All Documanager Bridge rules reside in a single library. The name of that library may be different on different platforms.

Platform	Library Name
Windows	dmg_brw32.dll
All other platforms	libdmg_br.so

Types of Rules

There are three types of rules in the Documanager Bridge:

- Server Rules run only on the bridge server (IDS). They may appear only on rules lists in docserv.ini.
- Client Rules run only on the standard IDS CGI client. They may appear only on rules lists in docclnt.ini.
- Utility Rules run on either IDS or the standard CGI client. They may appear on rules lists in either docserv.ini or docclnt.ini.

The three types of rules are distinguished by the first letters of their names.

Prefix	Type of rule	Example
DmgBrs	Server Rule	DmgBrsValidateSession
DmgBrc	Client Rule	DmgBrcDumpFile
DmgBru	Utility Rule	DmgBruLogAttachments

Unless otherwise noted, all Documanager Bridge rules do their work on the forward run of the rules list.

Attachment Variable Conventions

Unless otherwise noted, all Server Rules read their input attachment variables from the Output Queue and write their output attachment variables to the Output Queue. The rule DmgBrsCopyAttachment is provided for copying all attachment variables from the Input Queue to the Output Queue so the Server Rules can find them. Since DmgBrsCopyAttachment copies all attachment variables in this way, all input attachment variables also become output attachment variables unless otherwise noted.

Prefixes

In the Documange Bridge, there are some commonly used attachment variable prefixes that have conventional meanings. These are shown in the following table.

Prefix	Meaning	Example
DOC_	Relates to a Document	DOC_ID
FLD_	Relates to a Folder	FLD_KEYSTRING
NTE_	Relates to a Note	NTE_DESCRIPTION
PRJ_	Relates to a Project	PRJ_PRIORITY
X	Relates to an Extended Property, in combination with one of the other prefixes	XDOC_ denotes an Extended Document Property
TSK_	Relates to a Task	TSK_MANAGER
PS_	Relates to a Property Set	PS_LIST
DIA_	Relates to a Diary	DIA_TITLE
BRN_	Relates to a Project Branch	BRN_ID
CONTENTS_	Relates to a Document contents file or its Document.	CONTENTS_DECOMPRESSED_PATH
ANNOTS_	Relates to a Document annotations file or its Document.	ANNOTS_PATH

Document, Folder, and Project Specifications

In the description of the input attachment variable of many rules, the terms document specification, folder specification, or project specification are used. These terms are shorthand for a standard cluster of attachment variables that uniquely identify a document, folder, or project.

Current document specification

A current document specification refers to the current version of a document.

Variable	Description
CABINET	The name of the document's cabinet. Required.
DOC_ID	The ID property of the document. Required.

Version document specification

A version document specification refers to a particular version of a document. There are alternative ways to supply the version, described below

Variable	Description
CABINET	The name of the document's cabinet. Required.
DOC_ID	The ID property of the document. Required.
DOC_MAJORVERSION	The MajorVersion property of the document. Required if DOC_VERS is absent. This may be -1, to indicate the current version.
DOC_MINORVERSION	The MinorVersion property of the document. Required if DOC_VERS is absent, but ignored if DOC_MAJORVERSION is -1.
DOC_VERS	The MajorVersion and MinorVersion properties of the document expressed in the format M.m, where M is the MajorVersion and m is the MinorVersion separated by a dot. Required if DOC_MAJORVERSION or DOC_MINORVERSION is absent. M may be -1, to indicate the current version.

Folder specification

A folder specification refers to a specific folder. Either the KeyString property or the ID property of the folder must be supplied. If both are present, the ID is ignored.

Variable	Description
CABINET	The name of the folder's cabinet. Required.
FLD_LEVELNUMBER	The LevelNumber property of the folder in the cabinet. One (1) is the first level, that is, the level nearest the root. Optional, defaults to 1.
FLD_KEYSTRING	The KeyString property of the folder. Required if FLD_ID is absent.
FLD_ID	The ID property of the folder. Required if FLD_KEYSTRING is absent. Note that, in some Documange configurations, it may be possible for the ID property of the folders in a cabinet to be non-unique. In that case, use the KeyString in your bridge application. It is always unique.

Project specification A project specification refers to a specific project.

Variable	Description
----------	-------------

CABINET	The name of the project's cabinet. Required.
PRJ_KEYSTRING	The KeyString property of the project. Required.

DOCSPEC A DOCSPEC is a standardized output attachment variable that contains the elements of a version document specification formatted for use in a URL. This is provided as a convenience in the construction of URLs in a bridge client application. A DOCSPEC is in the following form:

CABINET=url_encoded_cabname&DOC_ID=ID&DOC_VERS=M.m

Parameter	Description
-----------	-------------

url_encoded_cabname	The name of the document's cabinet, encoded for use in a URL.
ID	The ID property of the document.
M.m	The MajorVersion and MinorVersion properties of the document separated by a period.

URL tail A URL tail is a standardized output attachment variable that contains a fragment suitable for use on the end of a URL that returns a file. This is provided as a convenience in the construction of URLs in a bridge client application. A URL tail is in the following form:

/url_encoded_document_name.ext

url_encoded_file_name is the Name property of a document—sometimes with an indication of a page number—with any forward slashes in the name replaced by underscores and encoded for use in a URL.

ext is a file extension, sometimes denoting the file's native file type and sometimes denoting a file type to which the file is to be converted.

URL tails are useful as the last element in URLs that return a file. These URLs will typically be for rules lists that contain rules with Return in their names, such as DmgBrsReturnDocument, DmgBrsReturnConvertedDocument, and so on. When such a URL is used with a browser, the URL tail provides a file name for display in any file saving dialog the browser may present.

Standard Extended Project Properties

Extended project properties may be classified into two groups: those that appear in all Documanager systems because they're used in managing workflow, and those that are unique to a particular Documanager system because they are derived from the LOB tables. In this document, call the former Standard Extended Project Properties, and the latter Extended Project Properties.

Attachment variables for extended project properties are named like other extended properties. They consist of a prefix of XPRJ_ followed by the bridge-encoded name of the extended project property. For example, a variable might be named XPRJ_Claim\$20Number, where "Claim\$20Number" is the bridge-encoded name of the Extended Project Property "Claim Number".

Attachment variables for standard extended project properties are named like basic properties. They begin with a prefix of PRJ_. The names of the standard extended properties are:

Variable Name	Equivalent To
PRJ_PROJECTID	P_ProjectID and I_ProjectID
PRJ_WORKFLOWID	P_WorkflowID
PRJ_DESCRIPTION	P_Description
PRJ_STARTEDON	P_StartedOn
PRJ_STARTEDBY	P_StartedBy
PRJ_LASTTASK	P_LastTask
PRJ_LASTTASKON	P_LastTaskOn
PRJ_DUEDATE	P_DueDate
PRJ_PRIORITY	P_Priority
PRJ_STATUS	P_Status
PRJ_I_TASKID	I_TaskID
PRJ_I_TASK	I_Task
PRJ_I_ASSIGNEDTO	I_AssignedTo
PRJ_I_LOCKEDBY	I_LockedBy
PRJ_I_SINCE	I_Since
PRJ_I_DUEDATE	I_DueDate
PRJ_I_SUSPENDBY	I_SuspendBy
PRJ_I_SUSPENDFOR	I_SuspendFor
PRJ_I_SUSPENDTILL	I_SuspendTill

Rule Parameters

A few rules take parameters that determine or modify their behavior.

In an INI file, the parameters are separated from the rule and from each other by commas. Never insert white space before or after the commas.

Some parameters take a value which the rule uses directly, and some parameters take the name of an attachment variable whose value is then used by the rule or which is otherwise processed by the rule. Be sure to distinguish these two cases carefully.

RecordSet Wildcard

In a few cases where the name of an attachment variable can be passed as a parameter, it will be noted that a RecordSet wildcard may be used. A RecordSet wildcard allows a particular field in each record of an attachment variable record set to be processed.

The general format of a RecordSet wildcard is

`RecordSetName*.FieldName`

RecordSetName is the name of the RecordSet and FieldName is the name of a field in the RecordSet.

For example, the rule DmgBrsEncodeURLField takes parameters that are the names of attachment variables, and any of these may be a RecordSet wildcard.

```
function = dmg_brw32->DmgBrsEncodeURLField,HITS*.FLD_KEYSTRING
```

The above would cause DmgBrsEncodeURLField to process all of the FLD_KEYSTRING variables in the HITS RecordSet. If the attachment contained the following variables, then HITS1.FLD_KEYSTRING and HITS2.FLD_KEYSTRING would be processed.

```
HITS = 2
HITS1.FLD_KEYSTRING = DMGSamples.dbo.Amergen.LName = 'Bradley'
HITS1.FLD_ID = 29
HITS2.FLD_KEYSTRING = DMGSamples.dbo.Amergen.LName = 'Wilson'
HITS2.FLD_ID = 174
```

Server Rule Validations

Many server rules validate certain conditions before they run. If the conditions are not true, the rule will do nothing and will not generate an error. The two common validations are listed here. There are a few other validations specific to particular rules that are described in the context of those rules.

- **Reqtype.** Validate that the reqtype is not the Failure reqtype, as configured in the INI option [ReqTypes] Failure.
- **Session.** Validate the Documange session by checking that the attachment variable SESS_VALID contains 1. See DmgBrsValidateSession.

Some server rules do both validations, some do one or the other, and some do neither. The description of each Server Rule lists its validations. Note that, for those rules that validate the Session, SESS_VALID is not listed as an input attachment variable even though it is a required input for the rule to run.

Rule Descriptions

The rule descriptions that follow have certain standard sections in them, which are illustrated and explained in the sample rule description given here.

DmgBrsSampleRule	<p>A brief description that summarizes the rule's purpose.</p> <p>The usage of the rule on a rules list, including any parameters. Example: <code>function = dmg_brw32->DmgBrsSampleRule,parametername</code></p>
Usage	
Parameters	<p>A table describing each of the rule's parameters. This section is absent if Usage does not list any parameters.</p>
Attachement inputs	<p>A table showing the name and description of each input attachment variable, or "None" if there are no inputs.</p>
Attachement outputs	<p>A table showing the name and description of each output attachment variable, or "None" if there are no outputs.</p>
Remarks	<p>An extended description of the rule, giving necessary details of the rule's operation.</p>
Examples	<p>Any examples that help illustrate the rule's use.</p>
ReqType changes	<p>How the rule changes the reqtype, if at all. See "ReqTypes" in "Using the Documanage Bridge" and the sections that follow it for a discussion of ReqType changes in the bridge. Server rules only.</p>
INI options	<p>A list of INI options that affect the rule, if any. See "Configuring the Documanage Bridge" in "Using the Documanage Bridge" for more information. Server rules only.</p>
Validations	<p>Which validations, if any, the rule performs. Validations are described above. Server rules only.</p>
Platform restrictions	<p>Any platform limitations.</p>
May be followed by	<p>A list of suggested rules, if any, that might logically follow on the same rules list. This list is not meant to be exhaustive, and other possibilities may exist. The list is meant only to be illustrative of the kinds of combinations that might be useful. Server rules only.</p>
See also	<p>A list of related or contrasting rules.</p>

Chapter 7

Rules Reference

This chapter contains descriptions of all Documanager Bridge rules, in alphabetical order.

Be sure to read the introductory material in chapter 1 and the General Reference, particularly "Configuring the Bridge to Documanager", before referencing these descriptions.

DMGBRCDUMPFILE

DmgBrcDumpFile writes a file into an HTTP response for a web server to return to a browser.

Usage

```
function = dmg_brw32-  
>DmgBrcDumpFile,directory,filename_variable,disposition
```

Parameters The parameters provide an alternative way to notify DmgBrcDumpFile of the location of the file to be output to the web server.

The parameters as a whole are optional. However, if you supply directory, then you must supply filename_variable since these two parameters work together.

Parameter	Description
directory	The directory path, with or without a trailing path delimiter, for the directory in which the file to be dumped is stored. Optional.
filename_variable	The name of an attachment variable (case independent) that contains the name of the file to be dumped, including any extension that may be on the filename. Required when directory is supplied, otherwise ignored.
disposition	Use a value of KEEP (case independent) for the disposition to prevent the file described in the first two parameters from being deleted after it is dumped. If this is set to any other value or is omitted, the file will be deleted after it is dumped. Optional. Ignored if either of the first two parameters is empty.

Attachment inputs DmgBrcDumpFile is designed to process ResTypes and reads the following variables from the input queue.

Variable	Description
TARGET_FILE	The path to the file to be dumped. Ignored if parameters are supplied, else required.
HTTP_CONTENTLENGTH	The size in bytes of the file, used in the HTTP header. This variable is created by some Server Rules. Optional. If missing, DmgBrcDumpFile will calculate the size of the file.
HTTP_MIMETYPE	The MIME type of the file, used in the HTTP header. Required.
HTTP_STATUS	The HTTP status, used in the HTTP header. Required.
HTTP_COOKIE_NAME	The name of the Bridge's cookie, used in creating a cookie to put in the HTTP header. Optional.
HTTP_COOKIE_TEXT	The content of the Bridge's cookie. Required if HTTP_COOKIE_NAME is supplied, else ignored.

HTTP_COOKIEPERSISTS	Contains 1 or 0 to indicate whether the Bridge's cookie persists or not. Use 0 for a non-persistent cookie. When it is 0, DmgBrcDumpFile creates a cookie without an expiration date, which the browser will discard when the browser is closed. This has the effect of logging the user out if cookies are used for security. Optional. Default is 1.
HTTP_COOKIEDATE	The expiration date of the Bridge's cookie. Ignored if HTTP_COOKIEPERSISTS is 0. Optional. A non-persistent cookie will be created if this is missing or empty even if HTTP_COOKIEPERSISTS is 1.

Attachment outputs

None.

Remarks

DmgBrcDumpFile creates an HTTP header for the target file and sends both the header and the target file through the CGI interface as an HTTP response by writing them to stdout. In this way, a file can be delivered to the browser. A number of Server Rules create files that can be sent to a browser using this mechanism. The names of all of these rules begin with "DmgBrReturn:" DmgBrReturnThumbnail, for example.

There are two ways that DmgBrcDumpFile can locate the file. One is through its parameters, and the other is through TARGET_FILE. DmgBrcDumpFile will use TARGET_FILE unless the parameters are supplied.

TARGET_FILE is output by the "Return" Server Rules, and includes a complete path to a file. For TARGET_FILE to be useful, DmgBrcDumpFile must be able to see the file using that path.

If using TARGET_FILE isn't feasible or convenient for some reason, use the parameters. They are described above, and illustrated in the examples. The rules that return TARGET_FILE also return a variable called TARGET_FILE_NAME, which contains only the name and extension of the file. This variable can be useful in the second parameter.

The Bridge Server Rules with which you will typically use DmgBrcDumpFile provide all of the required and optional HTTP.... variables.

When DmgBrcDumpFile's parameters are not in use, DmgBrcDumpFile does not delete the file specified in TARGET_FILE. This is because that file is nearly always stored in a Bridge Cache in this case, and the Bridge Server Rules DmgBrInit and DmgBrIdle will manage the deletion of the files.

When DmgBrcDumpFile's parameters are in use, then DmgBrcDumpFile will automatically delete the file after it is dumped unless you set the last parameter to "KEEP", case independent.

Examples

In this example, assume that the Bridge Server and Bridge Client are on separate computers. Assume that the Bridge Server's Cache directory, which is on the Bridge Server's machine, is accessible to the client via sharing. The rules list for running DmgBrcDumpFile for a particular ResType might look like the following.

```
[RESTYPE:RETURN_ANNOTS]
function = ATCw32->ATCLoadAttachment
function = dmg_brw32-
>DmgBrcDumpFile,\\host\bridgecache,Target_file_name,Keep
```

Suppose the variable TARGET_FILE_NAME happened to contain "myfile.jpg" for a particular transaction. In the above example, DmgBrcDumpFile would attempt to process a file whose full path was "\\host\bridgecache\myfile.jpg". Notice that we've set the last parameter to prevent deletion of the file, because it is in the Bridge Server's Cache.

Platform restrictions

None.

See also DmgBrReceiveTempFile
 DmgBruPurgeFiles
 DmgBrsValidateSession
 Rules whose names begin with DmgBrsReturn

DMGBRCRECEIVETEMPFILE

DmgBrcReceiveTempfile receives Bridge Temp Files sent over the queue from the Bridge Server with ATCSendFile.

Usage `function = dmg_brw32->DmgBrcReceiveTempfile,prefix,directory`

Parameters

Parameter	Description
prefix	A prefix that matches the prefix set for the corresponding ATCSendFile. Required.
directory	The directory path, with or without a trailing path delimiter, for the directory in which the Temp File being received is to be stored. Required.

Attachment inputs

DmgBrcReceiveTempfile is designed to process ResTypes and reads the following variable from the input queue.

Variable	Description
TEMP_FILE_NAME	An attachment variable that contains the name of the Temp File being received. Required. If this is absent, the rule does nothing.

Attachment outputs

None.

Remarks

DmgBrcReceiveTempfile makes it convenient to receive Bridge Temp Files over the queue from ATCSendFile. The transfer type used in ATCSendFile must be BINARY.

The use of Bridge Temp Files involves the use of the ReqType configured in [ReqTypes] Redirection. That ReqType is also used in other situations. However, since DmgBrcReceiveTempfile does nothing if TEMP_FILE_NAME is absent, it is safe to use DmgBrcReceiveTempfile on the rules list that processes redirections, and it will do its work of receiving a Temp File only when needed.

Examples

Suppose that a Temp File is being sent from the Bridge Server on a rules list where ATCSendFile is set up as follows:

```
function = atcw32->ATCSendFile,DmgTemp,TARGET_FILE,BINARY
```

Notice that the transfer type is BINARY.

Given the above call to ATCSendFile, the corresponding configuration of DmgBrcReceiveTempfile might be as shown below. Notice that the prefix for DmgBrcReceiveTempfile, "DmgTemp", is the same as that used for ATCSendFile. Suppose that the variable TEMP_FILE_NAME contains "myfile.pdf" for this transaction.

```
[RESTYPE:REDIR]
function = ATCW32->ATCLoadAttachment
function = ATCW32->ATCAppend2Attachment
function = ircltw32->IRCUnloadPage
function = dmg_brw32->DmgBrcReceiveTempfile,DmgTemp,e:\temp\
function = dmg_brw32->DmgBruPurgeFiles,e:\temp\,15,TEMP_FILE_NAME

[HTMLTemplates:REDIR]
Page      = dmg_br_redir.htm
HTTPHeader = dmg_br_header.htm
```

In the above, DmgBrcReceiveTempfile will receive the file sent by ATCSendFile and write it to “e:\temp\myfile.pdf”.

In the example, we’ve added DmgBruPurgeFiles to the rules list to delete files in e:\temp that are more than 15 minutes old so that the directory does not fill up. It operates only if the trigger TEMP_FILE_NAME is present.

If TEMP_FILE_NAME is not on the attachment, neither DmgBrcReceiveTempfile nor DmgBruPurgeFiles will run, but whatever redirection has been requested will still run unhindered.

Platform restrictions None.

See also DmgBrcDumpFile
 DmgBruPurgeFiles

DMGBRSADDDOCUMENTASRENDITION

DmgBrsAddDocumentAsRendition creates a Rendition relationship between two Document Versions.

Usage

```
function = dmg_brw32->DmgBrsAddDocumentAsRendition
```

Attachment inputs

Variable	Description
A Version Document Specification	The first of the two Document Versions that will be related as Renditions. Required.
A Version Document Specification, with “_2” suffixed to each variable name except Cabinet	The second of the two Document Versions that will be related as Renditions. The variables will be named CABINET (which is perforce the same Cabinet as that of the first Version Document Specification), DOC_ID_2, etc. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Cabinet is a Workflow Cabinet, else 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the first Document Version. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the first Document Version. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the first Document Version. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the first Document Version. Exists only if IS_WORKFLOW contains 1.
FLD_LEVELNUMBER_2	The LevelNumber Property of the Folder of the second Document Version. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING_2	The KeyString Property of the Folder of the second Document Version. Exists only if IS_WORKFLOW contains 0.
FLD_ID_2	The Id Property of the Folder of the second Document Version. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING_2	The Keystring Property of the Project of the second Document Version. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsAddDocumentAsRendition creates a Rendition relationship between two Document Versions.

The rule takes two Version Document Specifications as inputs. The two Specifications have the same Cabinet, because Documents that are renditions of each other must be in the same

Cabinet. The other variables in the Specifications are distinguished by appending “_2” to the names of the variables for one of the two Documents versions.

This rule will fail if the Bridge is not connected to Documange Service Release 2 or higher.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsPresentProject
See also	DmgBrsRemoveDocumentAsRendition DmgBrsQueryDocumentRenditions

DMGBRSADDNOTE

DmgBrsAddNote adds a Note to a Documanager Folder or Project.

Usage

```
function = dmg_brw32->DmgBrsAddNote
```

Attachment inputs

Variable	Description
A Folder or Project Specification	Required.
NTE_ADDEDDBY	The AddedBy Property of the new Note. It would be unusual to supply this variable since Documanager will automatically set it to the creator of the Note if it is omitted. Optional.
SESS_USERNAME	The username, as output by DmgBrsValidateSession. Used for the AddedBy Property of the new Note when NTE_ADDEDDBY doesn't exist.
NTE_ADDITIONALINFO	The AdditionalInfo Property of the new Note. Optional.
NTE_DESCRIPTION	The Description Property of the new Note. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Note was added to a Project. Contains 0 if the Note was added to a Folder.
NUM_NOTES	Returns the number of Notes that existed for this Folder or Project before the new Note was added.

Remarks

DmgBrsAddNote creates a new Note. Once a Note is created, it cannot be deleted. However, it can be marked as deleted.

The output attachment variable NUM_NOTES could be used as a NOTES_INDEX input attachment variable for DmgBrsPresentFolder or DmgBrsPresentProject to produce a list that would begin just after the previously existing Notes. The added Note would usually appear at the beginning of such a list, and be the only Note on the list. However, it cannot be guaranteed that the added Note will be at the beginning of the list and will be the only Note on the list. It may be later in the list. This is because it is possible that others were adding Notes at approximately the same time, and because those other Notes might precede this Note.

Examples

The following is a fragment from a rules list that is used to add a Note to a Folder and then call DmgBrsPresentFolder to present the Folder along with the Note that was just added. Notice the use of DmgBruCopyAttachmentVar to copy the value of NUM_NOTES to NOTES_INDEX.

```
function = dmg_brw32->DmgBrsAddNote
function = dmg_brw32-
>DmgBruCopyAttachmentVar, runf, output, NUM_NOTES, output, NOTES_INDEX
function = dmg_brw32->DmgBrsPresentFolder
```

Notice the use of DmgBruCopyAttachmentVar to copy the value of NUM_NOTES to NOTES_INDEX.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Reqtype, Session
Platform restrictions	None.
May be followed by	DmgPresentFolder DmgPresentProject
See also	DmgBrsMarkNoteCurrent DmgBrsMarkNoteDeleted DmgBrsPresentFolder DmgBrsPresentProject

DMGBRSADVANCEPROJECT

DmgBrsAdvanceProject checks in a checked out Project and advances it.

Usage `function = dmg_brw32->DmgBrsAdvanceProject`

Attachment inputs

Variable	Description
A Project Specification	Required.
BRN_ID	The Id of the checkin Branch. Required if DmgBrsPresentProject returned Branches, else ignored.

Attachment outputs

None.

Remarks

DmgBrsAdvanceProject advances a Project along a workflow. Before it can be advanced, a Project must be checked out with DmgBrsCheckOutProject. To get valid Branch Ids for BRN_ID, use DmgBrsPresentProject. However, there may be no Branches, in which case DmgBrsPresentProject will not return any Branches, and BRN_ID is ignored in DmgBrsAdvanceProject. To check in a Project without advancing it, use DmgBrsCheckInProject.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

After the Project is advanced, you may want to return to the Task lists to work on another Task.

DmgBrsGetTasks

See also

DmgBrsCheckInProject

DmgBrsCheckOutProject

DmgBrsPresentProject

DmgBrsUpdateProject

DmgBrsGetTasks

DmgBrsGetTaskQueue

DMGBRSCACHEANNOTATIONSFILE

DmgBrsCacheAnnotationsFile puts a Document annotations file into the Bridge Cache for use by other rules.

Usage

```
function = dmg_brw32->DmgBrsCacheAnnotationsFile
```

Attachment inputs

Variable	Description
A Version Document Specification	The Document whose annotations file is to be cached. Required.

Attachment outputs

All of the attachment outputs are written to both the input and the output queue.

Variable	Description
ANNOTS_PATH	The path to the Document annotations file in the Bridge Cache. Do not alter or delete this file.
ANNOTS_ID	The Id Property of the Document to which the annotations file belongs.
ANNOTS_MAJORVERSION	The MajorVersion Property of the Document to which the annotations file belongs.
ANNOTS_MINORVERSION	The MinorVersion Property of the Document to which the annotations file belongs.
ANNOTS_TAG	A numeric identifier that distinguishes different annotations files for the same version of the Document.
ANNOTS_HDOC	The Dmg_Api handle for the Document to which the annotations file belongs. This handle must not be closed. The handle is valid between the forward and reverse runs of this rule. On the reverse run, this rule deletes this variable from both attachments.
ANNOTS_HFLD	The Dmg_Api handle for the Folder that contains the Document to which the annotations file belongs. This handle must not be closed. The handle is valid between the forward and reverse runs of this rule. On the reverse run, this rule deletes this variable from both attachments.

Remarks

DmgBrsCacheAnnotationsFile puts a Document annotations file into the Bridge Cache for use by other rules, particularly DmgBrsReturnAnnotations. If it finds the correct file is already in the Bridge Cache, it marks the file as recently used so that DmgBrsIdle will not prematurely delete the file when it purges the Cache.

Since it provides the Document annotations file and related handles, DmgBrsCacheAnnotationsFile is useful for writing a custom rule that uses a Document or its annotations file. See DmgBrsValidateSession for details on accessing the Bridge's Dmg_Api Documanager Session handle.

This rule requires Documanager Server 6.4 service release 7 or later.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
See also	DmgBrsCacheContentsFile DmgBrsReturnAnnotations

DMGBRSCACHECONTENTSFILE

DmgBrsCacheContentsFile puts a Document contents file and any Subdocument files into the Bridge Cache for use by other rules.

UsageUsage

```
function = dmg_brw32->DmgBrsCacheContentsFile
```

Attachment inputs

Variable	Description
A Version Document Specification	The Document whose contents file and any Subdocument files are to be cached. Required.

Attachment outputs

All of the attachment outputs are written to both the input and the output queue.

Variable	Description
CONTENTS_COMPRESSED_PATH	The path to the DCZ compressed version of the Document contents file in the Bridge Cache. This variable will be empty when there is no DCZ compressed version of the Document. Do not alter or delete this file.
CONTENTS_DECOMPRESSED_PATH	The path to the decompressed version of the Document contents file in the Bridge Cache. This variable will be empty when CONTENTS_DECOMPRESSED_TYPE is FPP. Do not alter or delete this file.
CONTENTS_DECOMPRESSED_TYPE	The file type of the decompressed version of the Document contents file.
CONTENTS_ID	The Id Property of the Document to which the contents file belongs.
CONTENTS_MAJORVERSION	The MajorVersion Property of the Document to which the contents file belongs.
CONTENTS_MINORVERSION	The MinorVersion Property of the Document to which the contents file belongs.
CONTENTS_TAG	A numeric identifier that distinguishes different contents files for the same version of the Document.
CONTENTS_HDOC	The Dmg_Api handle for the Document to which the contents file belongs. This handle must not be closed. The handle is valid between the forward and reverse runs of this rule. On the reverse run, this rule deletes this variable from both attachments.

CONTENTS_HFLD	The Dmg_Api handle for the Folder that contains the Document to which the contents file belongs. This handle must not be closed. The handle is valid between the forward and reverse runs of this rule. On the reverse run, this rule deletes this variable from both attachments.
CONTENTS_ISCOMPOUND	Contains 1 if the Bridge recognizes the Document at CONTENTS_DECOMPRESSED_PATH as a Compound Document, else contains 0.
CONTENTS_SUBDOCS	Contains the number of Subdocuments in the Compound Document. Exists only if CONTENTS_ISCOMPOUND contains 1.
CONTENTS_SUBDOCSx.field	A record set describing the Subdocuments in the Document, where x is the record number and field is one of the following fields. Exists only if CONTENTS_ISCOMPOUND contains 1.
.PATH	The file path to the Subdocument file in the Bridge File Cache.
.LOCATION	The location property of the Subdocument. This is its stated location within the Compound Document, as returned by the Compound Document plugin that interpreted the Compound Document. This is not the path to the Subdocument in the Bridge File Cache. For example, if a Compound Document has something corresponding to a directory substructure, this could be the path to the Subdocument. May be empty.
.NAME	The name property of the Subdocument, as returned by the Compound Document plugin that interpreted the Compound Document. May be empty.
.TYPE	The file type (extension) of the Subdocument file. This will be the same as the extension on the end of .PATH
.HAS_XPROPS	Contains 1 if the Subdocument has extended properties, else contains 0. Extended Subdocument properties are properties defined by the Compound Document plugin that interpreted the Compound Document. Each extended property has a name and a value. The name contains only alphabetic and numeric characters and the underscore character. The value may be anything.

<code>.NESTED</code>	The name of the record set (row set) containing the extended properties for the Subdocument. Always use this name to locate the extended Subdocument properties, since the name used for the record set could change in future releases of the Bridge.
<code>[.NESTED value]</code>	This is a variable whose name is the value of <code>.NESTED</code> for the Subdocument, and whose value is the number of extended Subdocument properties for this Subdocument. Exists only when <code>.HAS_XPROPS</code> contains 1.
<code>[.NESTED value]y.field</code>	A record set describing the extended Subdocument properties of Subdocument x, where y is the record number and field is one of the following fields. Note that this record set is nested within one <code>DOC_SUBDOCS</code> record, so that each variable name is prepended with the value of <code>.NESTED</code> for the Subdocument.
<code>[.NESTED value]y.NAME</code>	The name of the Subdocument extended property.
<code>[.NESTED value]y.VALUE</code>	The value of the Subdocument extended property.

Remarks DmgBrsCacheContentsFile puts a Document contents file into the Bridge Cache for use by other rules. If it finds the correct file is already in the Bridge Cache, it marks the file as recently used so that DmgBrsIdle will not prematurely delete the file when it purges the Cache. If the Document is a Compound Document, DmgBrsCacheContentsFile also extracts the Subdocument files, puts them into the cache, and outputs information about them.

Various other rules, such as DmgBrsReturnDocument, require the outputs of DmgBrsCacheContentsFile among their inputs. Simply put DmgBrsCacheContentsFile onto the rules list before such rules.

Since it provides the Document contents file and related handles, DmgBrsCacheContentsFile is useful for writing a custom rule that uses a Document or its contents or Subdocumentfiles. See DmgBrsValidateSession for details on accessing the Bridge's Dmg_Api Documanage Session handle.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsCacheAnnotationsFile

DMGBRSCHECKINDOCUMENT

DmgBrsCheckInDocument checks in the checked out copy of a Document, committing any changes to permanent storage.

Usage

```
function = dmg_brw32->DmgBrsCheckInDocument
```

Attachment inputs

Variable	Description
A Current Document Specification	The Current Document Specification for the checked out copy of the Document. Required.
DOC_VERSIONCOMMENT	A value for the VersionComment Property of the Document. Optional. Default is the INI option [CheckInDocument] VersionComment. Whether from the attachment variable or the INI option, the VersionComment must not be empty or longer than 240 characters or the rule will fail.
CHECKIN_TYPE	The checkin type. Possible values are SAME, MAJOR, and MINOR, case independent. Optional. Default is the INI option [CheckInDocument] CheckInType.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the original Document (not the checked out copy) was in a Workflow Cabinet, else contains 0.
CABINET	The Cabinet of the original Document.
DOC_ID	The Id Property of the original Document.
DOC_MAJORVERSION	The MajorVersion Property of the original Document (after checkin).
DOC_MINORVERSION	The MinorVersion Property of the original Document (after checkin).
DOC_VERS	The DOC_VERS of the original Document (after checkin).
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the original Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the original Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the original Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the original Document. Exists only if IS_WORKFLOW contains 1.

Remarks	<p>DmgBrsCheckInDocument operates on the checked out copy of a Document for a particular user, and will fail if the Document specified in the input attachment variables is an original Document or a copy checked out to a different user.</p> <p>CHECKIN_TYPE denotes the type of checkin, with SAME denoting that the changes are to be checked in under the current latest version number, MAJOR denoting that DOC_MAJORVERSION is to be incremented, and MINOR denoting that DOC_MINORVERSION is to be incremented.</p> <p>The outputs provide a complete Version Document Specification of the original Document, that is, the Document that was checked out. Since the checked out copy of the Document could be in a different Cabinet than the original, and since the Id Property of the two Documents will differ, these output variables replace the corresponding input variables with new values. After a Document is checked in, the Cabinet and Id of the checked out copy are no longer of any use.</p> <p>The outputs also provide a complete Folder or Project Specification for the original Document, depending on whether IS_WORKFLOW contains 1 or 0.</p>
ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	[CheckInDocument] CheckInType [CheckInDocument] VersionComment
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	<p>The following rules require Specifications that are output by DmgBrsCheckInDocument. They could be used to present the original Document or its Folder or Project after checkin.</p> <p>DmgBrsPresentDocument</p> <p>DmgBrsPresentFolder</p> <p>DmgBrsPresentProject</p>
See also	<p>DmgBrsCheckOutDocument</p> <p>DmgBrsUndoDocumentCheckOut</p> <p>DmgBrsUpdateDocument</p> <p>DmgBrsUpdateDocumentFile</p>

DMGBRSCHECKINPROJECT

DmgBrsCheckInProject checks in a checked out project without advancing it.

Usage `function = dmg_brw32->DmgBrsCheckInProject`

Attachment inputs

Variable	Description
A Project Specification	The Project to check in. Required.

Attachment outputs

None.

Remarks

DmgBrsCheckInProject checks in a checked out Project without advancing it. To check in a project and advance it, use DmgBrsAdvanceProject. The project remains unmodified unless DmgBrsUpdateProject was used to modify it before it was checked in.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

After the project is checked in, you may want to return to the task lists to select another task.
DmgBrsGetTasks

See also

DmgBrsAdvanceProject
DmgBrsCheckOutProject
DmgBrsPresentProject
DmgBrsUpdateProject

DMGBRSCHECKOUTDOCUMENT

DmgBrsCheckOutDocument checks out a Document.

Usage

```
function = dmgb32->DmgBrsCheckOutDocument
```

Attachment inputs

Variable	Description
A Current Document Specification, with each variable name prefixed with "ORIG_"	The Document that is to be checked out. Required.
A Folder Specification, with each variable name prefixed with "COPY_"	The Folder for the checked out copy of the Document. Optional. Default is the value of the INI option [CheckOutDocument] UsePersonalCabinet, which may be over-ridden with USE_PERSONAL_CABINET.
USE_PERSONAL_CABINET	Over-rides the value of [CheckOutDocument] UsePersonalCabinet for this transaction. 1 means "check out to Personal Cabinet". 0 means "check out in place". Optional.
REASON	The reason for checking out the Document. Must be non-empty. Required.
DUEDATE	The due date for checking the Document in. Required.
SESS_USERNAME	The username, as output by DmgBrsValidateSession. Used when the Document is being checked out to the Personal Cabinet.

Attachment outputs

Variable	Description
RESULT_CHECKOUTDOCUMENT	A result code. Contains zero (0) if checkout was successful. Contains 1 if the requested Document has already been checked out by someone else.
ORIG_IS_WORKFLOW	Contains 1 if the original Document was in a Workflow Cabinet, else contains 0.
ORIGNAME	A URL tail with the original document's name and file type.
ORIG_FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the original Document. 1 is the first Level, that is, the Level nearest the root. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_FLD_KEYSTRING	The KeyString Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.

ORIG_FLD_ID	The Id Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_PRJ_KEYSTRING	The Keysting Property of the Project of the original Document. Exists only if ORIG_IS_WORKFLOW contains 1.
IS_WORKFLOW	Contains 1 if the checked out copy is in a Workflow Cabinet, else contains 0. Exists only if RESULT_CHECKOUTDOCUMENT contains 0.
CABINET	The Cabinet of the checked out copy. Exists only if RESULT_CHECKOUTDOCUMENT contains 0.
DOC_ID	The Id Property of the checked out copy. Exists only if RESULT_CHECKOUTDOCUMENT contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the checked out copy. 1 is the first Level, that is, the Level nearest the root. Exists only if RESULT_CHECKOUTDOCUMENT contains 0 and IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the checked out copy. Exists only if RESULT_CHECKOUTDOCUMENT contains 0 and IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the checked out copy. Exists only if RESULT_CHECKOUTDOCUMENT contains 0 and IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the checked out copy. Exists only if RESULT_CHECKOUTDOCUMENT contains 0 and IS_WORKFLOW contains 1.
COPYNAME	A URL Tail with the Name and FileType of the checked out copy.

Remarks

DmgBrsCheckOutDocument checks out a Document to the Document's Folder, the Personal Cabinet, or to any specified Folder.

The Document being checked out must not itself be a checked out copy or the rule will fail. The rule will also fail if the Document is already checked out to the user requesting that it be checked out. We recommend that a Bridge application should be written to avoid requesting a checkout for a checked out copy or for a Document that is already checked out to the current user.

The error conditions described just above are not likely to change between the time a user gets a list of Documents through a Query and the time that they request the checkout. However, since it is possible for a Document to be checked in during this interval, it is not an error to request checkout for a Document that is checked out to another user. If the Document is checked out to another user at the moment of the request, the rule will succeed and the output variable RESULT_CHECKOUTDOCUMENT will be set to 1.

Depending on how the inputs and the INI option [CheckOutDocument] UsePersonalCabinet are set up, the Document will be checked out in place (i.e., to the Folder containing the

original), to a standard Folder in the Personal Cabinet, or to any specified Folder in any Cabinet.

To check the Document out to any specified Folder, pass in a COPY_ Folder Specification.

To check the Document out in place, omit a COPY_ Folder Specification, and set the INI option [CheckOutDocument] UsePersonalCabinet to OFF, or over-ride the option with USE_PERSONAL_CABINET set to 0.

To check the Document out to the Personal Cabinet, omit a COPY_ Folder Specification, and set the INI option [CheckOutDocument] UsePersonalCabinet to ON, or over-ride the option with USE_PERSONAL_CABINET set to 1. When the checked out copy is put into the Personal Cabinet, it is always put into a Folder whose name is “Checked-Out Documents”.

The output attachment variables provide a Folder or Project Specification for the original Document, depending on whether it was in a Workflow Cabinet or not, a Document Specification for the checked out copy, and a Folder or Project Specification for the checked out copy.

Examples

To check a Document out to a selected folder, the input attachment variables might look like this:

```
ORIG_CABINET = MyCabinet
ORIG_DOC_ID = 74
COPY_CABINET = MyOtherCabinet
COPY_FLD_LEVELNUMBER = 2
COPY_FLD_ID = 228
REASON = Update invoice
DUEDATE = 4/15/2005
```

To check the Document out to its own folder or to the Personal Cabinet, the variables that begin with “COPY_” would be omitted, and the INI option [CheckOutDocument] UsePersonalCabinet would then govern the destination of the checked out copy, unless over-ridden with USE_PERSONAL_CABINET.

For example, suppose [CheckOutDocument] UsePersonalCabinet is OFF. The following would cause the Document to be checked out to the Personal Cabinet because USE_PERSONAL_CABINET overrides the option.

```
ORIG_CABINET = MyCabinet
ORIG_DOC_ID = 74
USE_PERSONAL_CABINET = 1
REASON = Update invoice
DUEDATE = 4/15/2005
```

Again supposing that [CheckOutDocument] UsePersonalCabinet is OFF, the following would cause the Document to be checked out in place.

```
ORIG_CABINET = MyCabinet
ORIG_DOC_ID = 74
REASON = Update invoice
DUEDATE = 4/15/2005
```

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

[CheckOutDocument] UsePersonalCabinet

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by You may want to follow DmgBrsCheckOutDocument with the following rules to present the checked out copy and/or its Folder or Project. There is one consideration, which is that these rules will fail if RESULT_CHECKOUTDOCUMENT contains 1. To prevent this, the Bridge application should not let a user select a Document for checkout if it is already checked out to someone else and one of these rules follows DmgBrsCheckOutDocument.

DmgBrsPresentDocument

DmgBrsPresentFolder

DmgBrsPresentProject

See also DmgBrsCheckInDocument
DmgBrsUndoDocumentCheckOut
DmgBrsUpdateDocument
DmgBrsUpdateDocumentFile

DMGBRSCHECKOUTPROJECT

DmgBrsCheckOutProject checks out a Project.

Usage

```
function = dmg_brw32->DmgBrsCheckOutProject
```

Attachment inputs

Variable	Description
IS_WORKFLOW	If this exists and contains 0, this rule does nothing. Optional. Defaults to 1.
A Project Specification	The Project to check out. Required.

Attachment outputs

Variable	Description
RESULT_CHECKOUTPROJECT	Contains 0 if the Project was checked out to the user, or was already checked out to the user. Contains 1 if the Project could not be checked out.

Remarks

DmgBrsCheckOutProject checks out a Project. Typically the intent is to modify the Project before checking it back in. To check the Project back in without modification, use DmgBrsCheckInProject without using DmgBrsUpdateProject.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

DmgBrsPresentProject, to present the Project that was just checked out.

See also

DmgBrsAdvanceProject
DmgBrsCheckInProject
DmgBrsPresentProject
DmgBrsUpdateProject

DMGBRSCHECKPAUSEDSEVER

DmgBrsCheckPausedServer runs on the PAUSE rules list to monitor Documanage and resume IDS when appropriate.

Usage	function = dmg_brw32->DmgBrsCheckPausedServer
Attachment inputs	None.
Attachment outputs	None.
Remarks	<p>DmgBrsCheckPausedServer is a specialized rule designed to be run only on the PAUSE rules list.</p> <p>This rule does the following:</p> <ul style="list-style-type: none"> • If the Documanage Bridge is not initialized, it attempts to re-initialize it. If re-initialization succeeds, it will resume IDS, that is, bring it out of its paused state. • If the Documanage Bridge is initialized, but its session is disconnected or not current, it will shut down the Bridge. • If the Documanage Bridge is initialized and its session is connected and current, it will run the diagnostic specified in [DMGDiagnosics]StartupDiagnostic, if any. If this diagnostic fails, it will shut down the Bridge. If there was no diagnostic, or if the diagnostic succeeds, it will resume IDS. <p>DmgBrsCheckPausedServer does not need to be preceded by DmgBrsCopyAttachment.</p> <p>See the Internet Document Server Guide for more information about pausing IDS.</p>
Examples	<p>Here is the typical configuration of the PAUSE rules list.</p> <pre>[ReqType:PAUSE] function = atcw32->ATCLogTransaction function = atcw32->ATCLoadAttachment function = atcw32->ATCUnloadAttachment function = dmg_brw32->DmgBrsCheckPausedServer</pre>
ReqType changes	None.
INI options	<p>[DMGDiagnosics] OnFail_Startup</p> <p>[DMGDiagnosics] StartupDiagnostic</p>
Validations	Reqtype
Platform restrictions	None.
See also	<p>DmgBrsIdle</p> <p>DmgBrsInit</p> <p>DmgBrsRunDocumanageDiagnostic</p>

DMGBRSCOPYATTACHMENT

DmgBrsCopyAttachment copies all of the input attachment variables to the output attachment.

Usage `function = dmg_brw32->DmgBrsCopyAttachment`

Attachment inputs None.

Attachment outputs Forward run:

Variable	Description
RESULTS	Always set to SUCCESS.
USERID	Always set to DMG_BR.
HTTP_STATU S	Always set to 200 Ok. Subsequent rules occasionally modify this value.

Reverse run:

Variable	Description
RESULTS	Always set to SUCCESS.
EXCEPTIONMESSA GE	Always set empty.

Remarks Most server rules read their input attachment variables from the output queue. On its forward run, DmgBrsCopyAttachment copies all of the attachment variables on the input queue to the output queue for use by other Documanage Bridge rules.

DmgBrsCopyAttachment does two other things on its forward run. It sets three attachment variables on the output queue, and it then writes the value of all attachment variables (see DmgBrsLogin for an exception) to the Documanage Bridge log.

On its reverse run, DmgBrsCopyAttachment does three things. It sets two attachment variables on the output queue, it deletes the variable ERRORFILE, and it then writes the value of all attachment variables to the Documanage Bridge log.

Examples The typical placement of DmgBrsCopyAttachment on a rules list:

```
[ ReqType:MOST_REQTYPES ]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
; other rules...
```

ReqType changes None.

INI options None.

Validations None.

Platform restrictions None.

See also [DmgBruCopyAttachmentVar](#)

DMGBRSCOPYDOCUMENT

DmgBrsCopyDocument makes a copy of a Document in any specified folder or project.

Usage function = dmg_brw32->DmgBrsCopyDocument

Attachment inputs

Variable	Description
A Current Document Specification	The Document to copy. Required.
A Folder or Project Specification, with each variable name prefixed with "DEST_"	The destination for the copy, that is, the Folder or Project where the copy will be made. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document to copy was in a Workflow Cabinet, else contains 0.
DOCSPEC	A DOCSPEC for the Document to copy.
DOC_MAJORVERSION	The MajorVersion Property of the Document to copy.
DOC_MINORVERSION	The MinorVersion Property of the Document to copy.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document to copy. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document to copy. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document to copy. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the Document to copy. Exists only if IS_WORKFLOW contains 1.
DEST_DOCSPEC	A DOCSPEC for the copy.
DEST_DOC_ID	The Id Property of the copy.
DEST_DOC_MAJORVERSION	The MajorVersion Property of the copy.
DEST_DOC_MINORVERSION	The MinorVersion Property of the copy.

Remarks DmgBrsCopyDocument makes a copy of a Document in any specified Folder or Project.

Note that the input and output attachment variables taken together provide Document Specifications and Folder or Project Specifications for both the Document to copy and the copy. The variables for the copy are all prefixed with "DEST_".

Examples Here are possible input attachment variables for copying a Document to a Folder. In this example, the variable DEST_FLD_LEVELNUMBER is omitted because the destination

Cabinet has only one level. The destination Folder is identified with its Id Property, but could also have been identified with its KeyString Property using DEST_FLD_KEYSTRING.

```
CABINET = MyCabinet
DOC_ID = 722
DEST_CABINET = MyCabinet
DEST_FLD_ID = 22998
```

Here is a possible rules list that causes the Document copy to be presented after it is made. Since DmgBrsPresentDocument requires a Version Document Specification, the values of the copy's variables that comprise that Specification must be copied to variable names that DmgBrsPresentDocument will recognize. This will over-write the values of output variables for the Document to copy. Notice that we need not copy the value of DEST_CABINET to CABINET because they are the same. Let's presume that our system has only one Cabinet. Even though the parameters for DmgBruCopyAttachmentVar may line wrap here, they should not be allowed to do so in an actual rules list.

```
[ ReqType:PRESENT_COPY ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCopyDocument
function = dmg_brw32-
>DmgBruCopyAttachmentVar, runf, output, dest_doc_id, runf, output,
doc_id
function = dmg_brw32-
>DmgBruCopyAttachmentVar, runf, output, dest_doc_majorversion,
output, doc_majorversion
function = dmg_brw32->DmgBruCopyAttachmentVar, runf, output, dest_
doc_minorversion, output, doc_minorversion
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsPresentDocument
```

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder
See also	DmgBrsCreateDocumentShortcut DmgBrsMoveDocument

DMGBRSCREATEDIARY

DmgBrsCreateDiary creates a new Diary.

Usage

```
function = dmgb_rbw32->DmgBrsCreateDiary
```

Attachment inputs

Variable	Description
DIA_TITLE	The Title Property of the Diary to create. Optional.
DIA_DESCRIPTION	The Description Property of the Diary to create. Optional.
DIA_PRIORITY	The Priority Property of the Diary to create. Optional.
DIA_STARTON	The StartOn Property of the Diary to create. Optional.
DIA_DUEON	The DueOn Property of the Diary to create. Optional.
DIA_NEXTALERTTIME	The NextAlertTime Property of the Diary to create. Optional.
DIA_OWNER	The Owner Property of the Diary to create. This is automatically set to the current user by Documanager if it isn't set here, so it isn't usually necessary to set this Property. Optional.
DIA_REFERENCE	The Reference Property of the Diary to create. Optional.
DIA_STATUS	The Status Property of the Diary to create. Optional.
MAKE_FILTER	If this exists and is set to 1, the output attachment variables will include a DIARY_FILTER set to the Keysting Property of the new Diary. Optional. Default is the INI option [CreateDiary] MakeFilter.

Attachment outputs

Variable	Description
DIA_TITLE	The Title Property of the new Diary.
DIA_DESCRIPTION	The Description Property of the new Diary.
DIA_PRIORITY	The Priority Property of the new Diary.
DIA_CREATEDON	The CreatedOn Property of the new Diary.
DIA_STARTON	The StartOn Property of the new Diary.
DIA_DUEON	The DueOn Property of the new Diary.
DIA_COMPLETEDON	The CompletedOn Property of the new Diary.

DIA_LASTMODIFIEDON	The LastModifiedOn Property of the new Diary.
DIA_NEXTALERTTIME	The NextAlertTime Property of the new Diary.
DIA_ASSIGNEDBY	The AssignedBy Property of the new Diary.
DIA_OWNER	The Owner Property of the new Diary.
DIA_REFERENCE	The Reference Property of the new Diary.
DIA_STATUS	The Status Property of the new Diary.
DIA_ID	The Id Property of the new Diary.
DIA_KEYSTRING	The Keystring Property of the new Diary.
DIARY_FILTER	A filter that specifies the new Diary, suitable for use with DmgBrsQueryDiaries. Exists only if MAKE_FILTER or its default value is on.

Remarks DmgBrsCreateDiary creates a new Diary. Any of the Properties listed among the input attachment variables may be supplied when creating the Diary. However, notice that none of the input attachment variables are required. Thus, a Diary can be created without supplying any input attachment variables.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [CreateDiary] MakeFilter

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsQueryDiaries
If MAKE_FILTER or its default value is on, DmgBrsQueryDiaries will return information about the new Diary in record set format, plus necessary outputs for doing further searches.

See also DmgBrsDeleteDiary
DmgBrsQueryDiaries
DmgBrsUpdateDiary

DMGBRSCREATEDOCUMENT

DmgBrsCreateDocument creates a new Document in a Folder or Project.

Usage

```
function = dmg_brw32->DmgBrsCreateDocument
```

Attachment inputs

Variable	Description
A Folder or Project Specification	A location for the new Document. Required.
DOC_FILEPATH	The path of the file being imported. See the remarks below for more information. Required.
HFLD	This variable is created automatically by DmgBrsCreateFolder and DmgBrsSetupFolder to provide a performance enhancement when either of those rules precedes DmgBrsCreateDocument. Optional. This variable is read from the Input Queue.
DOC_UPLOADERERR	Used to signal to DmgBrsCreateDocument that there was an error while uploading a file. If this exists and contains any value whatsoever, DmgBrsCreateDocument will immediately fail. See the remarks below for more information. Optional.
DOC_DELFILE	Set this to 1 to cause the file in DOC_FILEPATH to be deleted after DmgBrsCreateDocument runs, whether it succeeds or not. Set this to 0 to prevent file deletion. Optional. Default is the INI option [CreateDocument] DeleteFile.
DOC_CATEGORY	The Category Property of the new Document. If this is not a valid Category (case sensitive), DmgBrsCreateDocument will fail. Required.
DOC_SUBCATEGORY	A SubCategory Property for the new Document. It must be a valid SubCategory (case sensitive) for the Category or DmgBrsCreateDocument will fail. Optional.
DOC_FILETYPE	A FileType Property for the new Document. If this is not supplied, then the extension on DOC_FILEPATH is automatically used for the Document's FileType Property. Optional.
DOC_NAME	A Name Property for the new Document. If this variable doesn't exist or is empty, DmgBrsCreateDocument will use the name of the file from DOC_FILEPATH. Optional.
DOC_DATE	A Date Property for the new Document. This may be any date or date and time desired. Documanager will automatically set the AddedOn Property to the date and time the Document was imported. Optional.
DOC_STATUS	A Status Property for the new Document. It must be a valid Status (case sensitive) for the Category or DmgBrsCreateDocument will fail. Optional.
DOC_DESCRIPTION	A Description Property for the new Document. If this exceeds DMG_P_DOCUMENT_DESCRIPTION_LEN characters, DmgBrsCreateDocument will fail. Optional.

DOC_FLAG1	A Flag1 Property for the new Document. It must be a valid Flag1 (case sensitive) for the Category or DmgBrsCreateDocument will fail. Optional.
DOC_FLAG2	A Flag2 Property for the new Document. It must be a valid Flag2 (case sensitive) for the Category or DmgBrsCreateDocument will fail. Optional.
DOC_KEYWORD1	A Keyword1 Property for the new Document. If it exceeds DMG_P_DOCUMENT_KEYWORD1_LEN characters or if Keywords are not enabled in Documanager, DmgBrsCreateDocument will fail. Optional.
DOC_KEYWORD2	A Keyword2 Property for the new Document. If it exceeds DMG_P_DOCUMENT_KEYWORD2_LEN characters or if Keywords are not enabled in Documanager, DmgBrsCreateDocument will fail. Optional.
XDOC_propertyname	Variables containing the Extended Properties for the new Document, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name). These should be appropriate for the Category. Any that are not will be ignored. Optional.
SESS_USERNAME	The username, as output by DmgBrsValidateSession. Used to set the Author Basic Document Property.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was imported into a Workflow Cabinet, else contains 0.
DOC_ID	The ID Property of the new Document.
DOC_MAJORVERSION	The MajorVersion Property of the new Document.
DOC_MINORVERSION	The MinorVersion Property of the new Document.
DOCSPEC	The DOCSPEC of the new Document.
DOC_VERS	The DOC_VERS of the new Document.
DOC_NAME	The Name Property of the new Document.
DOC_FILETYPE	The FileType Property of the new Document.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the new Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the new Document. Exists only if IS_WORKFLOW contains 0.

FLD_ID	The ID Property of the Folder of the new Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the new Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsCreateDocument creates a new Document.

You must specify the Folder or Project that will contain the new Document (Folder or Project Specification), the location of the file for the new Document (DOC_FILEPATH), and the new Document's Category (DOC_CATEGORY). In addition to these requirements, you may specify other Basic and Extended Document Properties, and you may specify whether the file will be deleted or not after the rule runs.

DOC_FILEPATH may be a local directory path, beginning with the drive, or a relative path, although relative paths are not recommended due to their ambiguity. The path must, as always, be in a format readable by the operating system on which this rule is running. Alternatively, this may be a UNC path. In any case, the path must end with a filename and must include the correct extension of the file if you do not supply DOC_FILETYPE. To avoid name collisions while uploading to some directory on a busy system, you may want to give the file a temporary name. Do not change the extension unless you supply DOC_FILETYPE. When using temporary names, you will probably want to make sure that DOC_NAME is set correctly to keep the temporary name from becoming the Name Property of the Document.

You may use DOC_UPLOADERR to signal to this rule that there was an error uploading the file. If DOC_UPLOADERR exists and contains any value whatsoever, DmgBrsCreateDocument will immediately fail. There are two defined values (not case sensitive) that may be passed in DOC_UPLOADERR that cause DmgBrsCreateDocument to create pre-defined error messages. The value "VIRUS" causes an error that indicates the file to be uploaded contained a virus. The value "UPLOADFAILED" causes an error that indicates that file upload failed. If any other value is passed in DOC_UPLOADERR, that value will become the text of the error message that the rule reports. Generally, any virus checking of the uploaded file would be done by the IDS client, perhaps with a rule that runs on the initiating REQTYPE list if a CGI client is in use, or perhaps with other custom programming in a Java or ASP or other client. In any case, this variable may be used to signal a problem to DmgBrsCreateDocument, and from there to the user in the standard Bridge fashion.

A Bridge application developer can ignore HFLD. If HFLD exists, and if its properties match the properties in the Folder or Project Specification, DmgBrsCreateDocument will create the Document in this Folder or Project rather than searching for the Folder or Project again. Avoiding the search improves performance.

Note that the Content Management properties (DOC_OBSOLETE, DOC_RELEASED, and DOC_APPROVED) cannot be set with DmgBrsCreateDocument. They must be set with DmgBrsUpdateDocumentCM after the Document has been created. DmgBrsUpdateDocumentCM may follow DmgBrsCreateDocument on the same rules list.

The output variables of DmgBrsCreateDocument provide a Document Specification for the new Document, as well as a Folder or Project Specification. This makes it convenient to follow DmgBrsCreateDocument with other rules or with subsequent transactions that operate on either the new Document or its container.

DmgBrsCreateDocument may be used alone to create a Document. But note that it is also designed to be combined with DmgBrsCreateFolder on a rules list so that a Folder and Document can be created in one transaction. Run DmgBrsCreateFolder before DmgBrsCreateDocument when you want the Folder to be created if it doesn't exist. See the example below and DmgBrsCreateFolder.

DmgBrsCreateDocument can also be used in combination with DmgBrsSetupFolder so that a Folder can be found and a Document created in it in one transaction. Run DmgBrsSetupFolder before DmgBrsCreateDocument when you want DmgBrsCreateDocument to fail if the Folder doesn't exist. See DmgBrsSetupFolder.

Examples

In this example, we will create a Folder, if it doesn't already exist, create a Document, and set one of the Document's Content Management Properties, all in one transaction.

Here is what the rules list might look like.

```
[ ReqType:CREATE_DOC]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsCreateFolder
  function = dmg_brw32->DmgBrsCreateDocument
  function = dmg_brw32->DmgBrsUpdateDocumentCM
```

Suppose there is a Cabinet called Customers with one Level. The Extended Properties for that level are Last Name, which is Part of Folder Key, First Name, which is Part of Folder Key, State, and Phone Number.

To prepare for DmgBrsCreateFolder, we will provide the necessary XFLD_L1_propertyname variables to create a Folder in the Customers Cabinet.

DmgBrsCreateDocument requires a Folder Specification. Since DmgBrsCreateFolder precedes it on the rules list, DmgBrsCreateFolder will provide the Folder Specification. In the input attachment variables, we need only provide the file and the Category Property for the new Document. Of course, we could provide other Document Properties as desired.

DmgBrsUpdateDocumentCM requires a Document Specification. Since DmgBrsCreateDocument precedes it on the rules list, DmgBrsCreateDocument will provide the Document Specification. In the input attachment variables, we need only provide the changed Content Management Property. In this example, we'll set the Approved property.

Here's what our input attachment variables might be.

```
CABINET = Customers
XFLD_L1_Last$20Name = Doe
XFLD_L1_First$20Name = Jane
XFLD_L1_State = IA
XFLD_L1_Phone$20Number = 641-555-1212
DOC_FILEPATH = e:\importfiles\myfile.jpg
DOC_CATEGORY = Photo
DOC_APPROVED = 1
```

Notice that the entire transaction will succeed whether or not the Jane Doe Folder existed or not at the start of the transaction. DmgBrsCreateFolder will create it if it is missing, and will output the Folder Specification in either case.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	[CreateDocument]
Validations	Session and Reqtype
Platform restrictions	None.

May be followed by	DmgBrsPresentDocument
	DmgBrsPresentFolder
	DmgBrsPresentProject
	DmgBrsUpdateDocumentCM
See also	DmgBrsCreateFolder
	DmgBrsPresentDocument
	DmgBrsPresentFolder
	DmgBrsPresentProject
	DmgBrsSetupFolder
	DmgBrsLogPropertySets
	DmgBrsUpdateDocumentCM
	DmgBrsValidateSession, particularly the parameter

DMGBRSCREATEDOCUMENTSHORTCUT

DmgBrsCreateDocumentShortcut creates a Shortcut to a Document.

Usage

```
function = dmg_brw32->DmgBrsCreateDocumentShortcut
```

Attachment inputs

Variable	Description
A Current Document Specification	The original Document, that is, the Document for which a Shortcut is to be created. Required.
A Folder or Project Specification, with each variable name prefixed with "DEST_"	The destination for the Shortcut, that is, the Folder or Project where the Shortcut will be made. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the original Document was in a Workflow Cabinet, else contains 0.
DOCSPEC	A DOCSPEC for the original Document.
DOC_MAJORVERSION	The MajorVersion Property of the original Document.
DOC_MINORVERSION	The MinorVersion Property of the original Document.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the original Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the original Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the original Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the original Document. Exists only if IS_WORKFLOW contains 1.
DEST_DOCSPEC	A DOCSPEC for the Shortcut.
DEST_DOC_ID	The Id Property of the Shortcut.
DEST_DOC_MAJORVERSION	The MajorVersion Property of the Shortcut.
DEST_DOC_MINORVERSION	The MinorVersion Property of the Shortcut.

Remarks

DmgBrsCreateDocumentShortcut creates a Shortcut to a Document in any specified Folder or Project.

Note that the input and output attachment variables taken together provide Document Specifications and Folder or Project Specifications for both the original Document and the Shortcut. The variables for the Shortcut are all prefixed with "DEST_".

Examples	See the description of DmgBrsCopyDocument. The example for that rule applies equally well to DmgBrsCreateDocumentShortcut.
ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder
See also	DmgBrsCopyDocument DmgBrsMoveDocument

DMGBRSCREATEFOLDER

DmgBrsCreateFolder creates up to one new Folder at each Level of a Cabinet.

Usage

```
function = dmg_brw32->DmgBrsCreateFolder
```

Attachment inputs

Variable	Description
FLD_LEVELNUMBER	The LevelNumber of the Target Folder, the main folder to be created by this rule. 1 is the first Level, that is, the Level nearest the root. Optional. Default is 1.
CABINET	The name of the Cabinet in which the new folder or folders will be created. Required.
XFLD_Lx_propertyname	The Extended Properties of the Folders that will be created, where x is the LevelNumber and propertyname is the Bridge-encoded Folder Property Name. Required for FLD_LEVELNUMBER, and for related Folders at other Levels if needed to make the Target Folder findable after creation. For each unique x, DmgBrsCreateFolder will attempt to create a Folder at that Level. Required Properties must be supplied or the rule will fail.

Attachment outputs

Variable	Description
FLD_LEVELNUMBER	The LevelNumber of the Target Folder. This is always output even if it was not specified as an input.
FLD_ID	The Id Property of the Target Folder.
FLD_KEYSTRING	The KeyString Property of the Target Folder.
HFLD	A Dmg_API handle to the Target Folder. The reverse run of DmgBrsCreateFolder removes this from the attachment. It serves as an input to DmgBrsCreateDocument when that rule follows this one on a rules list. This variable is written to the Input Queue.

Remarks

DmgBrsCreateFolder creates up to one new Folder at each Level of a Cabinet. Even if Folders are being created at more than one Level, one Folder is always designated as the Target Folder with FLD_LEVELNUMBER.

The propertyname in each XFLD_Lx_propertyname variable must be Bridge Encoded, and must be the Folder Property Name as given in the Business Data Tables Dialog in the Documanager Administrator, and not the DB Field Name. For each unique x, all of the required Extended Properties (those that are Part of Folder Key) must be supplied, or the rule will fail. If the Folder described in the XFLD_Lx_propertyname variables already exists, as determined by examining the required Properties, it is not an error. However, no attempt is made to update the Folder with values supplied for non-required Properties.

This rule is very flexible in its application, so it may be helpful to consider the various cases to which it can be applied.

For a Cabinet with one Level, this rule is very simple. Only one Folder can be created in a run of this rule, and the Target Level will be 1, so FLD_LEVELNUMBER does not need to be

supplied. XFLD_Lx_propertyname variables must be supplied for any required Extended Properties, and may be provided for other desired Extended Properties.

For a Cabinet with more than one Level, care must be taken to ensure that, after all Folders described in the XFLD_Lx_propertyname variables have been created, the Target Folder can be found. If it can't be found, the rule will fail.

For example, suppose there is a Cabinet with two levels that is empty, i.e., has no Folders in it. Suppose further that the Target Folder is at level one and that you supply XFLD_Lx_propertyname variables only for Level 1. This will fail because a Documanager search cannot find a Folder in a multi-level Cabinet unless that Folder is part of a branch of relationships, as defined in the Relationships Dialog of the Documanager Administrator, that extends through all Levels of the Cabinet.

Let's consider another example with our empty two Level Cabinet. If you supply XFLD_Lx_propertyname variables for Level 1 and Level 2, and those variables create Folders that are related, then DmgBrsCreateFolder will succeed, regardless of whether you set FLD_LEVELNUMBER to 1 or to 2. After DmgBrsCreateFolder runs, there will be two Folders in the Cabinet, one at each Level, and a search of the Cabinet will be able to find those Folders.

Now that our two Level Cabinet has Folders in it, you may add more Folders in two ways. First, you may add two new Folders at once, one at each Level, just as you did to add the first two Folders to the Cabinet. Or, you can add a Folder at Level 2 of the Cabinet as long as it relates to the Folder at Level 1, so that it can be found after it is added.

The output attachment variables of DmgBrsCreateFolder provide a Folder Specification for the Target Folder. The HFLD output variable has little use outside of its role in improving the performance of DmgBrsCreateDocument.

Examples

Suppose there is a Cabinet called Customers with one Level. The Extended Properties for that level are Last Name, which is Part of Folder Key, First Name, which is Part of Folder Key, State, and Phone Number. Here is how you might set up the input attachment variables to create a Folder in that Cabinet.

```
CABINET = Customers
XFLD_L1_Last$20Name = Doe
XFLD_L1_First$20Name = Jane
XFLD_L1_State = IA
XFLD_L1_Phone$20Number = 641-555-1212
```

If no Folder already exists for Jane Doe, then one will be created with the above Properties, and the rule will succeed. If a Folder already exists for Jane Doe, the rule will still succeed. However, the existing State and Phone Number for Jane Doe won't be modified.

Suppose there is a Cabinet called States that has three Levels. The table at Level one has one Property, State, which is the key. The table at Level two has two Properties: County, which is Part of Folder Key, and State, which relates it to the table at Level one. The table at Level three has two Properties: City, which is Part of Folder Key, and County, which relates it to the table at Level two.

If there are no Folders in the States Cabinet, here are possible input attachment variables to add the first branch of three related Folders simultaneously.

```
CABINET = States
FLD_LEVELNUMBER = 3
XFLD_L1_State = IA
XFLD_L2_State = IA
XFLD_L2_County = Polk
XFLD_L3_County = Polk
```

```
XFLD_L3_City = Des Moines
```

For the above example, we chose to make the new City our Target Folder, by setting FLD_LEVELNUMBER to its Level, but we could have chosen the County or the State. All three Folders will be created whether FLD_LEVELNUMBER is 1, 2, or 3. The main difference is in the output attachment variables.

In our States Cabinet, now that we have some Folders, we could add a new Folder at Level 3 that relates to the existing Folder at Level 2 with input attachment variables like the following.

```
CABINET = States
FLD_LEVELNUMBER = 3
XFLD_L3_County = Polk
XFLD_L3_City = Johnston
```

If we attempted to add a new County in IA but didn't simultaneously add a new City in that County, the rule would fail because the new County would not be findable in a search.

In our States Cabinet, we could also add new Folders simultaneously at levels two and three as follows.

```
CABINET = States
FLD_LEVELNUMBER = 2
XFLD_L1_State = IA
XFLD_L2_State = IA
XFLD_L2_County = Black Hawk
XFLD_L3_County = Black Hawk
XFLD_L3_City = Waterloo
```

Notice in this last example, we supplied XFLD_L1_State = IA. IA already exists as a state, so supplying it is redundant. However, it is not an error, and the rule succeeds. If FLD_LEVELNUMBER were 1, then supplying XFLD_L1_State = IA would be required since the Properties of the Target Folder must always be supplied.

Now here's a very special case. Suppose we know that the LOB table that has the City in it contains a record whose City is Cedar Rapids and whose County is Linn. Since there is currently no County record for Linn, Cedar Rapids doesn't exist as a Folder since it isn't related to existing Folders. However, we could create the corresponding Folders that relate to Cedar Rapids, and then Cedar Rapids would become a Folder and become findable. Here's how:

```
CABINET = States
FLD_LEVELNUMBER = 1
XFLD_L1_State = IA
XFLD_L2_State = IA
XFLD_L2_County = Linn
```

For this special case to work, the existing but unrelated LOB table entry must be in the table at the Level that has the highest number.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentFolder DmgBrsCreateDocument

You may supply all the input attachment variables necessary for DmgBrsCreateFolder and DmgBrsCreateDocument to run on the same rules list. This allows you to create a Folder and put a Document into it in one operation. Since DmgBrsCreateFolder succeeds if the Folder already exists, you need not worry whether the Folder exists or not. It will be created if need be, and there will be no problem if it doesn't need to be created.

See also

- DmgBrsCreateDocument
- DmgBrsLogPropertySets
- DmgBrsSetupFolder
- DmgBrsCreateProject

DMGBRSCREATEPROJECT

DmgBrsCreateProject creates a new Workflow Project.

Usage

```
function = dmg_brw32->DmgBrsCreateProject
```

Attachment inputs

Variable	Description
CABINET	The name of the Workflow Cabinet for the new Project. Required.
PRJ_DESCRIPTION	The Description of the new Project. The value passed in here will be truncated to 100 characters. Required.
PRJ_DUEDATE	The DueDate of the new Project. Required.
PRJ_PRIORITY	The Priority of the new Project. This must be a number from 1 through 10 inclusive or the rule will fail. Required.
XPRJ_propertyname	Extended Project Properties of the new Project, where propertyname is the Bridge-encoded Folder Property Name. Do not set the Standard Extended Project Properties, whose DB Field Names begin with I_ or P_. Any Extended Project Properties that are required (Part of Folder Key) must be provided or the rule will fail.

Attachment outputs

Variable	Description
PRJ_KEYSTRING	The Keystring Property of the new Project.

Remarks

DmgBrsCreateProject creates a new Workflow Project. If you want to transform an existing Folder into a Workflow Project, use DmgBrsCreateProjectFromFolder. The propertyname in each XPRJ_propertyname variable must be Bridge Encoded, and must be the Folder Property Name as given in the Business Data Tables Dialog in the Documanage Administrator, and not the DB Field Name.

The output attachment variable plus the input attachment variable CABINET forms a Project Specification that may be used in other rules that require one as an input.

Examples

Suppose we wanted to start a new claim in the Claims Workflow Cabinet. Assume Claim Number is the only Property that is Part of Folder Key. The input attachment variables might look like the following.

```
CABINET = Claims
PRJ_DESCRIPTION = hail storm damage
PRJ_DUEDATE = 07/31/2005
PRJ_PRIORITY = 5
XPRJ_Claim$20Number = 54793
XPRJ_Vehicle$20ID = 7793A285XZ279BCB
XPRJ_Policy$20Number = 443829
```

Notice in the above that we set Claim Number. Since it is Part of Folder Key, we must do so, or the rule will fail.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options	None.
Validations	Session and Rectype
Platform restrictions	None.
May be followed by	DmgBrsPresentProject DmgBrsCheckOutProject
See also	DmgBrsCheckOutProject DmgBrsCreateProjectFromFolder DmgBrsLogPropertySets DmgBrsPresentProject

DMGBRSCREATEPROJECTFROMFOLDER

DmgBrsCreateProjectFromFolder transforms an existing Folder into a Workflow Project.

Usage

```
function = dmg_brw32->DmgBrsCreateProjectFromFolder
```

Attachment inputs

Variable	Description
A Folder Specification	The Folder that is to be made into a Project. Note that the CABINET in this Specification will be the Cabinet of the Folder, and not the Workflow Cabinet. Required.
WF_CABINET	The Workflow Cabinet for the new Project. Required.
PRJ_DESCRIPTION	The Description of the new Project. The value passed in here will be truncated to 100 characters. Required.
PRJ_DUEDATE	The DueDate of the new Project. Required.
PRJ_PRIORITY	The Priority of the new Project. This must be a number from 1 through 10 inclusive or the rule will fail. Required.
XPRJ_propertyname	Editable Extended Project Properties of the new Project that are to be changed from their current values, where propertyname is the Bridge-encoded Folder Property Name. Do not set the Standard Extended Project Properties, whose DB Field Names begin with I_ or P_. Any Extended Project Properties that are Part of Folder Key will be ignored. Optional.

Attachment outputs

Variable	Description
PRJ_KEYSTRING	The Keysting Property of the new Project.

Remarks

DmgBrsCreateProjectFromFolder transforms an existing Folder into a Workflow Project. If you want to create a completely new Project, use DmgBrsCreateProject.

The input attachment variables are similar to those of DmgBrsCreateProject, but there are three important differences in DmgBrsCreateProjectFromFolder. First, a Folder Specification must be provided. Second, the Workflow Cabinet is passed in WF_CABINET, not CABINET. Third, the XPRJ_propertyname variables are optional and are used only to update editable Properties.

The output attachment variable plus the input attachment variable WF_CABINET form a Project Specification.

Examples

Suppose we want to transform a Folder into a Project, and then check out the new Project all in one operation. Assume Claim Number is the only Property that is Part of Folder Key. The input attachment variables and the rules list to do this might look like the following.

```
CABINET = Claims List
FLD_ID = 99752
WF_CABINET = Claims
PRJ_DESCRIPTION = hail storm damage
PRJ_DUEDATE = 07/31/2005
```

```
PRJ_PRIORITY = 5
XPRJ_Claim$20Remarks = Damage appears extensive.
```

Compare the above inputs to those in the example for DmgBrsCreateProject. Notice in the above that we did not set Claim Number, which is Part of Folder Key. When transforming a Folder into a Project, we cannot change its keys. However, we did supply a value for Claim Remarks, which is not Part of Folder Key. The Folder and Project will be updated with that value.

Now let's take a look at the rules list. Even though the parameters for DmgBruCopyAttachmentVar may line wrap here, they should not be allowed to do so in an actual rules list.

```
[ ReqType:CREATE_PROJ ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCreateProjectFromFolder
function = dmg_brw32->DmgBruCopyAttachmentVar, runf, output, WF_CABINET, output, CABINET
function = dmg_brw32->DmgBrsCheckOutProject
```

On the above rules list, notice that we have used DmgBruCopyAttachmentVar to copy the value of WF_CABINET into CABINET, over-writing the previous value of CABINET. This is because DmgBrsCheckOutProject requires a Project Specification, where CABINET must be the name of the Workflow Cabinet.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentProject DmgBrsCheckOutProject
See also	DmgBrsCheckOutProject DmgBrsCreateProject DmgBrsLogPropertySets DmgBrsPresentProject

DMGBRSDELETEDIARY

DmgBrsDeleteDiary deletes a Diary.

Usage

```
function = dmg_brw32->DmgBrsDeleteDiary
```

Attachment inputs

Variable	Description
DIA_KEYSTRING	The Keysting Property of the Diary to delete. Required.

Attachment outputs

Variable	Description
DIA_TITLE	The Title Property of the deleted Diary.
DIA_DESCRIPTION	The Description Property of the deleted Diary.
DIA_PRIORITY	The Priority Property of the deleted Diary.
DIA_CREATEDON	The CreatedOn Property of the deleted Diary.
DIA_STARTON	The StartOn Property of the deleted Diary.
DIA_DUEON	The DueOn Property of the deleted Diary.
DIA_COMPLETEDON	The CompletedOn Property of the deleted Diary.
DIA_LASTMODIFIED ON	The LastModifiedOn Property of the deleted Diary.
DIA_NEXTALERTTI ME	The NextAlertTime Property of the deleted Diary.
DIA_ASSIGNEDBY	The AssignedBy Property of the deleted Diary.
DIA_OWNER	The Owner Property of the deleted Diary.
DIA_REFERENCE	The Reference Property of the deleted Diary.
DIA_STATUS	The Status Property of the deleted Diary.
DIA_ID	The ID Property of the deleted Diary.
DIA_KEYSTRING	The Keysting Property of the deleted Diary.

Remarks

DmgBrsDeleteDiary deletes a Diary and returns its Properties. With the Properties, it would be possible to confirm to the user that the Diary with Title So and So and Description Thus and Such was just deleted.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions	None.
May be followed by	DmgBrsQueryDiaries
See also	DmgBrsCreateDiary DmgBrsQueryDiaries DmgBrsUpdateDiary

DMGBRSDELETEDOCUMENT

DmgBrsDeleteDocument deletes all versions of a Document in a Cabinet.

Usage

```
function = dmg_brw32->DmgBrsDeleteDocument
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to delete. Required.
PS_LIST	A Property Set List to limit any Basic or Extended Document Properties to return for the deleted Document. Uses PS_DOC and PS_XDOC Property Sets. Optional. If absent, DmgBrsDeleteDocument will use a Property Set List called "DeleteDocumentDefault" if one exists.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the deleted Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the deleted Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the deleted Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the deleted Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the deleted Document. Exists only if IS_WORKFLOW contains 1.
DOC_propertyname	Variables containing any other Basic Document Properties of the deleted Document listed in the PS_DOC Property Set of the Property Set List, if any, or all these Properties if no Property Set List was used.
XDOC_propertyname	Variables containing any Extended Document Properties of the deleted Document listed in the PS_XDOC Property Set of the Property Set List, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name), or all these Properties if no Property Set List was used.

Remarks

DmgBrsDeleteDocument deletes all versions of a Document in a Cabinet. The output attachment variables provide a Folder or Project Specification for the Document to make it easy to return to the deleted Document's location.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions	None.
May be followed by	DmgBrsPresentFolder DmgBrsPresentProject
See also	DmgBrsLogPropertySets DmgBrsPresentFolder DmgBrsPresentProject DmgBrsRemoveDocumentAsRendition

DMGBRSDELETEFOLDER

DmgBrsDeleteFolder deletes a Folder and, optionally, all of its contents.

Usage

```
function = dmg_brw32->DmgBrsDeleteFolder
```

Attachment inputs

Variable	Description
A Folder Specification	The Folder to delete. Required.
DELETE_CHILDREN	Set this to 1 to cause all of the Folder descendents of this Folder to be deleted, else set it to 0. Optional. Default is the INI option [DeleteFolder] DeleteChildren.
DELETE_DOCUMENTS	Set this to 1 to cause all of the Documents in this Folder and in any descendent Folders being deleted to be deleted, else set it to 0. This rule will fail when this variable is 0 and a Folder contains Documents because a Folder that contains Documents cannot be deleted. Optional. Default is the INI option [DeleteFolder] DeleteDocuments

Attachment outputs

Variable	Description
STARTAT	A variable that will cause the contents of the parent of the deleted Folder to be found by the rule DmgBrsQuery.

Remarks

DmgBrsDeleteFolder deletes a Folder and, optionally, all of its contents. The contents of a Folder includes all of its Folder descendents (all of its Folder children, and all of their Folder children, etc.) and all of the Documents in the Folder and its Folder descendents.

If the Folder to delete cannot be found, this rule succeeds.

A Folder cannot be deleted if it contains Documents, and DmgBrsDeleteFolder will fail if you attempt to delete any such Folder. DELETE_DOCUMENTS is provided as a safeguard to allow you to prevent deleting a Folder that you wouldn't delete if you knew it contained Documents. If DeleteDocuments is ON, then DmgBrsDeleteFolder deletes all of the Documents in all of the affected Folders before it deletes any Folder.

For a Cabinet with one Level, DeleteChildren has no effect since no Folder in such a Cabinet has Folder children.

For a Cabinet with more than one Level, there are two recommended ways to approach Folder deletion. The first is to set DeleteChildren ON and delete a Folder at Level one. The second is to set DeleteChildren ON and delete at Folder that has at least one sibling Folder at any Level higher than one. With either of these procedures, no inaccessible Folders or Documents will be left after the deletion. "Inaccessible" means "not findable with a Documanager search".

However, DmgBrsDeleteFolder does not prevent you from deleting a Folder in a multi-Level Cabinet in such a way that inaccessible Folders and Documents are left behind. For example, if you delete a Folder that is the only child Folder of another Folder, that other Folder will become inaccessible. Also, if you delete a Folder that has Folder descendents without deleting the descendents, those descendents will become inaccessible.

The output attachment variable STARTAT will cause the rule DmgBrsQuery to find the contents of the parent of the deleted Folder. If the deleted Folder was at Level one of a Cabinet, then the other Folders at Level one, if any, would be found. If the deleted Folder was at some

other Level of a Cabinet, then the sibling Folders (and, optionally, Documents) of the deleted Folder would be found. Of course, if the deletion left inaccessible Folders, the STARTAT variable might not produce the desired results.

Examples Here is rules list to delete a Folder and then return a list of the Folder's siblings .

```
[ ReqType:DEL_FLD ]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmgb32->DmgbCopyAttachment
  function = dmgb32->DmgbValidateSession
  function = dmgb32->DmgbDeleteFolder
  function = dmgb32->DmgbQuery
```

The input attachments for a transaction using the above rules list might look like the following. Assume that [DeleteFolder] DeleteChildren and [DeleteFolder] DeleteDocuments are both ON.

```
CABINET = MyCab
FLD_LEVELNUMBER = 1
FLD_ID = 27
MAXHITS = 10
```

DmgbDeleteFolder requires the first three input attachment variables. DmgbQuery requires CABINET and uses MAXHITS.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [DeleteFolder]

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgbQuery

See also DmgbPresentFolder
DmgbQuery

DMGBRSENCODEURLFIELD

DmgBrsEncodeURLField URL encodes the values of designated attachment variables.

Usage `function = dmg_brw32->DmgBrsEncodeURLField,attachvarname...`

Parameters

Parameter	Description
attachvarname	The name of an attachment variable or a RecordSet Wildcard to URL encode. There may be any number of these. Case-independent.

Attachment inputs None.

Attachment outputs None.

Remarks

DmgBrsEncodeURLField URL encodes the values of attachment variables designated in its parameters. URL encoding makes it possible to use the value of the variable in a URL on a web page.

There may be any number of parameters, but as a practical matter, it may be preferable to list the rule repeatedly if there is a large number of variables to be encoded rather than list all of the variables at once.

If DmgBrsEncodeURLField cannot find a variable, it skips it. The rule never fails.

DmgBrsEncodeURLField does the encoding on its reverse run, and processes variables on the output queue.

DmgBrsEncodeURLField encodes any character except alphabetic characters, digits, and the following 11 characters:

`$-_.+!*'(),`

Note in particular that it will encode forward and backward slashes, spaces, and ampersands.

There are a few variables that the Bridge automatically URL encodes (e.g., QUERY_STEM, output by DmgBrsQuery). They are noted where they occur in the descriptions of individual rules. You must not URL encode any variable that has already been URL encoded. Do not encode the same variable twice, and do not encode a variable that the Bridge has already encoded. To do so will result in incorrect encoding.

Examples

In this example we are running the rule DmgBrsQuery, and we want to use the Cabinet and the KeyString Folder Property of each hit to construct a URL to run a rules list with DmgBrsPresentFolder on it. Our Cabinet has two levels.

Suppose that our Cabinet name has characters in it that need to be URL encoded. KeyStrings typically have such characters in them. We will encode the KeyStrings directly because we don't plan to use them for anything other than the URLs. But we'll make a copy of the CABINET variable with DmgBruCopyAttachmentVar before encoding it because we plan to display the name of the Cabinet on a web page unencoded.

Even though the parameters for rules may line wrap here, they should not be allowed to do so in an actual rules list.

```
[ ReqType:QUERY ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrSEncodeURLField,FHITS_PL0_*.KEYSTRING
, FHITS_PL1_*.KEYSTRING
function = dmg_brw32->DmgBrSEncodeURLField,CABINET_URL
function = dmg_brw32-
>DmgBruCopyAttachmentVar,runnr,output,cabinet,output,cabinet
_url
function = dmg_brw32->DmgBrSCopyAttachment
function = dmg_brw32->DmgBrSValidateSession
function = dmg_brw32->DmgBrSQuery
```

Notice in the above rules list that we're using DmgBrSEncodeURLField to encode the Folder Hits from DmgBrSQuery for parent Level 0 and parent Level 1 (see DmgBrSQuery for details). This is because the same rules list is used to process Folder Hits for either Level. Even though Folder Hits will occur for only one of these Levels on any given run of the rules list, it's still fine to use DmgBrSEncodeURLField to encode KeyStrings for both levels. DmgBrSEncodeURLField will simply ignore variables it doesn't find.

Here is what some of the relevant attachment variables might look like immediately after DmgBrSQuery runs.

```
CABINET = Policies and Endorsements
FHITS_PL0_1.KEYSTRING = dbo.Policy.Policy_ID = 9478225
FHITS_PL0_2.KEYSTRING = dbo.Policy.Policy_ID = 3794403
```

And here is what they look like after DmgBruCopyAttachmentVar and DmgBrSEncodeURLField run. Note that we now have an additional variable, created by DmgBruCopyAttachmentVar.

```
CABINET = Policies and Endorsements
CABINET_URL = Policies%20and%20Endorsements
FHITS_PL0_1.KEYSTRING = dbo.Policy.Policy_ID%20%3D%209478225
FHITS_PL0_2.KEYSTRING = dbo.Policy.Policy_ID%20%3D%203794403
```

ReqType changes	None.
INI options	None.
Validations	None.
Platform restrictions	None.
See also	DmgBruCopyAttachmentVar

DMGBRSFORWARDPROJECT

DmgBrsForwardProject forwards a Project to another member of the Project Team.

Usage `function = dmg_brw32->DmgBrsForwardProject`

Attachment inputs

Variable	Description
A Project Specification	The Project to be forwarded. Required.
TEAM_MEMBER_NAME	The name of the Team member, as returned by DmgBrsPresentProject, to whom the Project will be forwarded. Required.

Attachment outputs

None.

Remarks

DmgBrsForwardProject forwards a Project to another member of the Project Team. Team member names can be obtained with DmgBrsPresentProject.

If the Project is checked out, then only the user who has it checked out can forward it, and after it is forwarded, it will be checked out to the recipient. If the Project is not checked out, then anyone on the Team for the Project can forward it.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

It may be logical to return the user to their Task lists after they forward a Project.

DmgBrsGetTasks

See also

DmgBrsPresentProject

DmgBrsGetTasks

DMGBRSGETTASKQUEUE

DmgBrsGetTaskQueue returns the list of Projects waiting at a particular Task for the current user.

Usage

```
function = dmg_brw32->DmgBrsGetTaskQueue
```

Attachment inputs

Variable	Description
CABINET	The name of the Workflow Cabinet for a Task as returned by DmgBrsGetTasks or DmgBrsListWorkflows. Required.
TSK_FILTER	A Task Filter Property (weakly encrypted) exactly as returned by DmgBrsGetTasks. This determines which Task's Projects will be returned. Required.
AUTO_FILTER	May contain the defined value "CHECKED_OUT", case insensitive. If that defined value is set, then only those Projects in the Task that are checked out to the current user will be returned. If this variable does not contain the defined value or is missing, then the Projects list is not filtered. Optional. Default is the INI option [GetTaskQueue] AutoFilter.
PS_LIST	The name of a Property Set List to limit what Properties to return in each Project record. Uses a PS_XPRJ Property Set. If this variable is missing or empty, DmgBrsGetTaskQueue will automatically look for a Property Set List whose name is "GetTaskQueueDefault". Optional.
PROJECTS_INDEX	The zero-based index to start returning Projects. Optional. Default is 0.
MAXPROJECTS	The maximum number of Projects to return in this transaction. Limited by the INI option [GetTaskQueue] MaxProjects. Optional. Default is the INI option [GetTaskQueue] MaxProjects.
SHOW_DOCS_IND	If this is 1, then each Project record will include DOCS_IND to indicate whether the Project contains Documents. This has a performance cost. 0 means no DOCS_IND. Optional. Default is the INI option [DocsIndicator] ShowDocsIndicator.
SHOW_NOTES_IND	If this is 1, then each Project record will include NOTES_IND to indicate whether the Project contains Notes. 0 means no NOTES_IND. Optional. Default is the INI option [Notes] ShowNotesIndicator.

Attachment outputs

Variable	Description
TSK_INSTRUCTIONS	The Instructions Property for the Task.
TSK_MANAGER	The Manager Property for the Task.

TSK_DESCRIPTION	The Description Property for the Task.
TSK_CABINET	The Cabinet Property for the Task.
TSK_FILTER	The Filter Property for the Task. This is weakly encrypted.
TSK_WORKFLOW	The Workflow Property for the Task.
TSK_WORKFLOW_DESCRIPTION	The WorkflowDescription Property for the Task.
TSK_ID	The Id Property for the Task.
TSK_COUNT	The Count Property for the Task.
PROJECTS_EXIST	Contains 1 if there are Projects, else contains 0.
PROJECTS	The number of Projects. Exists only if PROJECTS_EXIST contains 1.
PROJECTSx.field	A record set describing the Projects that were found, where x is the record number and field is one of the following fields. Exists only if PROJECTS_EXIST contains 1.
.XPRJ_propertyname	For Project x, a set of variables containing the values of the Extended Project Properties requested in the PS_XPRJ Property set, if any, on the Property Set List that was input, where propertyname is the Bridge-encoded property name, or all these Properties if no Property Set List was used.
.DOCS_IND	Docs indicator for Project x. Contains 1 if there are Documents in this Project, else contains 0. Exists only when SHOW_DOCS_IND is 1.
.NOTES_IND	Notes indicator for Project x. Contains 1 if there are Notes for this Project, else contains 0. Exists only when SHOW_NOTES_IND is 1.
.PRJ_ID	The Id Property of Project x.
.PRJ_KEYSTRING	The Keysting Property of Project x.
.PRJ_TASKTYPE	The TaskType Property of Project x, in numeric form.
.PRJ_TASKTYPE_TEXT	The TaskType Property of Project x, in text form, where the text matches the enum DmgTaskTypes in dmg_api_types.h.
.PRJ_LABEL	The Label Property of Project x.
.PRJ_standardextendedproperty	A set of variables containing the values of the 19 Standard Extended Project Properties described in the Introduction.

MOREPROJECTS	Contains 1 if there are Projects after the ones that have been returned in this transaction, else contains 0
PREVPROJECTS	Contains 1 if there are Projects previous to the ones that have been returned in this transaction, else contains 0
PROJECTS_INDEX_NEXT	The index of the first Project in the next batch of Projects. Does not exist if MOREPROJECTS is 0.
PROJECTS_INDEX_PREV	The index of the first Project in the previous batch of Projects. Does not exist if PREVPROJECTS is 0.
REACHED_FLD_FETCH_LIMIT	Exists only if MOREPROJECTS contains 0. Contains 1 if MOREPROJECTS is 0 because the Folder Fetch Limit has been reached, else contains 0. The Folder Fetch Limit is a system option that may be set with the Documange Administrator.

Remarks DmgBrGetTaskQueue returns the list of Projects waiting at a particular Task for the current user. The typical sequence of transactions is to get a list of Tasks with a transaction that uses DmgBrGetTasks, select a Task and get a list of Projects for the Task with a transaction that uses DmgBrGetTaskQueue, and then select a Project and take action on it with a transaction that uses such rules as DmgBrPresentProject, DmgBrCheckOutProject, DmgBrSuspendProject, or DmgBrResumeProject.

While provision is made for paging through Projects MaxProjects at a time, the Project list is expected to be more volatile than other types of lists. For this reason, paging by index might cause items to be skipped or repeated more often than it would for paging through other types of items. In some applications, it may be sufficient to simply present the user with the first MaxProjects Projects and allow them to select from those, without providing a way to ask for more Projects.

Examples Suppose that DmgBrGetTaskQueue was called based on a Task returned by DmgBrGetTasks. For one of the Projects returned in the record set, here's how you might set up a link to DmgBrResumeProject, which would resume the suspended Project. The following is a snippet from an HTML template that contains an A tag based on the outputs of DmgBrGetTaskQueue. BridgeURL comes from DmgBrValidateSession.

```
<A HREF=#BRIDGEURL, #&REQTYPE=ResumeP&Cabinet=#
TSK_CABINET, #&PRJ_KEYSTRING =#PRJ_KEYSTRING, #>Resume</A>
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [GetTaskQueue]

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrCheckOutProject
DmgBrGetTasks
DmgBrLogPropertySets
DmgBrPresentProject
DmgBrQueryAllProjectsEx

DmgBrsQueryProjects

DmgBrsResumeProject

DmgBrsSuspendProject

DMGBRSGETTASKS

DmgBrsGetTasks returns the list of tasks for the logged-in user.

Usage

```
function = dmg_brw32->DmgBrsGetTasks
```

Attachment inputs

Variable	Description
CABINET	If this exists and contains a valid Workflow Cabinet name, then the outputs are limited to Tasks associated with this Workflow. Otherwise, the outputs include Tasks for all Workflows. Optional. Default is to output Tasks for all Workflows.
PROJECT_STATUS	This variable can be used to control what Tasks are output. It should contain exactly one of the following three values: PENDING -- Report only those Tasks with pending Projects. SUSPENDED -- Report only those Tasks with suspended Projects. ALL -- Report all Tasks. If the value of the variable isn't recognized as one of these three, it is treated as if it were ALL. The values are case-independent. Optional. Default value when absent is ALL.

Attachment outputs

Variable	Description
TASKS	Contains the total number of Tasks returned. Note that this may be 0 if, for example, PROJECT_STATUS was set to SUSPENDED and there are no Tasks with suspended Projects.
TASKS x . <i>field</i>	A record set describing the Tasks, where x is the record number and <i>field</i> is one of the following fields. The fields are the Basic Properties of a Task. Exists only if TASKS contains a value greater than 0.
.INSTRUCTIONS	The Instructions property of the Task.
.MANAGER	The Manager property of the Task.
.DESCRIPTION	The Description property of the Task.
.CABINET	The Cabinet property of the Task.
.FILTER	The Filter property of the Task. For security, this is weakly encrypted. It may be passed as an input to the variable TSK_FILTER for DmgBrsGetTaskQueue.
.WORKFLOW	The Workflow property of the Task.
.WORKFLOW_DESCRIPTION	The Workflow Description property of the Task.
.ID	The ID property of the Task.

.COUNT	The Count property of the Task.
.PARENT_TASK_CONTAINER_ID	The Parent ID property of the Task. This will be 0 if the task is not in a Container.
.PENDING_COUNT	The Pending Count property of the Task.
.SUSPENDED_COUNT	The SuspendedCount property of the Task.
.DEACTIVATED	The Deactivated property of the Task. This will contain 1 if the Task is deactivated, else it will contain 0.

Remarks DmgBrsGetTasks returns the list of tasks for the logged-in user, including Tasks whose Count property is 0, that is, Tasks for which there are currently no pending or suspended Projects.

Note that this rule has no required attachment inputs. However, you may supply attachment inputs to limit the list of returned Tasks to those for a particular Workflow Cabinet, or to those whose Projects have a particular status.

Use DmgBrsGetTaskQueue to get the list of Projects queued for a particular Task. For DmgBrsGetTaskQueue , pass in a FILTER from one of the Tasks listed by DmgBrsGetTasks.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype.

Platform restrictions None.

See also DmgBrsGetTaskQueue

DMGBRSIDLE

DmgBrsIdle runs on the SAR rules list and performs periodic Bridge housekeeping tasks.

Usage	<code>function = dmgb32->DmgBrsIdle</code>
Attachment inputs	None.
Attachment outputs	None.
Remarks	<p>DmgBrsIdle performs periodic Bridge housekeeping tasks. It should be run only on the SAR rules list.</p> <p>DmgBrsIdle does not need to be preceded by DmgBrsCopyAttachment. This rule does its work on the Termination run of the rules list.</p> <p>DmgBrsIdle does the following:</p> <ul style="list-style-type: none">• If the Bridge is not initialized, it attempts to initialize it.• If dmgb32.ini or dmgb32mime.ini has changed, it attempts to re-initialize the Bridge.• If the Documanager Bridge session's connection to the Documanager Server has been lost, it attempts to re-connect it. If re-connection fails, it attempts to re-initialize the Documanager Bridge. The act of checking the connection also serves to keep the Documanager Bridge's session active with the Documanager Server.• If running on the primary instance of IDS, it purges the Documanager Bridge File Cache and Documanager Bridge Tempfile Cache according to the options set in [FileCache].• It runs any IdleDiagnostic that may have been specified in [DMGDiagnosics] IdleDiagnostic.• If initialization or IdleDiagnostic fails, it puts the Bridge into the mode specified in the INI option [DmgDiagnosics] OnFail_Idle. If OnFail_Idle has not been specified, then it puts the Documanager Bridge into PAUSE mode by default (see <i>DmgBrsCheckPausedServer on page 35</i>).

NOTE: The AutorunInterval for the SAR rules list must have a value less than the Documanager client time-out interval, a value that can be set in the Documanager Administrator application. Otherwise, there is a risk that the Bridge's session may time out before DmgBrsIdle gets a chance to confirm it.

Examples	<p>DmgBrsIdle should be run only on the SAR rules list.</p> <pre>[ReqType:SAR] function = irlw32->IRLPurgeCache function = dmgb32->DmgBrsIdle</pre>
ReqType changes	None.
INI options	<p>[DMGDiagnosics] IdleDiagnostic</p> <p>[DMGDiagnosics] OnFail_Idle</p> <p>[FileCache] MaxSizeInMegabytes</p> <p>[FileCache] MinPurgeAgeInMinutes</p>

[FileCache] MaxPurgeAgeInMinutes

Validations Reqtype

Platform restrictions None.

See also DmgBrsInit
DmgBrsCheckPausedServer
DmgBrsRunDocumanageDiagnostic

DMGBRSINIT

DmgBrsInit runs on the INI rules list and initializes and shuts down the Bridge to Documanager.

Usage `function = dmg_brw32->DmgBrsInit`

Attachment inputs None.

Attachment outputs None.

Remarks DmgBrsInit initializes the Bridge on the Initialization run of the INI rules list, and shuts it down on the Termination run of the INI rules list. DmgBrsInit does not need to be preceded by DmgBrsCopyAttachment.

Initialization and shutdown are logged whether logging is on or not. A few messages indicating Initialization and shutdown are posted to the IDS console.

Initialization may fail because of invalid INI options, an outdated `dmg_api` library or Documanager, the inability to log in to Documanager, and various other conditions. If it fails, messages are logged and posted to the IDS console. You do not necessarily have to shut down IDS to correct the problem. After you have corrected the problem, the next run of `DmgBrsIdle` will again attempt to initialize the Bridge.

During initialization, DmgBrsInit does the following:

- It initializes logging and the INI options, and validates the INI options.
- It logs INI files according to the INI option [Logging] `LogINIFilesAtStartUp`.
- It validates the version of the `dmg_api` library.
- It initializes the Bridge File Cache and Bridge Tempfile Cache according to the INI options configured in [FileCache] and [TempCache].
- It logs into the Documanager system configured in the INI options [Documanager] using the Bridge Credentials configured in the INI options [BridgeCredentials] to establish the Bridge's Session, which will be maintained until shutdown.
- It validates the version of Documanager and runs the StartupDiagnostic, if any, configured in [DMGDiagnosics] `StartupDiagnostic`.
- Using the INI option [PropertySets] `PropertySetFile`, it finds and initializes any Property Set Lists.
- It initializes the file decompression code and, if needed, the file compression code. Windows only.
- It initializes Shuttle. Windows only.
- It initializes the Repository Manager. Windows only.
- If initialization or StartupDiagnostic fails, it puts the Bridge into the mode specified in the INI option [DmgDiagnosics] `OnFail_Startup`. If `OnFail_Startup` has not been specified, then it puts the Bridge into PAUSE mode by default (see *DmgBrsCheckPausedServer on page 35*).

Examples DmgBrsInit runs only on the INI rules list.

```
[ ReqType:INI ]  
function = irlw32->IRLInit  
function = dprw32->DPRInit
```

```

function = DSICoRul->Init
function = DSICoRul->Invoke,DocuCorp_IDS_DPRCo.DPR-
>DPRCoLoginInit
function = Tpdw32->TPDInitRule
function = dmg_brw32->DmgBrsInit

```

ReqType changes None.

INI options All INI options are initialized and validated. The following INI options or sections are used specifically in DmgBrsInit.

```

[BridgeProfile]
[BridgeCredentials]
[CDPlugins]
[DMGDiagnositics]
[Documanage]
[FileCache]
[InisToLog]
[Logging] LogINIFilesAtStartUp
[Logging] DeleteLogAtStartUp
[PropertySets] PropertySetFile
[TempCache]

```

Validations None.

Platform restrictions None, although some components that DmgBrsInit initializes are not available on all platforms. These include the file compressor and decompressor, Shuttle, and Repository Manager.

See also DmgBrsIdle
DmgBrsCheckPausedServer

DMGBRSListCABINETS

DmgBrsListCabinets lists the Cabinets available on the Documanage System.

Usage `function = dmg_brw32->DmgBrsListCabinets`

Attachment inputs None.

Attachment outputs

Variable	Description
CABINETS	The number of Cabinets available on the Documanage System.
CABINETSx.field	A record set describing the Cabinets that were found, where x is the record number and field is one of the following fields. Exists only if CABINETS is greater than 0.
.NAME	The Label Property (Name) of the Cabinet.
.XDP_ENABLED	The XDPEnabled Property of the Cabinet. Contains 1 if enabled, else 0.
.CASE_SENSITIVE	The CaseSensitive Property of the Cabinet. Contains 1 if case sensitive, else 0.
.DOC_TABLE	The DocTable Property of the Cabinet.
.LEVELS	The Levels Property of the Cabinet, that is, the number of Levels.
DMG_ENABLENULLVALUES	Contains 1 if the option [NullValues] EnableNullValues in dmg_brs.ini is ON, else contains 0.
DMG_NULLVALUE	Exists only if DMG_ENABLENULLVALUES contains 1. Contains the value of the option [NullValues] NullValue in dmg_brs.ini.

Remarks DmgBrsListCabinets lists the Cabinets available on the Documanage System. It lists only those Cabinets available to the current User. To list all of the Cabinets on the Documanage System, the user must have access to all of them and the Bridge Account must also have access to all of them.

The outputs DMG_ENABLENULLVALUES and DMG_NULLVALUE are provided for convenience in dynamically configuring a Bridge application. See the section “NULL Values as Inputs and Outputs” in the General Reference for more information.

DmgBrsListCabinets lists ordinary Cabinets. To list Workflow Cabinets, use DmgBrsListWorkflows. To get more information about a particular ordinary or Workflow Cabinet, use DmgBrsPresentCabinetEx.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [NullValues].

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsListWorkflows

See also DmgBrsPresentCabinetEx
DmgBrsListWorkflows
DmgBrsListCategories
DmgBrsListUsers
DmgBrsListGroups

DMGBRSLISTCATEGORIES

DmgBrsListCategories lists the categories available on the Documanage System.

Usage `function = dmg_brw32->DmgBrsListCategories`

Attachment inputs None.

Attachment outputs

Variable	Description
CATEGORIES	The number of Categories on the Documanage System.
CATEGORIESx.field Id	A record set containing the names of the Categories, where x is the record number and field is the following field.
.NAME	The Name Property of the Category.

Remarks DmgBrsListCategories lists the Categories available on the Documanage System.
To get more information about a particular Category, use DmgBrsPresentCategoryEx.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsPresentCategoryEx
DmgBrsListCabinets
DmgBrsListUsers
DmgBrsListGroups

DMGBRSLISTGROUPS

DmgBrsListGroup lists the Groups on the Documanager system, or the Groups for a particular User on the Documanager system.

Usage

```
function = dmg_brw32->DmgBrsListGroup
```

Attachment inputs

Variable	Description
USER_NAME	The Name Property of a User, as returned by DmgBrsListUsers. Optional. Default is empty. If this exists, the list of Groups returned is limited to those for this User. If the Name of the User is invalid, the rule fails.
OBn_GROUPS	For sorting the list of Groups. Syntax is parallel to that of SQL Builder Orderby. n is the sort level. The sort is done first by the property specified for level 1, then by the property specified for level 2, etc. The value of the variable must be a basic Group property name from dmg_api_props.h, case insensitive. Optional. If these variables are absent, the list of Groups will be ordered however it is returned from the Documanager server. See the description of DmgBrsListUsers for an example. (The only basic property that Groups have at this time is Name. This syntax allows for future expansion.)
OB_GROUPS_O P	For sorting the list of Groups. Syntax is similar to that of SQL Builder Orderby, except the operation applies to all sort levels. The value of the variable may be any word beginning with "ASC" or "DESC", case insensitive. Optional. Ignored if there are no OBn_GROUPS variables. Default is "ASC".

Attachment outputs

Variable	Description
GROUPS	The number of Groups being returned in this transaction.
GROUPSx.fiel d	A record set describing the Groups, where x is the record number and field is the following.
.NAME	The Name Property of the Group.

Remarks

DmgBrsListGroup lists the Groups on the Documanager system, or the Groups for a particular User on the Documanager system.

You may use the optional OB... input variables to control the order of the list of Groups. It may be sorted by any of the basic properties of a Group, and may be presented in ascending or descending order.

To get a list of Users, use DmgBrsListUsers.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

See also DmgBrsListUsers
 DmgBrsListCategories
 DmgBrsListCabinets

DMGBRSLISTUSERS

DmgBrsListUsers lists the users on the Documanage system, or the users for a particular group on the Documanage system.

Usage

```
function = dmg_brw32->DmgBrsListUsers
```

Attachment inputs

Variable	Description
GROUP_NAME	The Name Property of a Group, as returned by DmgBrsListGroup. Optional. Default is empty. If this exists, the list of Users returned is limited to those Users in this Group. If the Name of the Group is invalid, the rule fails.
USERS_INDEX	The zero-based index of where to start in the Users List. Optional. Default is 0.
MAX_USERS	The maximum number of Users to list. Limited by the INI option [ListUsers]MaxUsers. Optional. Default is the INI option [ListUsers]MaxUsers.
OBn_USERS	For sorting the list of Users. Syntax is parallel to that of SQL Builder Orderby. n is the sort level. The sort is done first by the property specified for level 1, then by the property specified for level 2, etc. The value of the variable must be a basic User property name from dmg_api_props.h, case insensitive. Optional. If these variables are absent, the list of Users will be ordered however it is returned from the Documanage server.
OB_USERS_OP	For sorting the list of Users. Syntax is similar to that of SQL Builder Orderby, except the operation applies to all sort levels. The value of the variable may be any word beginning with "ASC" or "DESC", case insensitive. Optional. Ignored if there are no OBn_USERS variables. Default is "ASC".

Attachment outputs

Variable	Description
USERS	The number of Users being returned in this transaction.
USERSx.field	A record set describing the Users, where x is the record number and field is one of the following fields.
.NAME	The Name Property of the User.
.DISPLAY_NAME	The DisplayName Property of the User.
.QUALIFIED_NAME	The QualifiedName Property of the User.
.DOMAIN	The Domain Property of the User.
MOREUSERS	Contains 1 if there are more Users after the ones returned in this transaction, else 0.
USERS_INDEX_NEXT	The index of the next batch of Users.

Variable	Description
PREVUSERS	Contains 1 if there are Users previous to the ones returned in this transaction, else 0.
USERS_INDEX_PREV	The index of the previous batch of Users.

Remarks DmgBrsListUsers lists the users on the Documanager system, or the users for a particular group on the Documanager system.

Navigate the list of Users forward or backward by passing in the value of USERS_INDEX_NEXT or of USERS_INDEX_PREV from one transaction for USERS_INDEX in the next transaction.

You may use the optional OB... input variables to control the order of the list of Users. It may be sorted by any of the basic properties of a User, and may be presented in ascending or descending order.

To get a list of groups, use DmgBrsListGroup.

Examples This example shows input attachment variables for DmgBrsListUsers. In the example, we want to page through a large list of users 20 users at a time. Assume that the value of the INI option [ListUsers]MaxUsers is 30. We set the input attachment variable MAX_USERS to 20 so that we get 20 users at a time. Assume that two batches of 20 users have been returned in previous calls to DmgBrsListUsers. On this call, we set the input attachment variable USERS_INDEX to the value that was returned in the output attachment variable USERS_INDEX_NEXT in the previous call, which was 40. In the example, we want to order the list first by Domain, then by DisplayName, in ascending order. Since ascending is the default order, we do not need to list OB_USERS_OP, but it is shown here to illustrate its use.

MAX_USERS = 20

USERS_INDEX = 40

OB1_USERS = Domain

OB2_USERS = DisplayName

OB_USERS_OP = Ascending

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [ListUsers]

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsListGroup
DmgBrsListCategories
DmgBrsListCabinets

DMGBRSLISTWORKFLOWS

DmgBrsListWorkflows lists the Workflow Cabinets available on the Documange System.

Usage

```
function = dmg_brw32->DmgBrsListWorkflows
```

Attachment inputs

None.

Attachment outputs

Variable	Description
WORKFLOWS	The number of Workflow Cabinets available on the Documange System. This may be zero.
WORKFLOWSx.field	A record set describing the Workflow Cabinets that were found, where x is the record number and field is one of the following fields. Exists only if WORKFLOWS is greater than 0.
.CABINET	The PowerCabinet Property (Name) of the Workflow Cabinet.
.WORK_TABLE	The WorkTable Property of the Workflow Cabinet.
.WORKFLOW	The Workflow Property of the Workflow Cabinet.
.DESCRIPTION	The Description Property of the Workflow Cabinet.

DmgBrsListWorkflows lists the Workflow Cabinets available on the Documange System. It lists only those Workflow Cabinets available to the current User. To list all of the Workflow Cabinets on the Documange System, the User must have access to all of them and the Bridge Account must also have access to all of them.

To get more information about a particular Workflow Cabinet, use DmgBrsPresentCabinetEx.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

[QueryAllProjects]

[DocsIndicator] ShowDocsIndicator

[Notes] ShowNotesIndicator

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

DmgBrsListCabinets

See also

DmgBrsPresentCabinetEx

DmgBrsListCabinets

DmgBrsListCategories

DmgBrsListUsers

DmgBrsListGroups

DMGBRSLOCKANNOTATIONS

DmgBrsLockAnnotations locks the Annotations of a Document to the current user so they can be updated.

Usage

```
function = dmg_brw32->DmgBrsLockAnnotations
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to lock Annotations. Required.

Attachment outputs

Variable	Description
RESULT_LOCKANNOTATIONS	A result code. Contains 0 if locking was successful. Contains 1 if the Annotations for the requested Document are already locked by this user. Contains 2 if the Annotations for the requested Document are locked by another user or the Document is checked out to another user.
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsLockAnnotations locks the Annotations of a Document to the current user so they can be updated with DmgBrsUpdateAnnotations. When the Annotations are locked, no other user can check the Document out, lock the Annotations, or modify the Annotations. Once Annotations are locked, they will remain locked until the user who locked them runs the rule DmgBrsUnlockAnnotations.

Checking out a Document also keeps other users from modifying Annotations, and may be used as an alternative to DmgBrsLockAnnotations when other changes are to be made to the Document.

Typically, you would lock Annotations before getting them (with DmgBrsReturnAnnotations). This assures that the copy to be modified is the current copy.

Examples

The following rules list locks Annotations for a Document and returns the Annotations file in one transaction.

The only inputs required would be a Current Document Specification

```
[ReqType:LOCKANNOTS]  
function = atcw32->ATCLogTransaction
```

```

function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsLockAnnotations
function = ATCW32->ATCAppend2Attachment
function = dmg_brw32->DmgBrsCacheAnnotationsFile
function = dmg_brw32->DmgBrsReturnAnnotations
[Attach: LOCKANNOTS]
DOC_MAJORVERSION = -1

```

Note that DmgBrsLockAnnotations requires a Current Document Specification, which does not include a Document version. However, DmgBrsCacheAnnotationsFile requires a Version Document Specification, which does include a Document version. Since a value of -1 for DOC_MAJORVERSION can be used in a Version Document Specification to indicate “current version”, we add that to the attachment. In this case, we used ATCAppend2Attachment to add it, but we could also have simply passed it in.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsPresentProject DmgBrsReturnAnnotations
See also	DmgBrsCheckOutDocument DmgBrsReturnAnnotations DmgBrsUnlockAnnotations DmgBrsUpdateAnnotations

DMGBRSLOGIN

DmgBrsLogin logs in an individual user.

Usage

```
function = dmg_brw32->DmgBrsLogin
```

Attachment inputs

Variable	Description
BRSLOGIN_NAME	The name of the user logging in. DmgBrsLogin will fail if this is empty. Required.
BRSLOGIN_DOMAIN	The domain of the user logging in. Some environments require a domain and some do not.
BRSLOGIN_PASSWORD_ENC	The scrambled password of the user logging in. See the scrambling algorithm below. You may use this when the INI option [BridgeURL] SecureServerUsage is OFF to prevent a password going over the network in clear text. This is weak security, and we recommend using a secure server for login. Required if BRSLOGIN_PASSWORD is not supplied.
BRSLOGIN_PASSWORD	The password of the user logging in. Required if BRSLOGIN_PASSWORD_ENC is not supplied.
REMOTE_ADDR	A standard Web server environment variable that contains the address of the user. Required.
BRS_DESTINATION	A URL for redirection after login. Optional.

Attachment outputs

Variable	Description
SESS_SOURCE	The source of user credentials. It will be Login in this case.
SESS_LOGINTIME	The time of login.
SESS_ROUTERIP	The IP address of the Documanager Router, as configured in the INI option [Documanager] Router.
SESS_USERNAME	The name of the user.
SESS_USERDOMAIN	The domain of the user.
SESS_GROUPACCESS	The user's groups, separated by commas.
SESS_REMOTEADDR	The value of REMOTE_ADDR.
SESS_VALID	Contains 1 to indicate a successful login, else 0. The rule will also fail if login is unsuccessful.
BRS_DESTINATION	If BRS_DESTINATION was not input, then it is output. It will be set to the basic Bridge URL plus the Reqtype configured in the INI option [ReqTypes] DestinationAfterLogin.

BRSLOGIN_ERRORPROMPT	Exists if login fails. Contains a number telling the reason for the failure. 4 - The user name or password is not valid 5 - Authentication failed.
BRSLOGIN_ERRORCODE	Exists if login fails. An error code associated with the failure.
BRSLOGIN_ERRORMSG	Exists if login fails. An error message corresponding to BRSLOGIN_ERRORPROMPT. 4 - "The name or password is not valid." 5 - "Authentication failed (error n)", where n is BRSLOGIN_ERRORCODE.

Remarks

DmgBrsLogin works in combination with DmgBrsValidateSession to log in an individual user. It is used when the INI option [UserCredentials] Source is set to Login. This rule is not needed if [UserCredentials] Source is set to any other value.

DmgBrsLogin logs the user described in the input attachment variables into Documanager to validate the user's credentials. If this login fails, then the rule fails. After a successful login, DmgBrsLogin immediately logs the user out again. DmgBrsValidateSession must always be on the rules list that runs DmgBrsLogin. On its reverse run, DmgBrsValidateSession will incorporate the SESS_... output attachment variables from DmgBrsLogin, except SESS_VALID, into the cookie text. In subsequent transactions, DmgBrsValidateSession will use the data from the cookie to identify the user, and will adjust the Bridge Session, established by DmgBrsInit on the basis of the options in [BridgeCredentials], to impersonate (proxy for) the user during the transaction.

DmgBrsLogin always deletes BRSLOGIN_PASSWORD_ENC and BRSLOGIN_PASSWORD from the attachment.

When DmgBrsLogin fails, its behavior is different from other rules. It will set the Reqtype to that configured in the INI option [ReqTypes] LoginUIDisplay, giving the Bridge application developer the opportunity to return the user to a login screen. It will also set the BRSLOGIN_ERROR... attachment variables. These variables may be used to construct error information for the user.

If login is successful, DmgBrsLogin will change the reqtype to that configured in [ReqTypes] Redirect.

DmgBrsValidateSession will create an appropriate BRS_DESTINATION during an Ad Hoc login. See DmgBrsValidateSession for more information. A Bridge application may also optionally construct a URL and pass it to DmgBrsLogin in BRS_DESTINATION to direct the user to any rules list with any desired input attachment variables after login. If BRS_DESTINATION was not passed in, DmgBrsLogin will output one that contains a Bridge URL with the reqtype configured in the INI option [ReqTypes] DestinationAfterLogin. The use of BRS_DESTINATION or the INI option [ReqTypes] DestinationAfterLogin allows the Bridge application to direct the user to any arbitrary rules list after login.

The DmgBrsValidateSession input attachment variables CONFIG, SESS_COOKIE_RESET, and SESS_USERDATA are sometimes passed in during the login transaction. See DmgBrsValidateSession for more information on these variables.

If the INI option [UserCredentials] Source is not configured to Login, DmgBrsLogin will still run, but it will simply use the Bridge credentials for login and ignore the user credentials. This is for convenience during Bridge application development or testing. It allows you to have an application that is configured for Login, but to turn Login off temporarily and run the

application without the need to supply valid credentials when the application presents the login screen.

The following is a Javascript function that illustrates the algorithm for scrambling a password for use in BRSLOGIN_PASSWORD_ENC.

```
function ScramblePword(pword)
{
    var newPassword = ""
    var num
    for (var j=0;j< pword.length;j++)
    {
        num = pword.charCodeAt(j)-13
        if(33 > num)
            num = num + 94
        newPassword=newPassword+String.fromCharCode(num)
    }
    return newPassword
}
```

Examples A rules list that runs DmgBrsLogin would typically look like this. Note that we included DmgBrsValidateSession. This assures that the validated outputs from DmgBrsLogin get written to the cookie text.

```
[ ReqType:BRIDGE_LOGIN ]
    function = atcw32->ATCLogTransaction
    function = atcw32->ATCLoadAttachment
    function = atcw32->ATCUnloadAttachment
    function = dmgb32->DmgBrsCopyAttachment
    function = dmgb32->DmgBrsValidateSession
    function = dmgb32->DmgBrsLogin
```

ReqType changes The ReqType configured in the INI option [ReqTypes] LoginUIDisplay if Login fails.
The ReqType configured in the INI option [ReqTypes] Redirect if Login succeeds.
The ReqType configured in the INI option [ReqTypes] Failure if there is an unanticipated failure.

INI options [Documanage] Router
[UserCredentials] Source
[ReqTypes] DestinationAfterLogin
[ReqTypes] LoginUIDisplay
[ReqTypes] Redirect

Validations Reqtype

Platform restrictions None.

See also DmgBrsLogout
DmgBrsValidateSession

DMGBRSLOGOUT

DmgBrsLogout logs out an individual user.

Usage `function = dmg_brw32->DmgBrsLogout`

Attachment inputs None.

Attachment outputs

Variable	Description
SESS_USERNAME	Set to "~~LOGGEDOUT", to indicate that the user is logged out.

Remarks DmgBrsLogout works in combination with DmgBrsValidateSession to log out an individual user. It is used when the INI option [UserCredentials] Source is set to Login, although the rule does not check any INI options. This rule is not needed if [UserCredentials] Source is set to any other value.

DmgBrsLogout logs out a user by setting SESS_USERNAME to a value that indicates the user is logged out. DmgBrsValidateSession will incorporate this value into the user's cookie on its reverse run. The cookie is then in a logged out state. In any subsequent transaction that includes DmgBrsValidateSession, DmgBrsValidateSession will require the user to log in again before proceeding.

For a way to cause users to be effectively logged out when they quit their browser session, see the INI option [ValidateSession] CookiesPersist.

Examples Even though DmgBrsLogout does not require a valid Session to run, you must include DmgBrsValidateSession on its rules list to set the user's cookie to a logged out state.

```
[ReqType:LOGOUT]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsLogout
```

ReqType changes None.

INI options None.

Validations None

Platform restrictions None.

See also DmgBrsLogin
DmgBrsValidateSession

DMGBRSLOGPROPERTYSETS

DmgBrsLogPropertySets logs Property Sets as an aid in configuring the Bridge.

Usage	<code>function = dmg_brw32->DmgBrsLogPropertySets</code>
Attachment inputs	None.
Attachment outputs	None.
Remarks	<p>DmgBrsLogPropertySets logs PS_XFLD Property Sets for all levels of all non-Workflow Cabinets on the Documanager system, PS_XPRJ Property Sets for all Workflow Cabinets, PS_XDOC Property Sets for all Categories for which Extended Document Properties are defined, and a complete PS_DOC Property Set appropriate for the version of Documanager to which the Bridge is connected. It also logs a section called [WorkflowCabinets] which has the Bridge-encoded name of all Workflow Cabinets, and a section for each Cabinet that lists the Bridge-Encoded names for the Cabinet Filter variables for the Cabinet, if any.</p> <p>Even if you do not use Property Sets, DmgBrsLogPropertySets can be very useful because its outputs show the Bridge encoding for all of the Property names on your system. Many rules require or output Bridge-encoded Property names.</p> <p>Property Sets are used by several Bridge Server Rules to limit what Properties are output for things such as Documents. See the section on Property Sets in Configuring the Bridge to Documanager in the Bridge General Reference. DmgBrsLogPropertySets logs Property Sets so that you have a convenient list of these when configuring the Bridge, if you choose to use Property Sets. The PS_XDOC, PS_XFLD and PS_XPRJ Property Sets are also convenient to refer to when configuring or interpreting any variable that has an Extended Property name in its name, such as the XFLD_Lx_propertyname used in DmgBrsCreateFolder.</p> <p>DmgBrsLogPropertySets logs its output without the usual date and time prefix so that you can easily copy and paste what you need into the file you're using to configure Property Sets. The Property Sets are output with the standard name "SetName" which you can edit for your own use. DmgBrsLogPropertySets does all necessary Bridge Encoding. The Property Sets this rule logs contain all Properties, but you can, of course, delete any that you don't need.</p> <p>DmgBrsLogPropertySets does not validate the Session, so DmgBrsValidateSession isn't required on its rules list. The Bridge must be initialized for this rule to run.</p>
ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Reqtype
Platform restrictions	None.

DMGBRSMARKNOTECURRENT

DmgBrsMarkNoteCurrent sets the Deleted Property of a Note to false.

Usage

```
function = dmg_brw32->DmgBrsMarkNoteCurrent
```

Attachment inputs

Variable	Description
A Folder or Project Specification	The Folder or Project that contains the Note. Required.
NOTES_INDEX_TO_CHANGE	The index of the Note to mark current, as returned from DmgBrsPresentFolder or DmgBrsPresentProject. Zero based. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Note was in a Project. Contains 0 if the Note was in a Folder.

Remarks

DmgBrsMarkNoteCurrent sets the Deleted Property of a Note to false. It isn't possible to delete a Note, but its Deleted Property can be changed. It is not an error to mark a Note current when it is already marked current.

Examples

Here are possible input attachment variables for marking a Note current. The Folder that contains the Note is identified with its Id Property, but could also have been identified with its KeyString Property using DEST_FLD_KEYSTRING.

```
CABINET = MyThreeLevelCabinet
FLD_LEVELNUMBER = 2
FLD_ID = 99044
NOTES_INDEX_TO_CHANGE = 2
```

Here is a possible rules list that causes the Folder specified in the input attachment variables to be presented after the Note is changed. In this rules list, we have used

DmgBruCopyAttachmentVar to set up an input attachment variable for DmgBrsPresentFolder or DmgBrsPresentProject that will cause the output attachment variables from either of those rules to list the Notes, starting with the Note that was just changed. Assume that the INI option [Notes] ShowNotes, which tells DmgBrsPresentFolder or DmgBrsPresentProject to return a list of Notes, is ON. Notice that this one rules list can be used with either DmgBrsPresentFolder or DmgBrsPresentProject since those rules are controlled by IS_WORKFLOW. Even though the parameters for DmgBruCopyAttachmentVar may line wrap here, they should not be allowed to do so in an actual rules list.

```
[ReqType:MARK_CUR]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsMarkNoteCurrent
function = dmg_brw32->DmgBruCopyAttachmentVar,runf,output,notes_index_to_change,
output,NOTES_INDEX
function = dmg_brw32->DmgBrsPresentFolder
```

```
function = dmgbrow32->DmgbPresentProject
```

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgbPresentFolder DmgbPresentProject
See also	DmgbMarkNoteDeleted DmgbPresentFolder DmgbPresentProject

DMGBRSMARKNOTEDELETED

DmgBrsMarkNoteDeleted sets the Deleted Property of a Note to true.

Usage

```
function = dmg_brw32->DmgBrsMarkNoteDeleted
```

Attachment inputs

Variable	Description
A Folder or Project Specification	The Folder or Project that contains the Note. Required.
NOTES_INDEX_TO_CHANGE	The index of the Note to mark deleted, as returned from DmgBrsPresentFolder or DmgBrsPresentProject. Zero based. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Note was in a Project. Contains 0 if the Note was in a Folder.

Remarks

DmgBrsMarkNoteDeleted sets the Deleted Property of a Note to true. It isn't possible to delete a Note, but its Deleted Property can be changed. It is not an error to mark a Note deleted when it is already marked deleted.

Examples

See the example for DmgBrsMarkNoteCurrent. It applies equally to DmgBrsMarkNoteDeleted when DmgBrsMarkNoteDeleted is substituted for DmgBrsMarkNoteCurrent in the rules list.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

DmgBrsPresentFolder
DmgBrsPresentProject

See also

DmgBrsMarkNoteCurrent
DmgBrsPresentFolder
DmgBrsPresentProject

DMGBRSMOVEDOCUMENT

DmgBrsMoveDocument moves a Document to a different Folder or Project.

Usage

```
function = dmg_brw32->DmgBrsMoveDocument
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to move. Required.
A Folder or Project Specification, with each variable name prefixed with "DEST_"	The destination of the move, that is, the Folder or Project where the Document will be moved. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document to move was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document to move. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document to move. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document to move. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document to move. Exists only if IS_WORKFLOW contains 1.
DEST_DOCSPEC	A DOCSPEC for the moved Document.
DEST_DOC_ID	The Id Property of the moved Document.
DEST_DOC_MAJORVERSION	The MajorVersion Property of the moved Document.
DEST_DOC_MINORVERSION	The MinorVersion Property of the moved Document.

Remarks

DmgBrsMoveDocument moves a Document to a different Folder or Project. The destination Folder or Project may be in the same Cabinet or a different Cabinet.

Note that the input and output attachment variables taken together provide Document Specifications for the moved Document and Folder or Project Specifications for the Document both before and after it is moved. The variables for the Document after it is moved are all prefixed with "DEST_".

Examples

Here are possible input attachment variables for moving a Document to a Folder. In this example, the variable DEST_FLD_LEVELNUMBER is omitted because the destination Cabinet has only one level. The destination Folder is identified with its Id Property, but could also have been identified with its KeyString Property using DEST_FLD_KEYSTRING.

```

CABINET = MyCabinet
DOC_ID = 722
DEST_CABINET = MyCabinet
DEST_FLD_ID = 22998

```

Here is a possible rules list that causes the Folder from which the Document was moved to be presented after the move. Since DmgBrsMoveDocument outputs the Folder Specification of the location of the Document before it was moved, we will automatically have the Folder Specification needed for DmgBrsPresentFolder.

```

[ReqType:MOVE]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsMoveDocument
  function = dmg_brw32->DmgBrsPresentFolder

```

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder
See also	DmgBrsCopyDocument DmgBrsCreateDocumentShortcut

DMGBRSPRESENTCABINETEX

DmgBrsPresentCabinetEx presents information about a particular Cabinet or Workflow Cabinet.

Usage

```
function = dmg_brw32->DmgBrsPresentCabinetEx
```

Attachment inputs

Variable	Description
CABINET	The name of the Cabinet. This may be a Workflow Cabinet. Required.

Attachment outputs

Variable	Description
FILTER_VARIABLES	The number of Cabinet Filter Variables for the Cabinet. May be 0.
FILTER_VARIABLESx.field	A record set describing the Filter Variables, where x is the record number and field is one of the following fields. Exists only if FILTER_VARIABLE is greater than 0.
.NAME	The Name of the Filter Variable.
.DATATYPE	The DataType of the Filter Variable. This is a numeric value from the enum DmgDataTypes in dmg_api_types.h.
.LENGTH	The Length of the Filter Variable, in bytes.
IS_WORKFLOW	Contains 1 if the Cabinet is a Workflow Cabinet, else 0.
WORK_TABLE	The WorkTable Property of the Cabinet. Exists only if IS_WORKFLOW contains 1.
WORKFLOW	The Workflow Property of the Cabinet.. Exists only if IS_WORKFLOW contains 1.
DESCRIPTION	The Description Property of the Cabinet. Exists only if IS_WORKFLOW contains 1.
XDP_ENABLED	The XDPEnabled Property of the Cabinet. Contains 1 if enabled, else 0.
CASE_SENSITIVE	The CaseSensitive Property of the Cabinet. Contains 1 if case sensitive, else 0.
DOC_TABLE	The DocTable Property of the Cabinet.
LEVELS	The Levels Property of the Cabinet, that is, the number of Levels. This is also the number of the following Levels records. This will always be at least 1, and will be exactly 1 when IS_WORKFLOW contains 1.
LEVELSx.field	A record set describing the Levels, where x is the record number and field is one of the following fields.
.NAME	The Name Property of the Level.

Variable	Description
.VISIBLE	The Visible Property of the Level. Contains 1 if visible, else 0.
.ALLOW_DOCS_IN_TABLE	The AllowDocsInTable Property of the Level. Contains 1 if allowed, else 0.
.ALLOW_DOCS_IN_CABINET	The AllowDocsInCabinet Property of the Level. Contains 1 if allowed, else 0.
.TABLE_NAME	The TableName Property of the Level.
.DESCRIPTION	The Description Property of the Level.
.QUALIFIED_TABLE_NAME	The QualifiedTableName Property of the Level.
.NESTED	The name of the record set (row set) containing the extended Level properties for the Level. Always use this name to locate the extended level properties, since the name used for the record set may change in future releases of the Bridge.
[.NESTED value]	This is a variable whose name is the value of .NESTED for the Level, and whose value is the number of Extended Properties of the Level.
[.NESTED value]y.field	A record set describing the Extended Properties of Level x, where y is the record number and field is one of the following fields. Note that this record set is nested within one LEVELS record, so that each variable name is prepended with the value of .NESTED for the Level.
[.NESTED value]y.NAME	The Name of the Extended Property.
[.NESTED value]y.NAME_BE	The Bridge-encoded Name of the Extended Property.
[.NESTED value]y.QUALIFIED_NAME	The QualifiedName of the Extended Property.
[.NESTED value]y.DATATYPE	The DataType of the Extended Property. This is a numeric value from the enum DmgDataTypes in dmg_api_types.h.
[.NESTED value]y.LENGTH	The Length of the Extended Property, in bytes.
[.NESTED value]y.EDITABLE	IsEditable for the Extended Property. Contains 1 if editable, else 0.
[.NESTED value]y.KEY	IsKey for the Extended Property. Contains 1 if is key, else 0.
[.NESTED value]y.REQUIRED	IsRequired for the Extended Property. Contains 1 if is required, else 0.
[.NESTED value]y.VISIBLE	IsVisible for the Extended Property. Contains 1 if is visible, else 0.

Variable	Description
[.NESTED value]y.IS_STANDARD	Contains 1 if the Extended Property is one of the Standard Extended Project Properties, else 0. This will always be 0 when IS_WORKFLOW contains 0.
CAB_CATEGORIES	The number of Categories that are legal for Documents being put into this Cabinet.
CAB_CATEGORIESy.NAM E	A record set of the names of the Categories that are legal for Documents put into this Cabinet, where y is the record number.

Remarks DmgBrsPresentCabinetEx presents information about a particular Cabinet or Workflow Cabinet.

The Extended Level Properties describe Extended Folder or Project Properties, depending on whether the Cabinet is an ordinary Cabinet or a Workflow Cabinet.

To get a list of Cabinets, use DmgBrsListCabinets or DmgBrsListWorkflows.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsListCabinets
DmgBrsListWorkflows
DmgBrsPresentCategory

DMGBRSPRESENTCATEGORY

DmgBrsPresentCategory presents information about a particular Category.

Usage

```
function = dmg_brw32->DmgBrsPresentCategory
```

Attachment inputs

Variable	Description
DOC_CATEGORY	The Name of the Category. Required.

Attachment outputs

Variable	Description
CAT_USE_FLAG_1	The UseFlag1 Property of the Category. Contains 1 if the Flag is in use, else 0.
CAT_USE_FLAG_2	The UseFlag2 Property of the Category. Contains 1 if the Flag is in use, else 0.
CAT_USE_KEYWORD_1	The UseKeyword1 Property of the Category. Contains 1 if the Keyword is in use, else 0.
CAT_USE_KEYWORD_2	The UseKeyword2 Property of the Category. Contains 1 if the Keyword is in use, else 0.
CAT_FLAG_1_LABEL	The Flag1Label Property of the Category. Exists only if CAT_USE_FLAG_1 contains 1.
CAT_FLAG_2_LABEL	The Flag2Label Property of the Category. Exists only if CAT_USE_FLAG_2 contains 1.
CAT_KEYWORD_1_LABEL	The Keyword1Label Property of the Category. Exists only if CAT_USE_KEYWORD_1 contains 1.
CAT_KEYWORD_2_LABEL	The Keyword2Label Property of the Category. Exists only if CAT_USE_KEYWORD_2 contains 1.
CAT_SUBCATEGORY_LABEL	The SubcategoryLabel Property of the Category.
CAT_STATUS_LABEL	The StatusLabel Property of the Category.
CAT_DATE_LABEL	The DateLabel Property of the Category.
CAT_ADDEDON_LABEL	The AddedOnLabel Property of the Category.
CAT_ADDEDBY_LABEL	The AddedByLabel Property of the Category.
CAT_ADDEDFROM_LABEL	The AddedFromLabel Property of the Category.
CAT_APPLICATION_LABEL	The ApplicationLabel Property of the Category.
CAT_FLAG1_PL	The number of Flag1 Pickiest strings. Exists only if CAT_USE_FLAG_1 contains 1.

Variable	Description
CAT_FLAG1_PL x.field	A record set containing the strings for the Flag1 Picklist, where x is the record number and field is the following field.
.STRING	A Picklist String.
CAT_FLAG2_PL	The number of Flag2 Picklist strings. Exists only if CAT_USE_FLAG_2 contains 1.
CAT_FLAG2_PL x.field	A record set containing the strings for the Flag2 Picklist, where x is the record number and field is the following field.
.STRING	A Picklist String.
CAT_STATUS_PL_EXISTS	Contains 1 if a Status Picklist exists, else 0.
CAT_STATUS_PL	The number of Status Picklist strings. Exists only if CAT_STATUS_PL_EXISTS contains 1.
CAT_STATUS_PL x.field	A record set containing the strings for the Status Picklist, where x is the record number and field is the following field.
.STRING	The Picklist String.
CAT_SUBCATEGORY_PL_EXISTS	Contains 1 if a Subcategory Picklist exists, else 0.
CAT_SUBCATEGORY_PL	The number of Subcategory Picklist strings. Exists only if CAT_STATUS_PL_EXISTS contains 1.
CAT_SUBCATEGORY_PL x.field	A record set containing the strings for the Picklist, where x is the record number and field is the following field.
.STRING	The Picklist String.
XCATS_EXIST	Contains 1 if the Category has Extended Properties, else contains 0.
XCATS	The number of Extended Properties. Exists only if XCATS_EXIST contains 1.
XCATSx.field	A record set describing the Extended Properties of the Category, where x is the record number and field is one of the following fields.
.NAME	The Name of the Extended Property.
.NAME_BE	The Bridge-encoded Name of the Extended Property.
.QUALIFIED_NAME	The QualifiedName of the Extended Property.
.DATATYPE	The DataType of the Extended Property. This is a numeric value from the enum DmgDataTypes in dmg_api_types.h.

Variable	Description
.LENGTH	The Length of the Extended Property, in bytes.
.EDITABLE	IsEditable for the Extended Property. Contains 1 if editable, else 0.
.KEY	IsKey for the Extended Property. Contains 1 if is key, else 0.
.REQUIRED	IsRequired for the Extended Property. Contains 1 if is required, else 0.
.VISIBLE	IsVisible for the Extended Property. Contains 1 if is visible, else 0.

RemarksRemarks DmgBrsPresentCategory presents information about a particular Category.
The Extended Category Properties describe Extended Document Properties.
To get a list of Categories, use DmgBrsListCategories.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsListCategories
DmgBrsListCabinets
DmgBrsPresentCategory

DMGBRSPRESENTCOMPOUNDDOCUMENT

DmgBrsPresentCompoundDocument presents information about a Compound Document, including a list of the Subdocuments in it and their characteristics.

Usage

```
function = dmg_brw32->DmgBrsPresentCompoundDocument
```

Attachment inputs

Variable	Description
RUN_PRESENTCOMPOUNDDOCUMENT	This rule will run if the Document is a Compound Document and this variable does not exist or this variable exists and contains 1. If this variable exists and contains 0, or if the Document is not compound, this rule will not run. Optional.
Contents file info	The outputs of DmgBrsCacheContentsFile for the Document to present. Required.
SHOW_ACTIONS	Set this to 1 to cause the DOC_ACTION_... output attachment variables to be output, else set to 0. Optional. Default is the INI option [PresentCompoundDocument] ShowActions.
SHOW_VERSIONS	Set this to 1 to cause the VERSIONSx.field output attachment record set to be output, else set to 0. Optional. Default is the INI option [PresentCompoundDocument] ShowVersions.
VERSIONS_INDEX	The index at which to start displaying versions for the Document. Zero-based. Optional. Default is 0. Ignored if SHOW_VERSIONS is 0.
MAX_VERSIONS	Maximum number of version records allowed to be returned in one transaction. Limited by the INI option [PresentCompoundDocument] MaxVersions. Optional. Default is [PresentCompoundDocument] MaxVersions. Ignored if SHOW_VERSIONS is 0.
SUBDOCS_INDEX	The index at which to start displaying Subdocuments for the Document. Zero-based. Optional. Default is 0.
MAX_SUBDOCS	Maximum number of Subdocument records allowed to be returned in one transaction. Limited by the INI option [PresentCompoundDocument] MaxSubdocuments. Optional. Default is [PresentCompoundDocument] MaxSubdocuments.

Variable	Description
SHOW_SUBODCCS_PAGES	Set this to 1 to show the variable .PAGES in the DOC_SUBDOCS output record set. Calculating the number of pages typically has a performance cost. Optional. Default is the INI option [PresentCompoundDocument] ShowSubdocsPages.
PS_LIST	A Property Set List that limits what Basic and Extended Document Properties are returned for each record in the VERSIONS record set. Uses the PS_DOC and PS_XDOC Property Sets. Optional. Ignored if SHOW_VERSIONS is 0. If this variable is missing or empty, DmgBrsPresentCompound Document will automatically look for a Property Set List whose name is "PresentCompoundDocumentDefault", which need not exist.

Attachment outputs

Variable	Description
DOC_CONVERSION_AVAILABLE	Contains 1 if the Document is a type that can be converted with the rule DmgBrsReturnConvertedDocument, else contains 0.
DOC_CONVERSION_TYPE	The value of the INI option [Rendering] DocumentConversion.
ORIGNAME	A URL Tail with the Document's Name and FileType.
CONVNAME	A URL Tail with the Document's Name and converted FileType.
CONVNAMEEROOT	A URL Tail with the Document's Name, but with no dot and no extension. This makes it convenient to build an application with an alternate conversion type for a Document.
DOCSPEC	The DOCSPEC for the Document.
DOC_ACTION_...	A set of variables whose names begin with "DOC_ACTION_" that contain either 1 or 0 to indicate whether that action is allowed for the current user for this Document. These exist only if ShowActions is ON. All of the variables are listed in the description of DmgBrsPresentDocument.
DOC_...	A set of variables whose names begin with DOC_ that contain the Basic Properties of the Document. All of the variables are listed in the description of DmgBrsPresentDocument.
XDOC_propertyname	Variables that contain all of the Extended Document Properties for this Document, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name). These depend on the Document's Category.

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.
VERSIONS	The number of Document versions being returned. Exists only if SHOW_VERSIONS is 1.
VERSIONSx.field	A record set describing the Document versions that were found, where x is the record number and field is one of the following fields. Exists only if VERSIONS exists and is greater than 0. The records in this record set are the same as those for DmgBrsPresentDocument.
MOREVERSIONS	Contains 1 if there are versions after the ones that have been returned in this transaction, else contains 0. Exists only if ShowVersions is on.
PREVVERSIONS	Contains 1 if there are versions previous to the ones that have been returned in this transaction, else contains 0. Exists only if SHOW_VERSIONS is 1.
VERSIONS_INDEX_NEXT	The index of the next batch of versions. Exists only if MOREVERSIONS contains 1.
VERSIONS_INDEX_PREV	The index of the previous batch of versions. Exists only if PREVVERSIONS contains 1.
DOC_NUMSUBDOCS	Contains the total number of Subdocuments in the Compound Document.
DOC_SUBDOCS	Contains the number of Subdocuments returned in this transaction.
DOC_SUBDOCSx.field	A record set describing the Subdocuments that are being returned, where x is the record number and field is one of the following fields.
.LOCATION	The location property of the Subdocument. This is its stated location within the Compound Document, as returned by the Compound Document plugin that interpreted the Compound Document. May be empty.
.NUM	The SUBDOC_NUM for this Subdocument, used to identify it in rules which use SUBDOC_NUM.

Variable	Description
.NAME	The name property of the Subdocument, as returned by the Compound Document plugin that interpreted the Compound Document. May be empty.
.PAGES	The number of pages in the Subdocument. This will be 0 if the number of pages cannot be calculated. This variable exists only if SHOW_SUBDOCS_PAGES contains 1.
.TYPE	The file type (extension) of the Subdocument file.
.ORIGNAME	Like ORIGNAME above, but applicable to the Subdocument.
.CONVERSION_AVAILABLE	Like DOC_CONVERSION_AVAILABLE above, but applicable to the Subdocument. DOC_CONVERSION_TYPE applies both to the Document and to the Subdocuments.
.CONVNAME	Like CONVNAME above, but applicable to the Subdocument. Does not exist if .CONVERSION_AVAILABLE is 0.
.CONVNAME_ROOT	Like CONVNAME_ROOT above, but applicable to the Subdocument. Does not exist if .CONVERSION_AVAILABLE is 0.
.HAS_XPROPS	Contains 1 if the Subdocument has extended properties, else contains 0.
.NESTED	The name of the record set (row set) containing the extended Subdocument properties for the Subdocument. Always use this name to locate the extended Subdocument properties, since the name used for the record set could change in future releases of the Bridge.
[.NESTED value]	This is a variable whose name is the value of .NESTED for the Subdocument, and whose value is the number of extended Subdocument properties for this Subdocument. Exists only when .HAS_XPROPS contains 1.
[.NESTED value]y.field	A record set describing the extended Subdocument properties of Subdocument x, where y is the record number and field is one of the following fields. Note that this record set is nested within one DOC_SUBDOCS record, so that each variable name is prepended with the value of .NESTED for the Subdocument.
[.NESTED value]y.NAME	The name of the Subdocument extended property.
[.NESTED value]y.VALUE	The value of the Subdocument extended property.
MORESUBDOCS	Contains 1 if there are Subdocuments after the ones that have been returned in this transaction, else contains 0.

Variable	Description
PREVSUBDOCS	Contains 1 if there are Subdocuments previous to the ones that have been returned in this transaction, else contains 0.
SUBDOCS_INDEX_NEXT	The index of the next batch of Subdocuments. Exists only if MORESUBDOCS contains 1.
SUBDOCS_INDEX_PREV	The index of the previous batch of Subdocuments. Exists only if PREVSUBDOCS contains 1.
CAB_CATEGORIES	The number of Categories that are legal for Documents being put into this Cabinet.
CAB_CATEGORIESy.NAME	A record set of the names of the Categories that are legal for Documents put into this Cabinet, where y is the record number.

Remarks

DmgBrsPresentCompoundDocument presents information about a Compound Document, including a list of the Subdocuments in it and their characteristics.

This rule is very similar to DmgBrsPresentDocument. The main difference between them is that DmgBrsPresentDocument lists pages in the Document whereas this rule lists Subdocuments in the Document.

This rule will not run under any of the following three conditions:

- 1 RUN_PRESENTCOMPOUNDDOCUMENT exists and contains 0. This variable allows this rule to be controlled by DmgBrsSetupCompoundDocument.
- 2 The Document is not a Compound Document, as indicated in the contents subdocument info. This allows this rule to be present on a rules list along with DmgBrsPresentDocument, and only one of the rules will execute depending on whether the Document was compound or simple.
- 3 The attachment variable SUBDOC_NUM exists. This allows this rule to be present on a rules list along with DmgBrsPresentSubdocument, and only one of the rules will execute depending on whether the Subdocument number was specified or not.

Examples

The following rules list can be used to present a simple Document, a Compound Document, or a Subdocument. If the Document is a simple Document, only DmgBrsPresentDocument will run. If the Document is a Compound Document, only DmgBrsPresentSubdocument will run if SUBDOC_NUM is input, otherwise only DmgBrsPresentCompoundDocument will run. We presume in all cases that the necessary inputs for DmgBrsCacheContentsFile exist.

```
[ ReqType: PRES_DOC ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmgs_brw32->DmgBrsCopyAttachment
function = dmgs_brw32->DmgBrsValidateSession
function = dmgs_brw32->DmgBrsCacheContentsFile
function = dmgs_brw32->DmgBrsPresentDocument
function = dmgs_brw32->DmgBrsPresentCompoundDocument
function = dmgs_brw32->DmgBrsPresentSubdocument
```

ReqType changes

[ReqTypes] Failure if the rule fails

INI options	[PresentCompoundDocument] [Rendering] DocumentConversion
Validations	Session and Reqtype
Platform restrictions	None
May be followed by	DmgBrsPresentDocument DmgBrsPresentSubdocument
See also	DmgBrsPresentDocument DmgBrsPresentSubdocument DmgBrsSetupCompoundDocument DmgBrsCacheContentsFile

DMGBRSPRESENTDOCUMENT

DmgBrsPresentDocument presents information about a Document and its versions.

Usage

```
function = dmg_brw32->DmgBrsPresentDocument
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document to present. Required.
SHOW_PAGES	Set this to 1 to cause the PAGESx.field output attachment record set and DOC_PAGES to be output, else set to 0. Setting this to 1 has a performance cost since the number of pages in the Document must be determined. Optional. Default is the INI option [PresentDocument] ShowPages.
MAX_THUMBNAILS	Maximum number of PAGES records that will have USE_MISSING set to 0. Limited by the INI option [Thumbnails] MaxThumbnails. See below. Optional. Default is [Thumbnails] MaxThumbnails. Ignored if SHOW_PAGES is 0.
SHOW_ACTIONS	Set this to 1 to cause the DOC_ACTION_...output attachment variables to be output, else set to 0. Optional. Default is the INI option [PresentDocument] ShowActions.
SHOW_VERSIONS	Set this to 1 to cause the VERSIONSx.field output attachment record set to be output, else set to 0. Optional. Default is the INI option [PresentDocument] ShowVersions.
VERSIONS_INDEX	The index at which to start displaying versions for the Document. Zero-based. Optional. Default is 0. Ignored if SHOW_VERSIONS is 0.
MAX_VERSIONS	Maximum number of version records allowed to be returned in one transaction. Limited by the INI option [PresentDocument] MaxVersions. Optional. Default is [PresentDocument] MaxVersions. Ignored if SHOW_VERSIONS is 0.
PS_LIST	A Property Set List that limits what Basic and Extended Document Properties are returned for each record in the VERSIONS record set. Uses the PS_DOC and PS_XDOC Property Sets. Optional. Ignored if SHOW_VERSIONS is 0. If this variable is missing or empty, DmgBrsPresentDocument will automatically look for a Property Set List whose name is "PresentDocumentDefault", which need not exist.

CAB_CATEGORIES	The number of Categories that are legal for Documents being put into this Cabinet.
CAB_CATEGORIESy.NAME	A record set of the names of the Categories that are legal for Documents put into this Cabinet, where y is the record number.

Attachment outputs

Variable	Description
DOC_CONVERSION_AVAILABLE	Contains 1 if the Document is a type that can be converted with the rule DmgBrsReturnConvertedDocument, else contains 0.
DOC_CONVERSION_TYPE	The value of the INI option [Rendering] DocumentConversion.
ORIGNAME	A URL Tail with the Document's Name and FileType.
CONVNAME	A URL Tail with the Document's Name and converted FileType.
CONVNAMEROOT	A URL Tail with the Document's Name, but with no dot and no extension. This makes it convenient to build an application with an alternate conversion type for a Document.
DOCSPEC	The DOCSPEC for the Document
DOC_THUMBNAILS	Contains 1 if the INI option [Thumbnails] Thumbnails is ON, else contains 0. This may used by a Bridge Application to conditionally show tables of thumbnail URLs. Exists only if SHOW_PAGES contains 1.
PAGES	The number of pages in the Document. Exists only if SHOW_PAGES contains 1.
PAGESx.field	A record set with a record for each page in the Document, where x is the record number and field is one of the following fields. The fields can be used to build thumbnail URLs, which could go on an HTML page that displays thumbnails. Exists only if SHOW_PAGES contains 1.
.PAGE	The page number for this record, starting with 1.
.PAGENAME	A URL Tail with the Document's Name, the page number, and the value of the INI option [Rendering] PageConversion for the extension (or the appropriate extension if EMF_RENDER is 1), suitable for use in forming a URL for DmgBrsReturnDocumentPage. Example: "/My%20Doc_p2.jpg"

.THUMBNAME	A URL Tail with the Document's Name, the page number prefixed with the letter 't' (or 'm' if USE_MISSING is 1), and JPG for the FileType, suitable for use in forming a URL for DmgBrReturnThumbnail. Example: "/My%20Doc_t2.JPG", "/My%20Doc_tm.JPG"
.USE_MISSING	This will contain 0 for the first [Thumbnails] MaxThumbnails records, then 1 thereafter, unless [Thumbnails] MissingFile is empty. In the latter case, it will always be 0. See DmgBrReturnThumbnail.
DOC_ACTION_...	A set of variables whose names begin with "DOC_ACTION_" that contain either 1 or 0 to indicate whether that action is allowed for the current user for this Document. These exist only if ShowActions is ON. All of the variables are listed below.
DOC_...	A set of variables whose names begin with DOC_ that contain the Basic Properties of the Document. All of the variables are listed below.
XDOC_propertyname	Variables that contain all of the Extended Document Properties for this Document, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name). These depend on the Document's Category.
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.
EMF_RENDER	If the Document is a DCD file and the INI option [Rendering] DCDEnhancedDisplay is 1, this variable contains 1, else it contains 0. This is intended for applications that want to conditionally use the DV Lite control to display DCD files.
VERSIONS	The number of Document versions being returned. Exists only if SHOW_VERSIONS is 1.

VERSIONSx.field	A record set describing the Document versions that were found, where x is the record number and field is one of the following fields. Exists only if VERSIONS exists and is greater than 0.
.DOCSPEC	The DOCSPEC for the version.
.DOC_ID	The Id Property of the version.
.DOC_MAJORVERSION	The MajorVersion Property of the version.
.DOC_MINORVERSION	The MinorVersion Property of the version.
.DOC_VERS	The DOC_VERS of the version.
.DOC_CATEGORY	The Category Property of the version.
.DOC_VERSIONCOMMENT	The VersionComment Property of the version.
.DOC_LASTEDITEDBY	The LastEditedBy Property of the version.
.DOC_LASTEDITEDON	The LastEditedOn Property of the version.
.DOC_...	Any additional Basic Document Properties listed in the Property Set List, or all these Properties if no Property Set List was used.
.XDOC_propertyname	Any Extended Document Properties listed in the Property Set List, or all these Properties (if any) if no Property Set List was used. These will be appropriate for the Document's Category.
MOREVERSIONS	Contains 1 if there are versions after the ones that have been returned in this transaction, else contains 0. Exists only if ShowVersions is on.
PREVVERSIONS	Contains 1 if there are versions previous to the ones that have been returned in this transaction, else contains 0. Exists only if ShowVersions is on.
VERSIONS_INDEX_NEXT	The index of the next batch of versions. Exists only if MOREVERSIONS contains 1.
VERSIONS_INDEX_PREV	The index of the previous batch of versions. Exists only if PREVVERSIONS contains 1.

Remarks

DmgBrsPresentDocument presents information about a Document and its versions. The rule DmgBrsCacheContentsFile must precede this rule on the rules list. This rule does nothing if the Document is a Compound Document. Use DmgBrsPresentCompoundDocument to present a Compound Document.

The information about the Document includes Basic and Extended Document Properties, the permissions the current user has relative to the Document, a Folder or Project Specification for the Document's container, variables that make it convenient to construct URLs for returning the Document, returning a page from the Document, returning thumbnails for the pages of the Document, or returning a converted version of the Document, and the versions of the Document.

The information about the versions of the Document includes Basic and Extended Properties of the versions.

The optional PAGES record set contains variables useful in setting up URLs for calling DmgBrsReturnDocumentPage and DmgBrsReturnThumbnail. This allows a table to be created in an HTML page that displays a list of pages. Thumbnail images of pages could be used as links to the pages. An important consideration when using thumbnail images is performance. Each thumbnail takes some time to create and return, so an HTML page with many thumbnails on it could take a long time to finish loading. To help control this, use the INI option [Thumbnails] MaxThumbnails and the input attachment variable MAX_THUMBNAILS. The variable USE_MISSING in the PAGES record set is set to 0 until the maximum number of thumbnails is reached. After that, it is set to one. USE_MISSING may be passed to DmgBrsReturnThumbnail to tell it to return the thumbnail file specified in [Thumbnails] MissingFile. Or, USE_MISSING may be used in an application to detect when to stop showing thumbnails or page records.

The names of the DOC_... variables are:

```
DOC_NAME, DOC_AUTHOR, DOC_CATEGORY, DOC_DESCRIPTION,  
DOC_SUBCATEGORY, DOC_STATUS, DOC_KEYWORD1, DOC_KEYWORD2, DOC_FLAG1,  
DOC_FLAG2, DOC_DATE, DOC_FILETYPE, DOC_ADDEDFROM, DOC_ADDEDON,  
DOC_DUEDATE, DOC_CHECKEDOUTBY, DOC_CHECKEDOUTBYLONG,  
DOC_CHECKEDOUTFOR, DOC_VOLUME, DOC_MAJORVERSION, DOC_MINORVERSION,  
DOC_LASTEDITEDBY, DOC_LASTEDITEDON, DOC_ID, DOC_KEYSTRING,  
DOC_OBSOLETE, DOC_RELEASED, DOC_APPROVED, DOC_CHECKEDOUT,  
DOC_LOCKED, DOC_REFERENCE, DOC_OWNER, DOC_CONTENTMODIFIEDON,  
DOC_CONTENTSIZE, DOC_LASTACCESSION, DOC_VERSIONCOMMENT,  
DOC_SOURCEDOCUMENTID, DOC_SOURCEDOCUMENTCABINET,  
DOC_CHECKEDOUTCABINET, DOC_CHECKEDOUTDOCID, DOC_ANNOTATIONSLOCKED,  
DOC_ANNOTATIONSLOCKUSER, DOC_ANNOTATIONSLOCKHOST,  
DOC_ANNOTATIONSLOCKTIME, DOC_CONTENTSTAG, DOC_ANOANNOTATIONSTAG,  
DOC_RETENTIONDATE, DOC_PROTECTIONSTATUS.
```

The names of the DOC_ACTION_... variables that are output if ShowActions is on are:

```
DOC_ACTION_COPY, DOC_ACTION_CHECKOUT, DOC_ACTION_DELETE,  
DOC_ACTION_EDIT_CONTENTS, DOC_ACTION_VIEW,  
DOC_ACTION_ANNOTATE_LOCAL, DOC_ACTION_CHECKIN_SAME,  
DOC_ACTION_MOVE, DOC_ACTION_INSERT, DOC_ACTION_REFRESH_DEPENDENTS,  
DOC_ACTION_REFRESH_CUTSOURCE, DOC_ACTION_MAKE_REFERENCE,  
DOC_ACTION_BLACKOUT_LOCAL, DOC_ACTION_CHECKIN_MAJOR,  
DOC_ACTION_CHECKIN_MINOR, DOC_ACTION_VIEW_HISTORY,  
DOC_ACTION_EDIT_ATTRIBUTES, DOC_ACTION_ANNOTATE_GLOBAL,  
DOC_ACTION_BLACKOUT_GLOBAL, DOC_ACTION_END_LOCAL_ANNOTATE,  
DOC_ACTION_END_GLOBAL_ANNOTATE, DOC_ACTION_VIEW_WORKING,  
DOC_ACTION_PUBLISH, DOC_ACTION_CHECKOUT_IN,  
DOC_ACTION_GET_UNIQUE_KEY, DOC_ACTION_DOWNLOAD,  
DOC_ACTION_GETREMOVEDNAME, DOC_ACTION_CHECKIN_UNDO,  
DOC_ACTION_MIGRATEFILE, DOC_ACTION_DELETEALL,  
DOC_ACTION_SET_STATUS, DOC_ACTION_RESET_STATUS.
```

- Examples**
- 1 In our first example, let's assume we have a system where Documents tend to have a large number of versions. We're going to use DmgBrsPresentDocument to get those versions 5 at a time, and we're not concerned with the other outputs of DmgBrsPresentDocument. Assume the following INI file settings.

```
[PresentDocument] ShowVersions= ON  
[PresentDocument] MaxVersions= 10  
[PresentDocument] ShowPages= ON  
[PresentDocument] ShowActions= ON
```

Assume that the Flag1 and Flag2 Properties of a version are of special interest to us. We will construct a Property Set and Property Set List as follows:

```
[PS_LIST:VersionDisplay]
PS_DOC:VersionDisplay

[PS_DOC: VersionDisplay]
DOC_FLAG1
DOC_FLAG2
```

The attachment inputs for the first batch of 5 versions might look like the following.

```
CABINET = MyCabinet
DOC_ID = 1274
DOC_MAJORVERSION = 22
DOC_MINORVERSION = 4
SHOW_PAGES = 0
SHOW_ACTIONS = 0
MAX_VERSION = 5
PS_LIST = VersionDisplay
VERSIONS_INDEX = 0
```

We passed in SHOW_PAGES and SHOW_ACTIONS set to 0 and MAX_VERSIONS set to 5 to over-ride the respective INI Options. We passed in VERSIONS_INDEX set to 0, although it defaults to 0 so that wasn't strictly necessary.

The outputs might look like the following.

```
DOC_CONVERSION_AVAILABLE = 1
DOC_CONVERSION_TYPE = PDF
CONVNAME = /MyDoc.PDF
CONVNAME_ROOT = /MyDoc
ORIGNAME = /MyDoc.MET
DOCSPEC = CABINET=myCabinet&DOC_ID=1274&DOC_VERS=22.4
DOC_... = [all of the Basic Properties of the Document]
XDOC_propertyname = [all of the Extended Properties of the Document]
IS_WORKFLOW = 0
FLD_LEVELNUMBER = 1
FLD_KEYSTRING = DMGSamples.dbo.Amergen.LName = 'Bradley'
FLD_ID = 29
EMF_RENDER = 0
VERSIONS = 5
VERSION1.DOCSPEC = CABINET=myCabinet&DOC_ID=1274&DOC_VERS=1.0
VERSION1.DOC_ID = 1274
VERSION1.DOC_MAJORVERSION = 1
VERSION1.DOC_MINORVERSION = 0
VERSION1.DOC_VERS = 1.0
VERSION1.DOC_CATEGORY = Policies
VERSION1.DOC_VERSIONCOMMENT =
VERSION1.DOC_LASTEDITEDBY = Charles Conway
VERSION1.DOC_LASTEDITEDON = 12/15/2004 11:15:29.433
VERSION1.DOC_FLAG1 = Automotive
VERSION1.DOC_FLAG2 = Truck
... [and so on through VERSION5.DOC_FLAG2 = ]
MOREVERSIONS = 1
PREVVERSIONS = 0
VERSIONS_INDEX_NEXT = 5
```

To get the next batch of versions, we will pass in the value of VERSIONS_INDEX_NEXT for VERSIONS_INDEX. The outputs from that transaction will include the following variables for navigating the versions.


```
MOREVERSIONS = 1
PREVVERSIONS = 1
VERSIONS_INDEX_NEXT = 10
VERSIONS_INDEX_PREV = 0
```

- 2 In our second example, we'll focus on the PAGES record set. Assume that ShowPages is ON, that MaxThumbnails is 2, and that the Document has 3 pages. The PAGES record set might look like the following.

```
PAGES = 3
PAGES1.PAGE = 1
PAGES1.PAGENAME = /MyDoc_p1.JPG
PAGES1.THUMBNAME = /MyDoc_t1.JPG
PAGES1.USE_MISSING = 0
PAGES2.PAGE = 2
PAGES2.PAGENAME = /MyDoc_p2.JPG
PAGES2.THUMBNAME = /MyDoc_t2.JPG
PAGES2.USE_MISSING = 0
PAGES3.PAGE = 3
PAGES3.PAGENAME = /MyDoc_p3.JPG
PAGES3.THUMBNAME = /MyDoc_tm.JPG
PAGES3.USE_MISSING = 1
```

In the above, notice that USE_MISSING is 0 for the first two pages, but is 1 starting with the third page to indicate that only 2 thumbnails are to be created and that the file configured in the INI option [Thumbnails] MissingFile is to be used after that.

- 3 The following shows how to set up a clickable thumbnail image of a page that takes the user to that page, for each page in the PAGES record set. It's a snippet from an HTML template based on the outputs of DmgBrsPresentDocument. BridgeURL comes from DmgBrsValidateSession. Assume a REQTYPE of PresentP runs a rules list with DmgBrsPresentDocumentPage on it, and ReturnT a rules list with DmgBrsReturnThumbnail.

```
<A HREF=#BRIDGEURL, #&REQTYPE= PresentP&#DOCSPEC, #&PAGE=#PAGE, #>
<IMG
SRC=#BRIDGEURL, #&REQTYPE=ReturnT&#DOCSPEC, #&PAGE=#PAGE, #&USE_MISSING=
G=#USE_MISSING, #&THUMBNAME=#THUMBNAME, #></A>
```

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

[PresentDocument]
[Rendering] DocumentConversion
[Rendering] PageConversion
[Rendering] DCDEnhancedDisplay
[Thumbnails] Thumbnails
[Thumbnails] MaxThumbnails
[Thumbnails] MissingFile

Validations

Session and Reqtype

Platform restrictions

The page count of a Document is available only on Windows. On other platforms, the number of pages in the Document will show as zero. On non-Windows platforms, it is recommended to set SHOW_PAGES to 0.

May be followed by

DmgBrsPresentFolder

DmgBrsPresentProject

You could follow DmgBrsPresentDocument with either of the above to return information about the Document's container as well as about the Document itself. Note that DmgBrsPresentDocument automatically outputs the required input variables of the above rules.

See also

DmgBrsLogPropertySets

DmgBrsPresentCompoundDocument

DmgBrsPresentSubdocument

DmgBrsPresentDocumentPage

DmgBrsReturnConvertedDocument

DmgBrsReturnDocument

DmgBrsReturnDocumentPage

DmgBrsReturnThumbnail

DmgBrsCacheContentsFile

DMGBRSPRESENTDOCUMENTPAGE

DmgBrsPresentDocumentPage presents information about a page of a Document.

Usage

```
function = dmg_brw32->DmgBrsPresentDocumentPage
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document that contains the page to present. Required.
SUBDOC_NUM	The one-based Subdocument number for the Subdocument whose page is to be presented. Required if the Document is Compound, else ignored.
PAGE	The one-based number of the page to present. For a Compound Document, this will be a page of Subdocument SUBDOC_NUM. Required.
ROT	The rotation state of the page image. Valid values are 0, 1, 2, or 3 for 0, 90, 180, or 270 degrees clockwise rotation. Optional. Defaults to 0.

Attachment outputs

Variable	Description
DOC_ISSUBDOC	Contains 1 if the page belongs to a Subdocument, else contains 0.
DOC_PAGES	The number of pages in the Document. Exists only if DOC_ISSUBDOC contains 0.
SUBDOC_PAGES	The number of pages in the Subdocument. Exists only if DOC_ISSUBDOC contains 1.
EMF_RENDER	If the Document is a DCD file and the INI option [Rendering] DCDEnhancedDisplay is 1, this variable contains 1, else it contains 0. This is intended for applications that want to conditionally use the DVLite control to display DCD files.
DOCSPEC	The DOCSPEC of the Document.
DOC_ID	The Id Property of the Document.
DOC_MAJORVERSION	The MajorVersion Property of the Document.
DOC_MINORVERSION	The MinorVersion Property of the Document.
DOC_VERS	The DOC_VERS of the Document.
PAGENAME	A URL Tail with the Document's Name, the page number from PAGES, and the value of the INI option [Rendering] PageConversion for the extension (or the appropriate extension if EMF_RENDER is 1), suitable for use in forming a URL for DmgBrsReturnDocumentPage. Example: "/My%20Doc_p2.jpg"

HAS_NEXT_PAGE	Contains 1 if there are pages in the Document after the input variable PAGE, else contains 0.
HAS_PREV_PAGE	Contains 1 if there are pages in the Document prior to the input variable PAGE, else contains 0.
NEXT_PAGENAME	A URL Tail formatted like PAGENAME, with the page number in NEXT_PAGENUM. Exists only if HAS_NEXT_PAGE contains 1.
PREV_PAGENAME	A URL Tail formatted like PAGENAME, with the page number in PREV_PAGENUM. Exists only if HAS_PREV_PAGE contains 1.
FIRST_PAGENAME	A URL Tail formatted like PAGENAME, with the page number in FIRST_PAGENUM.
LAST_PAGENAME	A URL Tail formatted like PAGENAME, with the page number in LAST_PAGENUM.
NEXT_PAGENUM	The page number of the next page of the Document. Exists only if HAS_NEXT_PAGE contains 1.
PREV_PAGENUM	The page number of the previous page of the Document. Exists only if HAS_PREV_PAGE contains 1.
FIRST_PAGENUM	The page number of the first page of the Document. This is always 1.
LAST_PAGENUM	The page number of the last page of the Document.
ROTATABLE	Contains 1 if the page image is a rotatable type, else contains 0.
ROT_90	The ROT value to cause the current image to be rotated 90 degrees clockwise from its current rotation state.
ROT_180	The ROT value to cause the current image to be rotated 180 degrees clockwise from its current rotation state.
ROT_270	The ROT value to cause the current image to be rotated 270 degrees clockwise from its current rotation state.
DOC_FILETYPE	The Document's FileType Property.
DOC_NAME	The Document's Name Property.
DOC_DATE	The Document's Date Property.
DOC_VOLUME	The Document's Volume Property.
SUBDOC_NAME	The name of the Subdocument. Exists only if DOC_ISSUBDOC contains 1.
SUBDOC_TYPE	The file type (extension) of the Subdocument. Exists only if DOC_ISSUBDOC contains 1.

Remarks DmgBrsPresentDocumentPage presents information about a page of a Document, and output attachments that make it convenient to construct an HTML page that has references to the page image and controls for navigating to other pages in the Document or to rotated versions of the page image. The rule DmgBrsCacheContentsFile must precede this rule on the rules list.

The usual method of displaying an image of a Document page on an HTML page is to put an IMG tag in the HTML with the SRC set to a URL that will run a rules list with DmgBrsReturnDocumentPage. The inputs of DmgBrsReturnDocumentPage include a Version Document Specification, a page number, a ROT value, and a PAGENAME URL tail.

To display the current page, simply pass DOCSPEC, PAGE, ROT, and PAGENAME to DmgBrsReturnDocumentPage.

To build URLs for such things as the first and last pages of the Document, the usual method is to create A (anchor) tags in the HTML with the HREF set to a URL that will run the rules list with DmgBrsPresentDocumentPage on it again, but varying the inputs.

DmgBrsPresentDocumentPage outputs sufficient information to create inputs for itself to return such pages. The Version Document Specification, returned in DOCSPEC, is the same for any image. The page number (PAGE) and ROT value, if relevant, to supply to DmgBrsPresentDocumentPage for the various images are shown in the following table.

Image	PAGE	ROT
next page	NEXT_PAGENUM	—
previous page	PREV_PAGENUM	—
first page	FIRST_PAGENUM	—
last page	LAST_PAGENUM	—
current page, rotated 90 degrees CW from current rotation	PAGE	ROT_90
current page, rotated 180 degrees CW from current rotation	PAGE	ROT_180
current page, rotated 270 degrees CW from current rotation	PAGE	ROT_270

NEXT_PAGENAME, etc., could be used to create an HTML page that had multiple IMG tags on it to display, for example, a page image along with a thumbnail image of the next page. In this case, you would construct an IMG in the HTML with a SRC that contained a URL to run DmgBrsReturnThumbnail, passing NEXT_PAGENAME as the THUMBNAME input attachment of DmgBrsReturnThumbnail.

Examples The following are some snippets from an HTML template that contains various IMG and A tags based on the outputs of DmgBrsPresentDocumentPage. BridgeURL comes from DmgBrsValidateSession. Assume a REQTYPE of PresentP runs a rules list with DmgBrsPresentDocumentPage on it, ReturnP a rules list with DmgBrsReturnDocumentPage, and ReturnT a rules list with DmgBrsReturnThumbnail.

Here's the main image on a page presentation page.

```
<IMG  
SRC=#BRIDGEURL, #&REQTYPE=ReturnP&#DOCSPEC, #&PAGE=#PAGE, #&ROT=#ROT, #  
&PAGENAME=#PAGENAME, #>
```

Here's a URL to re-display the page presentation page with the image rotated 90 degrees clockwise from its current rotation.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=PresentP&#DOCSPEC, #&PAGE=#PAGE, #&ROT=#ROT
  _90, #>Rotate right</A>
```

Here's a URL to re-display the page presentation page with the image unrotated, for use after it has been rotated.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=PresentP&#DOCSPEC, #&PAGE=#PAGE, #&ROT=0>Vi
  ew unrotated</A>
```

Here's a URL to display a page presentation page with the previous page on it. Notice that we have omitted ROT, so that the previous page will display unrotated.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=PresentP&#DOCSPEC, #&PAGE=#PREV_PAGENUM, #>
  Previous Page</A>
```

Here's a thumbnail image of the next page, to allow it to be previewed.

```
<IMG
  SRC=#BRIDGEURL, #&REQTYPE=ReturnT&#DOCSPEC, #&PAGE=#NEXT_PAGENUM, #&TH
  UMBNAME=#NEXT_PAGENAME, #>
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [Rendering] PageConversion
[Rendering] DCDEnhancedPageDisplay
[Rendering] CompressEnhancedDCD

Validations Session and Reqtype

Platform restrictions Windows only.

See also DmgBrsPresentDocument
DmgBrsReturnDocumentPage
DmgBrsReturnThumbnail
DmgBrsCacheContentsFile

DMGBRSPRESENTFOLDER

DmgBrsPresentFolder presents information about a Folder and any Notes associated with it.

Usage

```
function = dmg_brw32->DmgBrsPresentFolder
```

Attachment inputs

Variable	Description
IS_WORKFLOW	If this exists and contains 1, this rule does nothing. Allows the rule to co-exist on a rules list with DmgBrsPresentProject following rules that output IS_WORKFLOW, such as DmgBrsCheckInDocument. Optional.
A Folder Specification	The Folder to present. Required.
SHOW_ANCESTORS	Set this to 1 to cause information about the Folder's ancestors to be output, else to 0. Optional. Default is [PresentFolder] ShowAncestors.
SHOW_NOTES	Set this to 1 to cause any Notes associated with the Folder to be output, else to 0. Optional. Default is the INI option [Notes] ShowNotes.
NOTES_AUTO_FILTER	This limits the Notes that are returned. Possible values, case-insensitive, are ALL—return any Note CURRENT—return only those Notes not marked as deleted DELETED—return only those Notes marked as deleted Any unrecognized value is interpreted as ALL. Optional. Default is [Notes] NotesAutoFilter. Ignored if SHOW_NOTES contains 0.
SHOW_NUM_DOCS	Set this to 1 to cause FLD_HAS_DOCS and FLD_NUM_DOCS to be output, else to 0. Setting this to 1 has a performance cost. Default is the INI option [PresentFolder] ShowNumDocs.
SHOW_ACTIONS	Set this to 1 to cause the FLD_ACTION_... output attachment variables to be output, else set to 0. Optional. Default is the INI option [PresentFolder] ShowActions.
NOTES_INDEX	The zero-based index to start displaying Notes for the Folder. Ignored if SHOW_NOTES is 0. Optional. Default is 0.
MAXNOTES	The maximum number of Notes to return in this transaction. Limited by the INI option [Notes] MaxNotes. Optional. Default is the INI option [Notes] MaxNotes.

Attachment outputs

Variable	Description
FLD_LEVELNUMBER	The LevelNumber Property of the Folder being presented.
FLD_KEYSTRING	The KeyString Property of the Folder being presented.

FLD_ID	The Id Property of the Folder being presented.
LEVEL_EXISTS_Lx	A set of variables, one for each Level of the Cabinet. These contain 1 to indicate that FLD_Lx_... and XFLD_Lx_... variables exist for the level, else 0. If ShowAncestors is ON, these will contain 1 for Levels 1 through FLD_LEVELNUMBER. If ShowAncestors is OFF, these will contain 0 for all Levels except FLD_LEVELNUMBER.
FLD_Lx_LABEL	The Label Property for the Folder at Level x, where x is one based. Exists only if the corresponding LEVEL_EXISTS_Lx contains 1.
FLD_Lx_ID	The Id Property for the Folder at Level x, where x is one based. Exists only if the corresponding LEVEL_EXISTS_Lx contains 1.
FLD_Lx_KEYSTRING	The KeyString Property for the Folder at Level x, where x is one based. Exists only if the corresponding LEVEL_EXISTS_Lx contains 1.
XFLD_Lx_propertyname	The Extended Properties of the Folder at Level x, where x is one based and propertyname is the Bridge-encoded Folder Property Name. These exist if the corresponding LEVEL_EXISTS_Lx contains 1.
FLD_ACTION_...	A set of variables whose names begin with "FLD_ACTION_" that contain either 1 or 0 to indicate whether that action is allowed for the current user for the Folder being presented. These exist only if ShowActions is ON. All of the variables are listed below.
FLD_HAS_DOCS	Contains 1 if the Folder being presented contains Documents, else contains 0. Exists only if SHOW_NUM_DOCS contains 1.
FLD_NUM_DOCS	Contains the number of Documents in the Folder being presented. Exists only if SHOW_NUM_DOCS contains 1.
NOTES_EXIST	Contains 1 if the Folder being presented has Notes associated with it, else contains 0.
NOTES	The number of Notes associated with the Folder. Exists only if NOTES_EXIST contains 1 and SHOW_NOTES contains 1.
NOTESx.field	A record set describing the Notes that were found, where x is the record number and field is one of the following fields. Exists only if NOTES exists and is greater than 0.
.NTE_INDEX	A zero-based index identifying this particular Note for this Folder. It may be passed to such rules as DmgBrsMarkNoteDeleted to identify the Note.
.NTE_ADDEDON	The AddedOn Property of the Note.
.NTE_ADDEDBY	The AddedBy Property of the Note.

.NTE_DESCRIPTION	The Description Property of the Note.
.NTE_ADDITIONALINFO	The AdditionalInfo Property of the Note.
.NTE_DELETED	The Deleted Property of the Note. Contains 1 if the Note is marked deleted else 0 if the Note is marked current.
MORENOTES	Contains 1 if there are Notes after the ones that have been returned in this transaction, else contains 0. Exists only if SHOW_NOTES contains 1.
PREVNOTES	Contains 1 if there are Notes previous to the ones that have been returned in this transaction, else contains 0. Exists only if SHOW_NOTES contains 1.
NOTES_INDEX_NEXT	The index of the next batch of Notes. Exists only if MORENOTES contains 1.
NOTES_INDEX_PREV	The index of the previous batch of Notes. Exists only if PREVNOTES contains 1.
STARTAT	A variable containing a value that will cause the rule DmgBrsQuery to display the contents of the Folder being presented. If a STARTAT variable already exists on the output queue, then DmgBrsPresentFolder does not over-write it.

Remarks

DmgBrsPresentFolder presents information about a Folder and any Notes associated with it.

The information about the Folder includes the Basic and Extended Folder Properties of the Folder being presented and of its ancestor Folders in a multi-Level Cabinet, the permissions the current user has relative to the Folder being presented, whether the Folder contains Documents and how many, and whether the Folder has Notes.

The information about the Notes associated with the Folder includes the Basic Properties of each Note.

The names of the FLD_ACTION_... variables that are output if ShowActions is on are:

```
FLD_ACTION_INSERT, FLD_ACTION_GET_UNIQUE_KEY,  
FLD_ACTION_EDIT_ATTRIBUTES, FLD_ACTION_CHECKOUT_IN,  
FLD_ACTION_MAKE_REFERENCE_IN, FLD_ACTION_INSERT_IN,  
FLD_ACTION_DELETE.
```

Examples

- 1 In this example, we'll assume that we're presenting a Folder at the second Level of a Cabinet that has three Levels, and returning information about the Folder's ancestor Folder. We don't care about Notes. Assume for this example that [PresentFolder] ShowActions is ON, so we have to over-ride it with an input attachment variable.

The input attachment variables might look like the following.

```
CABINET = MyThreeLevelCab  
FLD_ID = 998  
FLD_LEVELNUMBER = 2  
SHOW_ANCESTORS = 1  
SHOW_NOTES = 0  
SHOW_NUM_DOCS = 1  
SHOW_ACTIONS = 0
```

The outputs might look something like the following.

```
FLD_LEVELNUMBER = 2
LEVEL_EXISTS_L1 = 1
LEVEL_EXISTS_L2 = 1
LEVEL_EXISTS_L3 = 0
FLD_L1_ID = 235
```

... [and the remaining basic Properties of the parent Folder of the Folder being presented]

```
XFLD_L1_Name = Brace
```

... [and the remaining extended Properties of the parent Folder of the Folder being presented]

```
FLD_L2_ID = 998
```

... [and the remaining basic Properties of the Folder being presented]

```
XFLD_L2_Policy = Brace
```

... [and the remaining extended Properties of the Folder being presented]

```
FLD_HAS_DOCS = 1
FLD_NUM_DOCS = 7
STARTAT = 1,2,DB.dbo.Name = 'Brace'
```

Notice in the above that nothing for Notes or Actions is returned because the inputs suppressed that information. FLD_HAS_DOCS and FLD_NUM_DOCS were returned because the inputs requested that information. They apply to the Folder being presented.

If SHOW_ANCESTORS had been set to 0, then LEVEL_EXISTS_L1 would have been 0, and the FLD_L1_... and XFLD_L1_... variables would not have appeared in the output.

- 2 In this example, we'll assume we're presenting a Folder from a single level Cabinet, and that the thing most of interest is the Notes, specifically the Notes marked current. Assume for this example that [PresentFolder] ShowActions is OFF, so there's no need to over-ride it with an input attachment variable.

The input attachment variables might look like the following.

```
CABINET = MyCab
FLD_ID = 998
SHOW_NOTES = 1
SHOW_NUM_DOCS = 0
NOTES_AUTO_FILTER = CURRENT
```

The output attachment variables might look like the following.

```
FLD_LEVELNUMBER = 1
LEVEL_EXISTS_L1 = 1
FLD_L1_ID = 661
```

... [and the remaining basic Properties of the Folder being presented]

```
XFLD_L1_Name = World
```

... [and the remaining extended Properties of the Folder being presented]

```
NOTES_EXIST = 1
```

```

NOTES = 2
NOTES1.NTE_INDEX = 0
NOTES1.NTE_DELETED = 0

... [and all the other NTE_ variables for record 1]

NOTES2.NTE_INDEX = 1
NOTES2.NTE_DELETED = 0

... [and all the other NTE_ variables for record 2]

MORENOTES = 0
PREVNOTES = 0
STARTAT = 0,70

```

- 3 In this example, we'll set up a rules list that uses DmgBrsQueryDocuments to show the list of Documents in the Folder. To do this, we'll use DmgBruCopyAttachmentVar to set up a FLD_FILTER, based on the KeyString Property of the Folder being presented, and a LEVEL_LIMITER, based on the LevelNumber Property of the Folder being presented, for DmgBrsQueryDocuments. Setting these variables assures that the Documents returned are limited to those in the intended Folder.

```

[ ReqType: PRES_FLD ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsPresentFolder
function = dmg_brw32->DmgBruCopyAttachmentVar, runf, output, FLD_LEVELNUMBER, output, LEVEL_LIMITER
function = dmg_brw32->DmgBruCopyAttachmentVar, runf, output, FLD_KEYSTRING, output, FLD_FILTER
function = dmg_brw32->DmgBrsQueryDocuments

```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [PresentFolder]

[Notes] ShowNotes

[Notes] NotesAutoFilter

[Notes] MaxNotes

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsQuery
DmgBrsQueryDocuments

See also DmgBrsAddNote
DmgBrsLogPropertySets
DmgBrsMarkNoteCurrent
DmgBrsMarkNoteDeleted

DmgBrsPresentProject

DmgBrsQuery

DMGBRSPRESENTPROJECT

DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.

Usage

```
function = dmg_brw32->DmgBrsPresentProject
```

Attachment inputs

Variable	Description
IS_WORKFLOW	If this exists and contains 0, this rule does nothing. Allows the rule to co-exist on a rules list with DmgBrsPresentFolder following rules that output IS_WORKFLOW, such as DmgBrsCheckInDocument. Optional.
A Project Specification	The Project to present. Required.
SHOW_DOCS	Set to 1 to cause a list of Documents in the Project to be returned, else 0. Optional. Default is the INI option [PresentProject] ShowDocs.
SHOW_NOTES	Set this to 1 to cause any Notes associated with the Project to be output, else to 0. Optional. Default is the INI option [Notes] ShowNotes.
NOTES_AUTO_FILTER	This limits the Notes that are returned. Possible values, case-insensitive, are ALL—return any Note CURRENT—return only those Notes not marked as deleted DELETED—return only those Notes marked as deleted Any unrecognized value is interpreted as ALL. Optional. Default is [Notes] NotesAutoFilter. Ignored if SHOW_NOTES contains 0.
SHOW_TEAM	Set this to 1 to cause a list of the Team members for the Project to be returned, else 0. Optional. Default is the INI option [PresentProject] ShowTeam.
SHOW_HISTORY	Set this to 1 to cause the History of the Project to be returned, else 0. Optional. Default is the INI option [PresentProject] ShowHistory.
SHOW_ACTIONS	Set this to 1 to cause the PRJ_ACTION_...output attachment variables to be output, else set to 0. Optional. Default is the INI option [PresentProject] ShowActions.
NOTES_INDEX	The zero-based index to start displaying Notes for the Project. Ignored if SHOW_NOTES contains 0. Optional. Default is 0.
MAXNOTES	The maximum number of Notes to return in this transaction. Limited by the INI option [Notes] MaxNotes. Optional. Default is the INI option [Notes] MaxNotes.
DOCS_INDEX	The zero-based index at which to start returning Documents. Optional. Default is 0. Ignored if SHOW_DOCS contains 0.

MAXDOCS	The number of Documents allowed to be returned in one transaction. Limited by the INI option [PresentProject] MaxDocumentHits. Optional. Default is [PresentProject] MaxDocumentHits. Ignored if SHOW_DOCS contains 0.
PS_LIST	A Property Set List to limit any Basic or Extended Document Properties to return for each Document found. Uses the PS_DOC and PS_XDOC Property Sets. Optional. If absent, DmgBrsPresentProject will use a Property Set List called "PresentProjectDefault" if one exists. Ignored if SHOW_DOCS contains 0.

Attachment outputs

Variable	Description
TSK_INSTRUCTIONS	The Instructions Property of the Task the Project is at.
TSK_MANAGER	The Manager Property of the Task the Project is at.
TSK_DESCRIPTION	The Description Property of the Task the Project is at.
TSK_CABINET	The Cabinet Property of the Task the Project is at.
TSK_FILTER	The Filter Property of the Task the Project is at. This is weakly encrypted to make it harder to pass in a malicious filter.
TSK_WORKFLOW	The Workflow Property of the Task the Project is at.
TSK_WORKFLOW_DESCRIPTION	The WorkflowDescription Property of the Task the Project is at.
TSK_ID	The Id Property of the Task the Project is at.
TSK_COUNT	The Count Property of the Task the Project is at.
PRJ_LABEL	The Label Property of the Project.
PRJ_ID	The Id Property of the Project.
PRJ_KEYSTRING	The Keysting Property of the Project.
PRJ_TASKTYPE	The TaskType Property of the Project.
PRJ_TASKTYPE_TEXT	The TaskType Property of the Project, in text form, where the text matches the enum DmgTaskTypes in dmg_api_types.h.
PRJ_standardextendedproperty	A set of variables containing the values of the 19 Standard Extended Project Properties described in the Introduction.
XPRJ_propertyname	A set of variables containing the values of the Extended Project Properties, where propertyname is the Bridge-encoded property name.

PRJ_ACTION_...	A set of variables whose names begin with "PRJ_ACTION_" that contain either 1 or 0 to indicate whether that action is allowed for the current user for this Project. These exist only if SHOW_ACTIONS contains 1. All of the variables are listed below.
CHECKED_OUT	Contains 1 if the Project is checked out to this user, else contains 0.
DOCS_EXIST	Contains 1 if a Document list is output, else contains 0. Always 0 if SHOW_DOCS contains 0.
DOCS	The number of Documents in the Documents list. Exists only if DOCS_EXIST contains 1 and SHOW_DOCS contains 1.
DOCSx.field	A record set that lists the Document returned, where x is the record number and field represents fields the same as those returned in the DHITS list of DmgBrsQuery, with the specific Properties returned controlled by the Property Set List passed in, or all these Properties if no Property Set List was used. Exists only if DOCS_EXIST contains 1 and SHOW_DOCS contains 1.
MOREDOCS	Contains 1 if there are Documents after the ones that have been returned in this transaction, else contains 0. Exists only if DOCS_EXIST contains 1 and SHOW_DOCS contains 1.
PREVDOCS	Contains 1 if there are Documents previous to the ones that have been returned in this transaction, else contains 0. Exists only if DOCS_EXIST contains 1 and SHOW_DOCS contains 1.
DOCS_INDEX_NEXT	The index of the next batch of Documents. Exists only if MOREDOCS contains 1.
DOCS_INDEX_PREV	The index of the previous batch of Documents. Exists only if PREVDOCS contains 1.
BRANCHES_EXIST	Contains 1 if there is a list of Branches, else contains 0. This will contain 1 only if the task type is DMG_TASK_HUMANDECISION.
BRANCHES	The number of Branches. Exists only if BRANCHES_EXIST contains 1.
BRANCHESx.field	A record set describing the Branches that were found, where x is the record number and field is one of the following fields. Exists only if BRANCHES exists and is greater than 0.
.BRN_NAME	The Name Property of the Branch.
.BRN_ID	The Id Property of the Branch. This may be passed in to DmgBrsAdvanceProject.

TEAM_EXISTS	Contains 1 when the Team is being returned, else contains 0. Always contains 0 when SHOW_TEAM contains 0.
TEAM_MEMBERS	The number of Team members. Exists only if TEAM_EXISTS contains 1.
TEAM_MEMBERSx.field	A record set listing the Team members, where x is the record number and field is one of the following fields. Exists only if TEAM_EXISTS contains 1.
.TEAM_MEMBER_NAME	A Team member name. This name may be passed in to DmgBrsForwardProject.
.TEAM_MEMBER_DISPLAYNAME	A Team member display name.
.TEAM_MEMBER_QUALIFIEDNAME	A Team member qualified name.
HISTORY_EXISTS	Contains 1 when the History is being returned, else contains 0. Always contains 0 when SHOW_HISTORY contains 0.
HISTORY	The number of History records. Exists only if HISTORY_EXISTS contains 1.
HISTORYx.field	A record set listing the Project's History, where x is the record number and the field is one of the following fields. Exists only if HISTORY_EXISTS contains 1.
.HISTORY_DESCRIPTION	The Description property of the History record.
.HISTORY_RESULT_ID	The Result ID property of the History record.
.HISTORY_PERFORMED_BY	The Performed by property of the History record.
.HISTORY_LOCKED	The Locked property of the History record.
.HISTORY_SINCE	The Since property of the History record.
.HISTORY_ROUTE_TO	The Route To property of the History record.
NOTES_EXIST	Contains 1 if the Project being presented has Notes associated with it, else contains 0.
NOTES	The number of Notes associated with the Project. Exists only if NOTES_EXIST contains 1 and SHOW_NOTES contains 1.
NOTESx.field	A record set describing the Notes that were found, where x is the record number and field is one of the following fields. Exists only if NOTES exists and is greater than 0.

.NTE_INDEX	A zero-based index identifying this particular Note for this Project. It may be passed to such rules as DmgBrsMarkNoteDeleted to identify the Note.
.NTE_ADDEDON	The AddedOn Property of the Note.
.NTE_ADDEDBY	The AddedBy Property of the Note.
.NTE_DESCRIPTION	The Description Property of the Note.
.NTE_ADDITIONALINFO	The AdditionalInfo Property of the Note.
.NTE_DELETED	The Deleted Property of the Note. Contains 1 if the Note is marked deleted else 0 if the Note is marked current.
MORENOTES	Contains 1 if there are Notes after the ones that have been returned in this transaction, else contains 0. Exists only if SHOW_NOTES contains 1.
PREVNOTES	Contains 1 if there are Notes previous to the ones that have been returned in this transaction, else contains 0. Exists only if SHOW_NOTES contains 1.
NOTES_INDEX_NEXT	The index of the next batch of Notes. Exists only if MORENOTES contains 1.
NOTES_INDEX_PREV	The index of the previous batch of Notes. Exists only if PREVNOTES contains 1.

Remarks DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.

The information about the Project includes the Properties of the Task that the Project is at, the Basic and Extended Properties of the Project, the permissions the current user has relative to the Project, whether the Project is checked out to the current user, any Branches available for advancing the Project, and the names of Team Members to whom the Project could be forwarded.

The information about the Documents in the Project includes a list of Documents with Basic and Extended Document Properties for each Document, as specified in a Property Set List.

The information about the Notes associated with the Folder includes the Basic Properties of each Note.

The names of the PRJ_ACTION_... variables that are output if SHOW_ACTIONS contains 1 are:

```
PRJ_ACTION_INSERT, PRJ_ACTION_GET_UNIQUE_KEY,
PRJ_ACTION_EDIT_ATTRIBUTES, PRJ_ACTION_CHECKOUT_IN,
PRJ_ACTION_MAKE_REFERENCE_IN, PRJ_ACTION_INSERT_IN,
PRJ_ACTION_DELETE, PRJ_ACTION_CHECKOUT, PRJ_ACTION_CHECKIN_SAME,
PRJ_ACTION_CHECKIN_PROJ, PRJ_ACTION_CHECKIN_PROJ_MOVE,
PRJ_ACTION_CHECKIN_PROJ_ASSIGN, PRJ_ACTION_SUSPEND,
PRJ_ACTION_RESUME
```

Examples See DmgBrsPresentFolder for an example of how Notes work.

Also see the examples for DmgBrsQuery for ways to link the Documents on the above hit list to various other rules, such as DmgBrsPresentDocument.

The following example shows possible input and output attachment variables for a Project that contains a Document and has Branches. The Property Set List has a PS_XPRJ Property Set with several Properties, a PS_DOC Property Set with the DOC_NAME and DOC_FILETYPE Properties, and PS_XDOC Property Sets for the CLAIM and POLICY Categories. The input attachment variables might look like the following:

```
CABINET = Claim Properties
PRJ_KEYSTRING = dbo.WF_Claims.Claim_Number = 8723.000000 and
dbo.WF_Claims.I_TaskID = 8009
PS_LIST = CLAIMPROPERTIESWFCABINET
SHOW_DOCS = 1
SHOW_TEAM = 0
```

The output attachment variables might look like the following.

```
BRANCHES = 2
BRANCHES1.BRN_ID = 8003
BRANCHES1.BRN_NAME = Pay Claim
BRANCHES2.BRN_ID = 8010
BRANCHES2.BRN_NAME = Investigate
BRANCHES_EXIST = 1
CHECKED_OUT = 0
DOCS = 1
DOCS1.CONVNAME = /bg.pdf
DOCS1.DOC_CATEGORY = CLAIM
DOCS1.DOC_FILETYPE = JPG
DOCS1.DOC_ID = 586
DOCS1.DOC_MAJORVERSION = 1
DOCS1.DOC_MINORVERSION = 0
DOCS1.DOC_NAME = bg
DOCS1.DOCSPEC = CABINET=Claim%20Properties&
DOC_ID=586&DOC_VERS=1.0
DOCS1.ORIGNAME = /bg.JPG
DOCS1.XDOC_Accident$20Cause =
DOCS1.XDOC_Cust$20at$20Fault? =
DOCS_EXIST = 1
MOREDOCS = 0
NOTES_EXIST = 0
PREV_DOCS = 0
PRJ_ID = 104
PRJ_KEYSTRING = dbo.WF_Claims.Claim_Number = 8723.000000 and
dbo.WF_Claims.I_TaskID = 8009
PRJ_LABEL = 8723
PRJ_TASKTYPE = 9
PRJ_TASKTYPE_TEXT = Human Decision
TEAM_EXISTS = 0
TSK_CABINET = Claim Properties
TSK_COUNT = 2
TSK_DESCRIPTION = Inside Adjuster
TSK_FILTER = 1yTqCtAEAy-
hJvylsEUlpURoTlAGpvSEqvTkM9BlrCM3sV3fqZOasOVYthxzzlpntitz0i8Li8wXa8
duokgWuE0itLzM2a00
TSK_ID = 8009
TSK_INSTRUCTIONS = The inside adjuster will add all supporting
documents for the claim: medical records, police reports, pictures,
or any statements supporting the claim. The inside adjuster will
also determine if the claim should be paid or investigated.
TSK_MANAGER = DEMO1
```

```
TSK_WORKFLOW = Claim Properties
TSK_WORKFLOW_DESCRIPTION =
XPRJ_Claim$20Approved =
XPRJ_Claim$20Number = 8723.000000
XPRJ_Claim$20Remarks =
XPRJ_I$20AssignedTo =
XPRJ_I$20DueDate = 2004-08-13 21:22:32
XPRJ_I$20ProjectID = 104
XPRJ_I$20TaskID = 8009
XPRJ_P$20Description = Project 3
XPRJ_P$20DueDate = 2004-08-13 16:41:22
XPRJ_P$20Priority = 1
XPRJ_P$20ProjectID = 104
XPRJ_P$20WorkflowID = Claim Properties
XPRJ_Policy$20Number = 9873.000000
XPRJ_Vehicle$20ID = 5134.000000
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [PresentProject]
[Notes] ShowNotes
[Notes] NotesAutoFilter
[Notes] MaxNotes

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsAdvanceProject
DmgBrsCheckOutProject
DmgBrsForwardProject
DmgBrsGetTaskQueue
DmgBrsLogPropertySets
DmgBrsPresentFolder
DmgBrsQuery
DmgBrsQueryDocuments

DMGBRSPRESENTSUBDOCUMENT

DmgBrsPresentSubdocument presents information about a specific Subdocument in a Compound Document.

Usage

```
function = dmg_brw32->DmgBrsPresentSubdocument
```

Attachment inputs

Variable	Description
SUBDOC_NUM	The one-based Subdocument number of the Subdocument to present. If this variable is absent, this rule does nothing.
Contents file info	The outputs of DmgBrsCacheContentsFile for the Document that contains the Subdocument. Required when SUBDOC_NUM exists.
SHOW_SUBDOCS_PAGES	Set this to 1 to cause the PAGESx.field output attachment record set to be output, else set to 0. Setting this to 1 has a performance cost since the number of pages in the Subdocument must be determined. Optional. Default is the INI option [PresentSubdocument] ShowSubdocsPages.
MAX_THUMBNAILS	Maximum number of PAGES records that will have USE_MISSING set to 0. Limited by the INI option [Thumbnails] MaxThumbnails. See below. Optional. Default is [Thumbnails] MaxThumbnails. Ignored if SHOW_SUBDOCS_PAGES is 0.

Attachment outputs

Note that many of these output variables apply to the Compound Document that contains the Subdocument, and not to the Subdocument. This provides a context for the Subdocument.

Variable	Description
SUBDOC_CONVERSION_AVAILABLE	Contains 1 if the Subdocument is a type that can be converted with the rule DmgBrsReturnConvertedDocument, else contains 0.
SUBDOC_CONVERSION_TYPE	The value of the INI option [Rendering] DocumentConversion.
ORIGNAME	A URL Tail with the Subdocument's Name and FileType.
CONVNAME	A URL Tail with the Subdocument's Name and converted FileType.
CONVNAMEROOT	A URL Tail with the Subdocument's Name, but with no dot and no extension. This makes it convenient to build an application with an alternate conversion type for a Subdocument.
DOCSPEC	The DOCSPEC for the Document.

Variable	Description
SUBDOC_LOCATION	The location property of the Subdocument. This is its stated location within the Compound Document, as returned by the Compound Document plugin that interpreted the Compound Document. May be empty.
SUBDOC_NAME	The name property of the Subdocument, as returned by the Compound Document plugin that interpreted the Compound Document. May be empty.
SUBDOC_TYPE	The file type (extension) of the Subdocument file.
SUBDOC_HAS_XPROPS	Contains 1 if the Subdocument has extended properties, else contains 0.
SUBDOC_XPROPS	The number of extended Subdocument properties for this Subdocument. Exists only when SUBDOC_HAS_XPROPS contains 1.
SUBDOC_XPROPSx.field	A record set describing the extended Subdocument properties of the Subdocument, where x is the record number and field is one of the following fields.
.NAME	The name of the Subdocument extended property.
.VALUE	The value of the Subdocument extended property.
SUBDOC_THUMBNAIIS	Contains 1 if the INI option [Thumbnails] Thumbnails is ON, else contains 0. This may used by a Bridge Application to conditionally show tables of thumbnail URLs. Exists only if SHOW_SUBDOCS_PAGES contains 1.
PAGES	The number of pages in the Subdocument. Exists only if SHOW_SUBDOCS_PAGES contains 1.
PAGESx.field	A record set with a record for each page in the Subdocument, where x is the record number and field is the field name. The fields are identical to those for the PAGES record set in DmgBrsPresentDocument. Exists only if SHOW_SUBDOCS_PAGES contains 1.
DOC_...	A set of variables whose names begin with DOC_ that contain the Basic Properties of the Document. All of the variables are listed in the description of DmgBrsPresentDocument.
XDOC_propertyname	Variables that contain all of the Extended Document Properties for this Document, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name). These depend on the Document's Category.

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.
CAB_CATEGORIES	The number of Categories that are legal for Documents being put into this Cabinet.
CAB_CATEGORIESy.NAME	A record set of the names of the Categories that are legal for Documents put into this Cabinet, where y is the record number.

Remarks

DmgBrsPresentSubdocument presents information about a specific Subdocument in a Compound Document.

Note that many of the output variables for this rule apply to the Compound Document that contains the Subdocument, and not to the Subdocument. This provides a context for the Subdocument.

This rule is very similar to DmgBrsPresentDocument. The main difference between them is that DmgBrsPresentDocument lists versions and pages of the Document whereas this rule lists Subdocument properties and pages of the Subdocument.

This rule will not run under either of the following two conditions:

- 1 The Document is not a Compound Document, as indicated in the contents subdocument info. This allows this rule to be present on a rules list along with DmgBrsPresentDocument, and only one of the rules will execute depending on whether the Document was compound or simple.
- 2 The attachment variable SUBDOC_NUM does not exist. This allows this rule to be present on a rules list along with DmgBrsPresentCompoundDocument, and only one of the rules will execute depending on whether the Subdocument number was specified or not.

The input variable SUBDOC_NUM may be obtained from the outputs for the rule DmgBrsPresentCompoundDocument.

Examples

See the example in the description of DmgBrsPresentCompoundDocument.

ReqType changes

[ReqTypes] Failure if the rule fails

INI options

[PresentSubdocument]

[Thumbnails] Thumbnails

[Thumbnails] MaxThumbnails

[Thumbnails] MissingFile

Validations

Session and Rectype

Platform restrictions

On non-Windows platforms, the number of pages in a subdocument is usually unavailable, and so will be set to zero. This will cause the PAGES record set to be empty.

May be followed by

DmgBrsPresentDocument

DmgBrsPresentCompoundDocument

See also

DmgBrsPresentDocument

DmgBrsPresentCompoundDocument

DmgBrsCacheContentsFile

DMGBRSPRESENTWORKFLOW

DmgBrsPresentWorkflow presents information about a workflow schema.

Usage

```
function = dmg_brw32->DmgBrsPresentWorkflow
```

Attachment inputs

Variable	Description
CABINET	The name of the Workflow Cabinet to present, as returned from DmgBrsListWorkflows. Case sensitive. Required.
SHOW_TASKS	Set this to 1 to return the list of Tasks for the Workflow. Set it to 0 or any value other than 1 to omit the list of Tasks. Optional. The default value is 1.
SHOW_TASK_CONTAINERS	Set this to 1 to return the list of Task Containers for the Workflow. Set it to 0 or any value other than 1 to omit the list of Task Containers. Optional. The default value is 1.

Attachment outputs

Variable	Description
WORKFLOW	The Workflow Property of the Workflow Schema
WORK_TABLE	The Worktable Property of the Workflow Schema
DESCRIPTION	The Description Property of the Workflow Schema
TASKS	The number of Tasks for the Workflow. This will be 0 if the input attachment variable SHOW_TASKS contained 0.
TASKS x . <i>field</i>	A record set describing the Tasks, where x is the record number and <i>field</i> is one of the following fields. The fields are the Basic Properties of a Task Schema. Exists only if TASKS contains a value greater than 0
.INSTRUCTIONS	The Instructions property of the Task Schema.
.MANAGER	The Manager property of the Task.
.DESCRIPTION	The Description property of the Task.
.WORKFLOW	The Workflow property of the Task.
.ID	The Id property of the Task.
.PARENT_TASK_CONTAINER_ID	The Parent Id property of the Task. This will be 0 if the Task is not in a container.
TASK_CONTAINERS	The number of Task Containers for this Workflow Schema. This will be 0 if the input attachment variable SHOW_TASK_CONTAINERS was 0, or if no Containers were defined for the Workflow Schema.

Variable	Description
TASK_CONTAINERS x . <i>field</i>	A record set describing the Task Containers, where x is the record number and <i>field</i> is one of the following fields. The fields are the Basic Properties of a Task Container Schema. Exists only if TASK_CONTAINERS contains a value greater than 0.
.ID	The Id property of the Container.
.NAME	The Name property of the Container.
.DESCRIPTION	The Description property of the Container.
.PARENT_TASK_CONTAINER_ID	The Parent Id property of the Container. This will be 0 if the Container is not in a Container.
.MANAGER	The Manager property of the Container.

Remarks DmgBrsPresentWorkflow presents information about a Workflow Schema, including Task Schemas and Containers. This can be used, for example, to construct a UI showing the structure of workflow containers, if any, and what tasks are in what containers.

You must supply the case-sensitive name of a valid Workflow Cabinet in the input attachment variable CABINET, or the rule will fail.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype.

Platform restrictions None.

See also DmgBrsListWorkflows

DMGBRSQUERY

DmgBrsQuery searches for Folders and Documents within a Folder.

Usage

```
function = dmg_brw32->DmgBrsQuery
```

Attachment inputs

Variable	Description
CABINET	The name of the Cabinet to search. Required. Note that this is the only required input attachment variable for this rule, unless the Cabinet requires Cabinet filters to be searched. See next variable.
CAB_FILTER_variablename	A set of variables containing Cabinet filter terms, where variablename is the Bridge encoded name of a particular Cabinet filter variable, as defined for the Cabinet in the Documange Administrator application. Pass in as many as needed, if any, to open the Cabinet. Optional, unless required by the Cabinet definition.
FULLTEXT_FILTER	A full text filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
FLD_FILTER	An Extended Folder Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
DOC_FILTER	A Basic Document Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
XDOC_FILTER	An Extended Document Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
VER_FILTER	A Document Version Basic Property filter for the search. This filter has no effect if the input attachment variable QUERY_VERSIONS contains zero. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
FLD_ORDERBY	A Folder order by SQL clause for the search. Optional. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini.
DOC_ORDERBY	A Document order by SQL clause for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
DOC_CATEGORY	The Category Property for the Extended Document Property filter. Required if there is an Extended Document Property filter, else ignored.
QUERY_DOC_...	SQL Builder variables describing a Basic Document Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DOC_FILTER, and will be ignored if DOC_FILTER is passed in. Optional.

QUERY_VER_...	SQL Builder variables describing a Document Version Property filter. The syntax for these variables is identical to the syntax for the QUERY_DOC_... variables. This is an alternative to VER_FILTER, and will be ignored if VER_FILTER is passed in. Optional.
QUERY_XDOC_...	SQL Builder variables describing an Extended Document Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to XDOC_FILTER, and will be ignored if XDOC_FILTER is passed in. Optional.
QUERY_XFLD_...	SQL Builder variables describing an Extended Folder Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to FLD_FILTER, and will be ignored if FLD_FILTER is passed in. Optional.
OBn_XFLD_...	SQL Builder variables describing a Folder Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to FLD_ORDERBY, and will be ignored if FLD_ORDERBY is passed in. Optional.
OBn_DOC_...	SQL Builder variables describing a Document Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DOC_ORDERBY, and will be ignored if DOC_ORDERBY is passed in. Optional.
PS_LIST	A Property Set List to limit any Basic or Extended Document Properties to return for each Document found and any Extended Folder Properties to return for each Folder found. Uses PS_DOC, PS_XDOC, and PS_XFLD Property Sets. Optional. If absent, DmgBrsQuery will use a Property Set List called "QueryDefault" if one exists.
QUERY_VERSIONS	Set this variable to 1 to cause this rule to apply the criteria to all Documents and return any Document Versions that match, rather than just the current version. Optional. Default is the INI option [Query] QueryVersions.
MAXHITS	The total number of Folder and Document hits allowed to be returned in one transaction. Limited by the INI option [Query] MaxHits. Optional. Default is the INI option [Query] MaxHits.
MAXFLDCOUNT	The maximum number of Level 1 Folders to retrieve on each Query of Documange. Multiple Queries of Documange may be required to accumulate all (MaxHits) of the results for DmgBrsQuery if Folders at Level 1 are being returned. Infrequently needed. Can be used to tune performance. Optional. Default is 30.
SHOW_DOCS_IND	If this is 1, then each Folder record will include DOCS_IND to indicate whether the Folder contains Documents. This has a performance cost. 0 means no DOCS_IND. Optional. Default is the INI option [DocsIndicator] ShowDocsIndicator.

SHOW_NOTES_IND	If this is 1, then each Folder record will include NOTES_IND to indicate whether the Folder contains Notes. 0 means no NOTES_IND. Optional. Default is the INI option [Notes] ShowNotesIndicator.
STARTAT	The spot at which to begin returning results. Typically you will pass in one of the STARTAT variables passed out. See explanation below. Optional. Default is empty.
JUMP_TO_DOCS	Set this to 1 to cause DmgBrsQuery to begin returning results starting with the first Document found (if any) within a Folder or, if STARTAT specified starting at a Document hit, then whatever Document hit STARTAT specified. The results may be empty if, for example, there are no Documents in the Folder. Set this variable to 0 or omit it to avoid jumping to Documents. Has no effect when returning Folder results from the contents of the root. Optional. Default is 0.
FLDS_ONLY	Set this to 1 to allow only Folder hits to be returned, else to 0 to allow both Folder and Document hits to be returned. If this is 1, the output variable DHITS_EXIST_PL is always -1. This may be considered an additional type of filter. By preventing navigation into the Documents, it improves performance just a bit in those cases where Folders are the only concern. This over-rides JUMP_TO_DOCS. It also over-rides STARTAT if STARTAT says to start at a Document. In this latter case, hits will be returned starting with the first Folder in the Folder. Otherwise, hits will be returned starting with the Folder that STARTAT designates. Optional. Default is the value of the INI option [Query] FldsOnly.

Attachment outputs

Variable	Description
CURRENT_PL	The current parent Level. Ranges from 0 to the highest Level number in the Cabinet. The root of a Cabinet is defined as the parent of the Folders at Level 1. A CURRENT_PL of 0 means that the root of the Cabinet is the parent of the hits, 1 means that a Folder at Level 1 is the parent of the hits, etc. So, if the parent Level is 0, all of the hits will be children of the root of the Cabinet, i.e., the Folders at Level 1. If the parent Level is 2, all of the hits will be children of a Folder at Level 2, i.e., Folders at Level 3 or Documents in the Level 2 Folder.
DHITS_EXIST_PL	The parent Level at which Document hits (DHITS...) exist. -1 means there are no Document hits. If hits exist, then this number will be equal to CURRENT_PL.
FHITS_EXIST_PL	The parent Level at which Folder hits (FHITS...) exist. -1 means there are no folder hits. If hits exist, then this number will be equal to CURRENT_PL.

DHITS_PLx_	Contains the number of Document hits returned, where x is equal to CURRENT_PL. Exists only if DHITS_EXIST_PL is greater than -1. Note that x can never be 0 here because the root of a Cabinet can never be the parent of Documents.
DHITS_PLx_y.field	A record set describing the Documents that were found, where x is the parent Level, y is the record number, and field is one of the following fields. Exists only if DHITS_EXIST_PL is greater than -1.
.DOC_ID	The Id Property of the Document.
.DOC_MAJORVERSION	The MajorVersion Property of the Document.
.DOC_MINORVERSION	The MinorVersion Property of the Document.
.DOC_CATEGORY	The Category Property of the Document.
.DOCSPEC	The DOCSPEC of the Document.
.DOC_ISCOMPOUND	Contains 1 if the Document is a Compound Document, else contains 0.
.DOC_ <i>propertyname</i>	Variables containing other Basic Document Properties of the Document listed in the PS_DOC Property Set of the Property Set List, if any, or all these Properties if no Property Set List was used.
.XDOC_ <i>propertyname</i>	Variables containing Extended Document Properties of the Document listed in the PS_XDOC Property Set of the Property Set List, if any, where <i>propertyname</i> is the Bridge-encoded Extended Document Property Name (not DB Field Name), or all these Properties if no Property Set List was used.
.CONVNAME	A URL Tail with the Document's Name and converted FileType.
.ORIGNAME	A URL Tail with the Document's Name and FileType.
.EVEN	Contains 1 if the number of this hit record is even, else contains 0. Can be useful in formatting a table of results when using CGI templates.

FLD_ID_Lx	A set of variables containing the Id Property of the Folder at each ancestor Level, where x is a Level (not a parent Level) number. x ranges from 1 to CURRENT_PL. So, if the hits are from Level 1 (CURRENT_PL = 0), none of these variables will appear because Folders in Level 1 do not have Folder ancestors. If the hits are from level 2, then one of these variables will appear for x = 1. If the hits are from Level 3, then one of these variables will appear for x = 1 and one for x = 2, and so forth.
FLD_KEYSTRING_Lx	A set of variables containing the KeyString Property of the Folder at each ancestor Level, where x is a Level (not a parent Level) number. See FLD_ID_Lx.
FLD_LABEL_Lx	A set of variables containing the Label Property of the Folder at each ancestor Level, where x is a Level (not a parent Level) number. See FLD_ID_Lx.
XFLD_Lx_propertyname	A set of variables containing the values of the Extended Folder Properties requested in the corresponding PS_XFLD_Lx Property Sets, if any, of the Folder at each ancestor Level, or all of these Properties if no Property Set List was used. See FLD_ID_Lx.
FHITS_PLx_	Contains the number of Folder hits returned, where x is equal to CURRENT_PL. Exists only if FHITS_EXIST_PL is greater than -1. For example, if x is 0, then the Folders are children of the root, i.e., Level 1 Folders.
FHITS_PLx_y.field	A record set describing the Folders that were found, where x is the parent Level, y is the record number, and field is one of the following fields. Exists only if FHITS_EXIST_PL is greater than -1.
.STARTAT_FLD	The value to append to QUERY_STEM to create a URL that returns a complete result set beginning with the start of the contents of this Folder. See the examples.
.XFLD_Lx_propertyname	A set of variables containing the values of the Extended Folder Properties requested in the corresponding PS_XFLD_Lx Property Set, where x is the level of this Folder and propertyname is the Bridge-encoded Folder Property Name (not DB Field Name), or all these Properties if no Property Set List was used. These variables apply to this Folder only. See XFLD_Lx_propertyname below, separate from this record set, for variables that return the Extended Folder Properties of the ancestors of this Folder.

.DOCS_IND	Contains 1 if there are Documents in this Folder that match whatever Document filters are in effect, else contains 0. Exists only if SHOW_DOCS_IND contains 1.
.NOTES_IND	Contains 1 if there are Notes associated with this Folder, else contains 0. Exists only if SHOW_NOTES_IND contains 1.
.EVEN	Contains 1 if the number of this hit record is even, else contains 0. Can be useful in formatting a table of results when using CGI templates.
.FLDS_IND	Contains 1 if there are Folders in this Folder, else contains 0.
.FLD_ID	The Id Property of this Folder.
.FLD_KEYSTRING	The KeyString Property of this Folder.
.FLD_LABEL	The Label Property of this Folder.
HASDOCS	Contains 1 if there are Documents in the current parent Folder that match whatever Document filters are in effect, else contains 0. Always contains 0 when the contents of parent Level 0 are being returned.
HITSTYPE	The type of hits returned. 0 means no hits were returned, 1 means only Folder hits were returned, 2 means only Document hits were returned, 3 means both Folder and Document hits were returned. This makes it convenient for the application developer to display a notice like "No hits found", or to determine when a JumpToDocs control for the current level might be most appropriate (such as when there are Folder but no Document hits in the results returned, but Documents do exist in the Folder).
FLD_FILTER	The Extended Folder Property filter. May be empty. If QUERY_XFLD_ variables were used to create a filter, this will contain that filter.
DOC_FILTER	The Basic Document Property filter. May be empty. If QUERY_DOC_ variables were used to create a filter, this will contain that filter.
XDOC_FILTER	The Extended Document Property filter. May be empty. If QUERY_XDOC_ variables were used to create a filter, this will contain that filter.
LEVEL1_FLD_COUNT	The total number of Folders at level 1 of the Cabinet that match the search criteria, as returned by the dmapi function DmgResultGetTopLevelFolderCount().

MOREHITS	Contains 1 if there are more hits of any type and in the current parent Level after the ones that have been returned in this transaction, else contains 0.
PREVHITS	Contains 1 if there are previous hits of any type and in the current parent Level after the ones that have been returned in this transaction, else contains 0.
QUERY_STEM	A variable that provides convenience in calling DmgBrsQuery again with the same settings and filters. See the explanation under Remarks, and the examples.
REACHED_FLD_FETCH_LIMIT	Exists only if MOREHITS contains 0. Contains 1 if MOREHITS is 0 because the Folder Fetch Limit has been reached, else contains 0. The Folder Fetch Limit is a system option that may be set with the Documanager Administrator.
STARTAT_PLx	A set of variables providing STARTAT values for the ancestor Folders in the Cabinet, to make navigation to previous Levels easy. The maximum value of x for any given transaction will be CURRENT_PL - 1, when CURRENT_PL is at least 1. If CURRENT_PL is 0, then there are no STARTAT_PLx variables.
STARTAT_NEXT	A STARTAT value to get more hits or to jump to Documents at the current Level. To get more hits, combine the value of BRIDGEURL and QUERY_STEM with STARTAT = STARTAT_NEXT. For the typical use of jumping to Documents, combine the value of BRIDGEURL and QUERY_STEM with STARTAT = STARTAT_NEXT and add JUMP_TO_DOCS = 1. Exists only if MOREHITS contains 1.
STARTAT_PREV	A STARTAT value to get the previous MAXHITS hits for the current level. Combine the value of BRIDGEURL and QUERY_STEM with STARTAT = STARTAT_PREV. Exists only if PREVHITS contains 1.
STARTAT_TOP	A STARTAT value to return to the top of the hits list for the current level, i.e., the beginning of the contents of the current Folder or the beginning of the contents of the root of the Cabinet, as filtered. To jump to the top, combine the value of BRIDGEURL and QUERY_STEM with STARTAT = STARTAT_TOP. This variable always exists.

Remarks

DmgBrsQuery searches for Folders and Documents within a Folder.

SEARCH METHODS IN THE BRIDGE TO DOCUMANAGE

There are two major searching rules in the Bridge to Documanager for searching non-Workflow Cabinets: DmgBrsQuery and DmgBrsQueryDocuments. DmgBrsQueryDocuments returns a

list of Documents across Folders within a Cabinet. DmgBrsQuery returns a list of the Folders at the root of a Cabinet, or a list of the contents of a Folder.

The contents of a Folder are Documents and, in the case of a Cabinet with more than one level, Folders. The Documents that DmgBrsQuery returns always belong to a particular Folder. The way that DmgBrsQuery organizes its results is similar to the way the Documange Workstation presents Folders and Documents. The way that DmgBrsQueryDocuments organizes its results has no analog in the Documange Workstation.

Other rules are provided to find Projects and Documents in Workflow Cabinets. To search for Projects, use DmgBrsGetTaskQueue. To search for the Documents in a Project, use DmgBrsPresentProject.

SEARCHING WITH DMGBRSQUERY

DmgBrsQuery allows navigation through the Folders in a Cabinet. DmgBrsQuery could, for example, be used to get a list of Folders at Level 1 of a Cabinet. The user could then select one of those Folders and DmgBrsQuery could (by passing in the STARTAT for the Folder) be used to return the contents of the Folder. If the Cabinet had a second Level, then one or more Folders at Level 2 would be among the contents of the Folder at Level 1. The user could select one of the Level 2 Folders, and DmgBrsQuery could (by passing in the STARTAT for the Level 2 Folder) be used to return the contents of the Level 2 Folder. If there were no more Levels in the Cabinet, those contents would be zero or more Documents. Notice that the outputs of DmgBrsQuery also include a way to navigate to the parent of a Folder. So, if a user had just requested the contents of the Level 2 Folder, the STARTAT_PL1 variable would be available and could be passed to DmgBrsQuery to cause the contents of the Level 1 Folder to be returned again. In this way, a user can navigate into a Cabinet, and up and down the hierarchy of a Cabinet.

DmgBrsQuery can return a list of Folders, a list of Documents, or both. Folders and Documents are always returned on separate lists. HITSTYPE tells you exactly what lists were returned for any given transaction. If a Folder contains both Folders and Documents and both hits types are to be returned, then DmgBrsQuery will return Folder hits first until it runs out of Folders or reaches MaxHits. If it runs out of Folders before reaching MaxHits, then it will return Documents until it runs out of Documents or reaches MaxHits (total for Folders and Documents).

DmgBrsQuery returns Basic and Extended Properties for the ancestor Folders of the hits. For example, if the hits were the contents of a Folder at Level 3, then the Properties for the Folder at Level 3, the Properties for the Folder at Level 2 that contained the Folder at Level 3, and the Properties for the Folder at Level 1 that contained the Folder at Level 2 would all be returned. If the hits were the Folders at Level 1 (i.e., the children of the root of the Cabinet), then no ancestor data would be returned because Folders at Level 1 have no ancestor Folders.

STARTAT

DmgBrsQuery returns a set of STARTAT variables that can be used to navigate to any ancestor Folder (STARTAT_PLx), to the next batch of hits in the current Folder or at Level 1, if there are more (STARTAT_NEXT), to the previous batch of hits in the current Folder or at Level 1, if there are previous hits (STARTAT_PREV), to the beginning of the contents of the current Folder or of Level 1 (STARTAT_TOP), and to the contents of a Folder in the FHITS list (STARTAT_FLD). To refresh the results, the value of STARTAT passed in (which will be returned because of DmgBrsCopyAttachment) can be passed in again. See the examples below.

Since DmgBrsQuery generally returns the STARTAT variables you need, you typically won't need to be concerned with how the value of a STARTAT is constructed. In case you want to formulate your own STARTAT variables, the format is as follows:

```
[PLevel[, StartingIndex[, Keysting]]]
```

There are three comma delimited fields within STARTAT. All may be absent.

- Field 1: PLevel. The parent Level of the query results to be returned. A value of 0 means "return contents of the root". The default is 0.
- Field 2: StartingIndex. Fn|Dn. This field will be either the letter F (for Folder) followed by a number which is the index within the parent at which to start returning Folders, or the letter D (for Document) followed by a number which is the index within the parent Folder at which to start returning Documents. If PLevel is 0, then Field 2 will be the Fn indicating the StartFolder of the Documange Query and Field 3 will normally be empty. The default is F0.
- Field 3: Keystring. The Keystring Property of a Folder. The Keystring will be for a folder at level PLevel, or will be empty when PLevel is 0. This KeyString determines which Folder at PLevel is the parent Folder whose contents will be returned. The default is empty.

QUERY_STEM

The QUERY_STEM that DmgBrsQuery returns is for convenience in building URLs to call DmgBrsQuery again to continue navigating through results with the same settings and filters. The QUERY_STEM contains all non-empty inputs listed below in the format of an http query. Any filters built with SQL Builder are included in the QUERY_STEM. BRIDGEURL plus this stem and REQTYPE is a valid URL that points to the start of a search with DmgBrsQuery. BRIDGEURL plus this stem and REQTYPE and any valid STARTAT value begins returning results from DmgBrsQuery at the spot indicated by the STARTAT. All text parts of QUERY_STEM are URL encoded.

The inputs included in QUERY_STEM for DmgBrsQuery are:

```
CABINET
CAB_FILTER_variablename variables
FLD_FILTER
DOC_FILTER
XDOC_FILTER
FLD_ORDERBY
DOC_ORDERBY
FULLTEXT_FILTER
DOC_CATEGORY
PS_LIST
MAXHITS
MAXFLDCOUNT
SHOW_DOCS_IND
SHOW_NOTES_IND
```

For example, the value of a QUERY_STEM when the PS_LIST and MAXHITS variables were passed in might be as follows:

```
CABINET=My%20Cabinet&PS_LIST=QueryPSList&MAXHITS=10
```

See the examples for more information about using QUERY_STEM.

Examples

- 1 In this example, we'll search a Cabinet with two Levels. We start by returning a list of Folders at Level 1. Assume that there are 79 Folders in Level 1 of the Cabinet. Assume that [Query] MaxHits is 20, [DocsIndicator] ShowDocsIndicator is OFF, [Notes] ShowNotesIndicator is OFF. We're going to do an unfiltered search. A Property Set List called "QueryDefault" exists. It has a PS_XDOC Property Set for Category Photos with one entry and a PS_XFLD_L1 Property Set with one entry. Our only input attachment variable is CABINET = My2LevelCab.

Here is what the output attachment variables might look like.

```
CURRENT_PL = 0
```

```
DHITS_EXIST_PL = -1
DOC_FILTER =
FHITS_EXIST_PL = 0
FHITS_PL0_ = 20
FHITS_PL0_1.STARTAT_FLD = 1,F0,DB.dbo.Table.LName = 'Jones'
FHITS_PL0_1.XFLD_L1_Last$20Name = Jones
FHITS_PL0_1.EVEN = 0
FHITS_PL0_1.FLDS_IND = 1
FHITS_PL0_1.FLD_ID = 237
FHITS_PL0_1.FLD_KEYSTRING = DB.dbo.Table.LName = 'Jones'
FHITS_PL0_1.FLD_LABEL = Jones
...[plus records for FHITS_PL0_2 through FHITS_PL0_20]
FLD_FILTER =
HASDOCS = 1
HITSTYPE = 1
LEVEL1_FLD_COUNT = 79
MOREHITS = 1
PREVHITS = 0
QUERY_STEM = CABINET=My2LevelCab
STARTAT_NEXT = 0,F20
STARTAT_TOP = 0,F0
XDOC_FILTER =
```

In the above, notice that no Documents are returned. The root of the Cabinet has no Documents in it. Notice also that there are no Basic (FLD_...) or Extended (XFLD_Lx_propertyname) Properties and no STARTAT_PLx variables returned for ancestor Folders because there are no ancestor Folders of the Folders at Level 1. XFLD_L1_Last\$20Name is the Extended Folder Property in the PS_XFLD_L1 Property Set. There is no DOCS_IND or NOTES_IND because they are turned off.

The above output is the same as the output would be if our Cabinet had only 1 Level, but with one exception: the FLDS_IND field of the Folder hits records would always be 0 for a single Level Cabinet because the Folders at Level 1 of such a Cabinet never contain Folders.

- 2 In this example, we'll continue with the situation from Example 1, with the outputs returned from that transaction. Suppose that the user wants to see the next batch of Folders at Level 1. Since MOREHITS = 1, we've provided a link on the HTML page as illustrated in the following snippet from an HTML template. BridgeURL comes from DmgBrsValidateSession.

```
<A
HREF=#BRIDGEURL,#&REQTYPE=Query&#QUERY_STEM,#&STARTAT=#STARTAT_NEXT
,#>GET MORE</A>
```

Notice that we basically just combined QUERY_STEM with STARTAT_NEXT to get the next batch of hits. The output attachment variables might look something like the following.

```
CURRENT_PL = 0
DHITS_EXIST_PL = -1
DOC_FILTER =
FHITS_EXIST_PL = 0
FHITS_PL0_ = 20
FHITS_PL0_1.STARTAT_FLD = 1,F0,DB.dbo.Table.LName = 'Smith'
FHITS_PL0_1.XFLD_L1_Last$20Name = Smith
FHITS_PL0_1.EVEN = 0
FHITS_PL0_1.FLDS_IND = 1
```

```

FHITS_PL0_1.FLD_ID = 237
FHITS_PL0_1.FLD_KEYSTRING = DB.dbo.Table.LName = 'Smith'
FHITS_PL0_1.FLD_LABEL = Smith
...[plus records for FHITS_PL0_2 through FHITS_PL0_20]
FLD_FILTER =
HASDOCS = 1
HITSTYPE = 1
LEVEL1_FLD_COUNT = 79
MOREHITS = 1
PREVHITS = 1
QUERY_STEM = CABINET=My2LevelCab
STARTAT_NEXT = 0,F40
STARTAT_PREV = 0,F0
STARTAT_TOP = 0,F0
XDOC_FILTER =

```

The main differences between this set of outputs and the set in Example 1 are the fact that the actual hits list is for the next 20 Folders and that we now have hits previous to this batch. Because we have previous hits, PREVHITS = 1 and STARTAT_PREV exists.

- 3 In this example, we'll continue with the situation from Example 2, with the outputs returned from that transaction. Suppose now that the user wants to get a listing of the contents of the first Folder returned in Example 2. To allow the user to do this, we've provided a link on the HTML page for each Folder hit record as illustrated in the following snippet from an HTML template. It allows the user to click on the Last Name Extended Property of the Folder and get a list of the contents. BridgeURL comes from DmgBrsValidateSession.

```

<A
HREF=#BRIDGEURL,#&REQTYPE=Query&#QUERY_STEM,#&STARTAT=#STARTAT_FLD,
#>#XFLD_L1_Last$20Name,#</A>

```

Assume that the Folder contains one Folder and 33 Documents. The Folder it contains will, of course, be at Level 2. The output attachment variables might look like this.

```

CURRENT_PL = 1
DHITS_EXIST_PL = 1
DHITS_PL1_ = 19
DHITS_PL1_1.CONVNAME = /Homeowner.pdf
DHITS_PL1_1.DOC_CATEGORY = Policy
DHITS_PL1_1.DOC_DATE =
DHITS_PL1_1.DOC_ID = 829
DHITS_PL1_1.DOC_MAJORVERSION = 1
DHITS_PL1_1.DOC_MINORVERSION = 0
DHITS_PL1_1.DOC_NAME = Homeowner
DHITS_PL1_1.DOCSPEC = CABINET= My2LevelCab &
DOC_ID=829&DOC_VERS=1.0
DHITS_PL1_1.EVEN = 0
DHITS_PL1_1.ORIGNAME = /Homeowner.met
... [plus records for DHITS_PL1_2 through DHITS_PL0_19]
DOC_FILTER =
FHITS_EXIST_PL = 1
FHITS_PL1_ = 1
FHITS_PL1_1.EVEN = 0
FHITS_PL1_1.FLDS_IND = 0
FHITS_PL1_1.FLD_ID = 88347
FHITS_PL1_1.FLD_KEYSTRING = DB.dbo.Table2.ID = 'A47'
FHITS_PL1_1.FLD_LABEL = A47

```

```
FHITS_PL1_1.STARTAT_FLD = 2,F0,DB.dbo.Table2.ID = 'A47'  
FOLDER_FILTER =  
FLD_ID_L1 = 2372  
FLD_KEYSTRING_L1 = DB.dbo.Table.LName = 'Smith'  
FLD_LABEL_L1 = Smith  
HASDOCS = 1  
HITSTYPE = 3  
MOREHITS = 1  
PREVHITS = 0  
QUERY_STEM = CABINET= My2LevelCab  
STARTAT_PL0 = 0,F20  
STARTAT_NEXT = 1,D19,DB.dbo.Table.LName = 'Smith'  
STARTAT_TOP = 1,F0,DB.dbo.Table.LName = 'Smith'  
XDOC_FILTER =  
XFLD_L1_Last$20Name = Smith
```

In the above, notice that the parent level is now 1 since we are looking at the contents of a Folder from Level 1.

We have two hits lists, one for Folders and one for Documents. We have 1 Folder hit for the Folder at Level 2 that the Folder at Level 1 contains. Then we have 19 Document hits, to make our total of 20 maximum hits to return. Since there are 33 Documents in the Folder and we only had room in this transaction for 19 of them, MOREHITS is 1 and STARTAT_NEXT indicates to start at the 20th Document (index D19). If we executed a search using this STARTAT_NEXT, the output attachment variables would not have any Folder hits. We would get the next 14 Documents.

Note that we don't see any Extended Document Properties for the example Document shown in the hits list above because it is not in the Category for which we have a PS_XDOC Property Set.

Because this Folder at Level 2 has an ancestor Folder at Level 1, we now see Properties and a STARTAT variable for that ancestor. The Properties are the Basic Properties FLD_ID_L1, FLD_KEYSTRING_L1 and FLD_LABEL_L1, and the Extended Property from the PS_XFLD_L1 Property Set, XFLD_L1_Last\$20Name.

- 4 In this example, we'll again continue with the situation from Example 2, with the outputs returned from that transaction. Suppose this time that the user wants to get a listing of just the Documents of the first Folder returned in Example 2. To allow the user to do this, we've provided a link on the HTML page for each Folder hit record as illustrated in the following snippet from an HTML template. It allows the user to click on the Last Name Extended Property of the Folder and get a list of the Documents in the Folder, if any. This might be especially handy if the Folder contained many Folders that the user would otherwise have to page through to reach the Documents. BridgeURL comes from DmgBrValidateSession.

```
<A  
HREF=#BRIDGEURL,#&REQTYPE=Query&#QUERY_STEM,#&STARTAT=#STARTAT_FLD,  
#&JUMP_TO_DOCS=1># XFLD_L1_Last$20Name,# -- Documents</A>
```

The output attachment variables would be very similar to those for Example 3, except that there would be no Folder hits list, and there would be 20 hits on the Document hits list.

- 5 In this example, we'll see the use of DOC_IND. We'll start with the situation in Example 1, but with one change: assume that [DocsIndicator] ShowDocsIndicator is ON. This will allow us to tell the user when a Folder in the hits lists contains Documents.

In this example, the output attachment variables would contain the additional variable DOCS_IND for each Folder hit record, and it would contain 1 if the Folder contained Documents, else 0. Let's assume our first Folder hit had this output variable.

```
FHITS_PL0_1.DOCS_IND = 1
```

Suppose we want to use an asterisk next to the LastName to tell a user that the Folder contained Documents. We might set up a rules list like the following.

```
[ ReqType:QUERY ]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsSetIndicators, FHITS_PL0_*.DOCS_IND, *
  function = dmg_brw32->DmgBrsQuery
```

The rules list would cause the value of each of the DOCS_IND variables in the FHITS_PL0_record set to be replaced with * if it contained 1 and empty if it contained 0.

The snippet from the HTML template shown in Example 3 could be modified as follows.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=Query&#QUERY_STEM, #&STARTAT=#STARTAT_FLD,
  #>#XFLD_L1_Last$20Name, ##DOCS_IND, #</A>
```

If the Folder contained Documents, the link would look like this when displayed on the page: Jones*. If the Folder didn't contain Documents, the link would look like this: Smith.

- 6 In this example, we'll see various ways that the results of DmgBrsQuery can be used to link to other rules. We'll illustrate the links with snippets from a hypothetical HTML template. BridgeURL comes from DmgBrsValidateSession.

To link a Folder on a Folder hits list to DmgBrsPresentFolder, the following could be used. Notice that, in the context of output attachment variables from DmgBrsQuery, we always know what Folder Level we're working with because of the way the variables are named.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=PresentFld&CABINET=#CABINET#&FLD_LEVELNUM
  BER=2&FLD_ID=#FLD_ID, #>Folder Info</A>
```

To link a Document on a Document hits list to DmgBrsReturnConvertedDocument, the following could be used.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=ConvertDoc&#DOCSPEC, #&CONVNAME=#CONVNAME,
  #>#DOC_NAME, #. PDF</A>
```

To link a Document on a Document hits list to DmgBrsReturnDocument, the following could be used.

```
<A
  HREF=#BRIDGEURL, #&REQTYPE=DocFile&#DOCSPEC, #&ORIGNAME=#ORIGNAME, #>#
  DOC_NAME, #. #DOC_FILETYPE, #</A>
```

To link a Document on a Document hits list to DmgBrsPresentDocument, the following could be used.

```
<A  
  HREF=#BRIDGEURL, #&REQTYPE=PresentDoc&#DOCSPEC, #&SHOW_PAGES=1>Docume  
nt Info</A>
```

To link a Document on a Document hits list to DmgBrsPresentDocumentPage, the following could be used.

```
<A HREF=#BRIDGEURL, #&REQTYPE=PresentPg&#DOCSPEC, #&PAGE=0>Document  
Info</A>
```

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	[Query] [DocsIndicator] ShowDocsIndicator [Notes] ShowNotesIndicator
Validations	Session and Reqtype
Platform restrictions	None.
See also	DmgBrsGetTaskQueue DmgBrsLogPropertySets DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsPresentProject DmgBrsPresentDocumentPage DmgBrsPresentProject DmgBrsQueryDiaries DmgBrsQueryDocuments DmgBrsReturnConvertedDocument DmgBrsReturnDocument DmgBrsSetIndicators

DMGBRSQUERYALLPROJECTSEX

DmgBrsQueryAllProjectsEx searches for a user's Projects across all or many Workflow Cabinets, without regard to Task.

Usage

```
function = dmg_brw32->DmgBrsQueryAllProjectsEx
```

Attachment inputs

Variable	Description
WF_CABINET_LIST	<p>This optional variable allows the list of Workflow cabinets to be searched to be limited. Its value is the name of a section in the Property Set file (see the INI option [PropertySets] PropertySetFile, described in the General Reference).</p> <p>This section must contain a list of the Bridge-encoded names of the Workflow Cabinets to be searched and, optionally, associated Property Set Lists. Optional.</p> <p>If missing, DmgBrsQueryAllProjectsEx looks for a section called "WFCabinetListDefault". If that section can't be found, or if this variable doesn't exist, then all Workflow Cabinets will be searched, and there is no way to limit the output with Property Sets.</p> <p>See below for details and an example.</p>
MAXHITS	<p>The number of Project hits allowed to be returned in one transaction. Limited by the INI option [QueryAllProjects] MaxHits. Optional. Default is the INI option [QueryAllProjects] MaxHits.</p>
CHECKEDOUT_STATE	<p>A filter that will be ANDed with PROJECT_STATE, if any. Optional. The default is the INI option [QueryAllProjects] CheckedoutState. This has three recognized values, case independent:</p> <p>ANY — return Projects without regard to their checked out state.</p> <p>CHECKED_OUT — return only those Projects that are checked out.</p> <p>NOT_CHECKED_OUT — return only those Projects that are not checked out.</p>
PROJECT_STATE	<p>A filter that will be ANDed with CHECKED_OUT_STATE, if any. Optional. The default is the INI option [QueryAllProjects] ProjectState. This has three recognized values, case independent:</p> <p>ANY — return Projects without regard to their Project state.</p> <p>PENDING — return only those Projects that are Pending.</p> <p>SUSPENDED — return only those Projects that are Suspended.</p>
SHOW_DOCS_IND	<p>If this is 1, then each Project record will include DOCS_IND to indicate whether the Project contains Documents. This has a performance cost. 0 means no DOCS_IND. Optional. Default is the INI option [DocsIndicator] ShowDocsIndicator.</p>
SHOW_NOTES_IND	<p>If this is 1, then each Project record will include NOTES_IND to indicate whether the Project contains Notes. 0 means no NOTES_IND. Optional. Default is the INI option [Notes] ShowNotesIndicator.</p>

Attachment outputs

Variable	Description
PROJECTS_EXIST	Contains 1 if any Projects were found for any Workflow Cabinet, else contains 0. Note that this does not indicate whether Projects exist for the user. It merely indicates whether any were found given any filters that were passed in. Projects may exist for the user even if none were found after applying the filters.
CABINETS	The number of Workflow Cabinets for which Projects are being returned. The Projects are grouped by Cabinet. This variable contains 0 if PROJECTS_EXIST contains 0.
CABINETSx.field	A record set describing the Cabinets for which Projects are being returned, where x is the record number and field is one of the following fields. The Projects themselves are nested within each Cabinet record.
.NAME	The Name of the Workflow Cabinet.
.NESTED	The name of the record set (row set) containing the Projects for this Workflow Cabinet. Always use this name to locate the Projects, since the name used for the record set may change in future releases of the Bridge.
[.NESTED value]	This is a variable whose name is the value of .NESTED for the Workflow Cabinet, and whose value is the number of Projects being returned for this Workflow Cabinet.
[.NESTED value]y.field	A record set describing the Projects being returned for Workflow Cabinet x, where y is the record number and field is one of the following fields. Note that this record set is nested within one CABINETS record, so that each variable name is prepended with the value of .NESTED for the Workflow Cabinet.
[.NESTED value]y.XPRJ_propertyname	A set of variables containing the values of the Extended Project Properties requested in the PS_XPRJ Property Set, if any, on the Property Set List associated with cabname, where propertyname is the Bridge-encoded Project Property Name (not DB Field Name). If there was no Property Set List, then all these Properties are returned.
[.NESTED value]y.DOCS_IND	Contains 1 if there are Documents in this Project, else contains 0. Exists only if SHOW_DOCS_IND contains 1.

[.NESTED value]y.NOTES_IND	Contains 1 if there are Notes associated with this Project, else contains 0. Exists only if SHOW_NOTES_IND contains 1.
[.NESTED value]y.PRJ_LABEL	The Label Property of this Project.
[.NESTED value]y.PRJ_ID	The ID Property of this Project.
[.NESTED value]y.PRJ_KEYSTRING	The Keysting Property of this Project.
[.NESTED value]y.PRJ_TASKTYPE	The numeric TaskType Property of this Project.
[.NESTED value]y.PRJ_TASKTYPE_TEXT	The TaskType Property of this Project, in text form, where the text matches the enum DmgTaskTypes in dmg_api_types.h.
[.NESTED value]y.PRJ_standardextendedproperty	A set of variables containing the values of the 19 Standard Extended Project Properties described in the Introduction.
MOREPROJECTS	Contains 1 if there are Projects after the ones that have been returned in this transaction, else contains 0.
REACHED_FLD_FETCH_LIMIT	Exists only if MOREPROJECTS contains 0. Contains 1 if MOREPROJECTS is 0 because the Folder Fetch Limit has been reached, else contains 0. The Folder Fetch Limit is a system option that may be set with the Documange Administrator.

DmgBrsQueryAllProjectsEx searches for a user's Projects across all or many Workflow Cabinets, without regard to Task.

This rule is similar to DmgBrsQueryProjectsEx, except that it does not provide paging through the results and, of course, it searches multiple Cabinets.

Note that there are no required input attachment variables for DmgBrsQueryAllProjects. With WF_CABINET_LIST, you may optionally limit the number of Workflow Cabinets that are searched. You may also request automatic filtering by Suspended or Pending, and/or checked out or not checked out Projects, just as you can with DmgBrsQueryProjectsEx.

The input attachment variable WF_CABINET_LIST provides a way to specify which Workflow Cabinets to search and, optionally, to provide a Property Set List for each one. Separate Property Set Lists must be specified for each Cabinet to be searched, when Property Set Lists are desired. Suppose the value of WF_CABINET_LIST is "MyWFCabList". DmgBrsQueryAllProjectsEx would look in the configured PropertySetFile for a section named MyWFCabList. Assume the Documange system has a Workflow Cabinet called "WF Cabinet One" and another called "WFCabinetTwo". The section might look like the following:

```
[MyWFCabList]
WF$20Cabinet$20One = WFCabPropsetList
WFCabinetTwo
```

Notice the following characteristics of the entries in the section:

- The Workflow Cabinet names are Bridge encoded.
- Providing a Property Set List for any given Cabinet is optional.

Cabinet names that contain the “=” sign are not supported on Workflow Cabinet lists.

DmgBrsQueryAllProjectsEx doesn’t return lists of Documents in the Projects. To see the Documents in a given Project, use DmgBrsPresentProject.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [QueryAllProjects]
[DocsIndicator] ShowDocsIndicator
[Notes] ShowNotesIndicator

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsQueryProjects
DmgBrsGetTaskQueue
DmgBrsPresentProject
DmgBrsQuery

DMGBRSQUERYDIARIES

DmgBrsQueryDiaries searches for a user's Diaries and presents information about them.

Usage

```
function = dmg_brw32->DmgBrsQueryDiaries
```

Attachment inputs

Variable	Description
DIARY_FILTER	A filter for Diaries. To return information about a single Diary, simply pass in a value from DIA_KEYSTRING. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional. Default is empty.
QUERY_DIA_...	SQL Builder variables describing a Diary filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DIARY_FILTER, and will be ignored if DIARY_FILTER is passed in. Optional.
DIARY_ORDERBY	An order-by SQL clause for Diaries. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional. Default is empty.
OBn_DIA_...	SQL Builder variables describing a Diary Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DIARY_ORDERBY, and will be ignored if DIARY_ORDERBY is passed in. Optional.
DIARY_INDEX	The zero-based index to start returning Diaries. Optional. Default is 0.
MAXDIARIES	The number of Diaries to return. Limited by the INI option [QueryDiaries] MaxDiaries. Optional. Default is the INI option [QueryDiaries] MaxDiaries.

Attachment outputs

Variable	Description
DIARIES_EXIST	Contains 1 if any Diaries were found, else contains 0. Note that this does not indicate whether Diaries exist for the current user. It merely indicates whether any were found given any filters and the DIARY_INDEX that were passed in. Diaries may exist even if none were found given the inputs.
DIARIES	The number of Diaries found. Will contain 0 if DIARIES_EXIST contains 0.
DIARIESx.field	A record set describing the Diaries that were found, where x is the record number and field is one of the following fields. Exists only if DIARIES is greater than 0.
.DIARY_TITLE	The Title Property of the Diary.
.DIA_DESCRIPTION	The Description Property of the Diary.
.DIA_PRIORITY	The Priority Property of the Diary.

.DIA_CREATEDON	The CreatedOn Property of the Diary.
.DIA_STARTON	The StartOn Property of the Diary.
.DIA_DUEON	The DueOn Property of the Diary.
.DIA_COMPLETEDON	The CompletedOn Property of the Diary.
.DIA_LASTMODIFIED ON	The LastModifiedOn Property of the Diary.
.DIA_NEXTALERTTI ME	The NextAlertTime Property of the Diary.
.DIA_ASSIGNEDBY	The AssignedBy Property of the Diary.
.DIA_OWNER	The Owner Property of the Diary.
.DIA_REFERENCE	The Reference Property of the Diary.
.DIA_STATUS	The Status Property of the Diary.
.DIA_ID	The ID Property of the Diary.
.DIA_KEYSTRING	The Keystring Property of the Diary.
MOREDIARIES	Contains 1 if there are Diaries after the ones that have been returned in this transaction, else contains 0.
DIARY_INDEX_NEXT	The index of the next batch of Diaries. Exists only if MOREDIARIES contains 1.
PREVDIARIES	Contains 1 if there are Diaries previous to the ones that have been returned in this transaction, else contains 0.
DIARY_INDEX_PREV	The index of the previous batch of Diaries. Exists only if PREVDIARIES contains 1.

Remarks DmgBrsQueryDiaries searches for a user's Diaries and presents information about them.

It may be used to present information about one Diary simply by passing in a DIA_KEYSTRING for DIARY_FILTER.

Note that all of the input attachment variables are optional. You can use this rule with no inputs and it will return a list of up to the number of Diaries in the INI option [QueryDiaries] MaxDiaries.

Examples With the rules list below, DmgBrsQueryDiaries does double duty as a way to search for Diaries and a way to present them. You could call DmgBrsQueryDiaries with no inputs to get a list of Diaries for the user. You could display to the user one or a limited number of indentifying Properties of each Diary on the list, such as its Title and/or Description. The user could choose a Diary from that list, and you could call DmgBrsQueryDiaries again, setting the value of DIARY_FILTER to the DIA_KEYSTRING Property of the Diary that the user chose. You could then present all of the relevant Properties of that Diary to the user.

```
[ReqType:DIARIES]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
```

```
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsQueryDiaries
```

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	[QueryDiaries]
Validations	Session and Reqtype
Platform restrictions	None.
See also	DmgBrsCreateDiary DmgBrsDeleteDiary DmgBrsUpdateDiary

DMGBRSQUERYDOCUMENTRENDITIONS

DmgBrsQueryDocumentRenditions returns all the Renditions of a Document.

UsageUsage

```
function = dmg_brw32->DmgBrsQueryDocumentRenditions
```

Attachment inputs

Variable	Description
A Version Document Specification	The Document Version whose Renditions are being requested. Required.
OBSOLETE_FILTER	A filter that will be ANDed with any other filters. Optional. The default is the INI option [QueryDocumentRenditions] ObsoleteFilter. This has three recognized values, case independent: NONE—return Renditions without regard to their Obsolete Property. 1 — return only those Renditions whose Obsolete Property is DmgTrue. 0 — return only those Renditions whose Obsolete Property is DmgFalse.
RELEASED_FILTER	A filter that will be ANDed with any other filters. Optional. The default is the INI option [QueryDocumentRenditions] ReleasedFilter. This has three recognized values, case independent: NONE — return Renditions without regard to their Released Property. 1 — return only those Renditions whose Released Property is DmgTrue. 0 — return only those Renditions whose Released Property is DmgFalse.
APPROVED_FILTER	A filter that will be ANDed with any other filters. Optional. The default is the INI option [QueryDocumentRenditions] ApprovedFilter. This has three recognized values, case independent: NONE — return Renditions without regard to their Approved Property. 1 — return only those Renditions whose Approved Property is DmgTrue. 0 — return only those Renditions whose Approved Property is DmgFalse.
LATEST_VERSIONS_ONLY	If this is 1, then for a Document with multiple Versions that belong to the Rendition set, the query will return only the latest Version. Optional. Default is the INI option [QueryDocumentRenditions] LatestVersionsOnly.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Cabinet is a Workflow Cabinet, else 0.

FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document in the input Version Document Specification. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document in the input Version Document Specification. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document in the input Version Document Specification. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document in the input Version Document Specification. Exists only if IS_WORKFLOW contains 1.
HITS_EXIST	Contains 1 if there were Renditions that matched the filters, else contains 0.
HITS	The number of Renditions found. Exists only if HITS_EXIST contains 1.
HITSx.field	A record set describing the Renditions that were found, where x is the record number and field is one of the following fields. Exists only if HITS is greater than 0.
.DOC_ID	The Id Property of the Rendition.
.DOC_MAJORVERSION	The MajorVersion Property of the Rendition.
.DOC_MINORVERSION	The MinorVersion Property of the Rendition.
.DOC_RENDITIONKEY	The RenditionKey Property of the Rendition.
.DOC_NAME	The Name Property of the Rendition.
.DOC_FILETYPE	The FileType Property of the Rendition.
.DOC_ISCOMPOUND	Contains 1 if the Document is a Compound Document, else contains 0.
.CABINET	The Cabinet label of the Rendition.

Remarks DmgBrsQueryDocumentRenditions returns all the Renditions of a Document.
This rule will fail if the Bridge is not connected to Documange Service Release 2 or higher.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [QueryDocumentRenditions]

Validations Session and Reqtype

Platform restrictions None.

- May be followed by
- DmgBrsPresentFolder
 - DmgBrsPresentDocument
- See also
- DmgBrsAddDocumentAsRendition
 - DmgBrsRemoveDocumentAsRendition
 - DmgBrsQuery
 - DmgBrsQueryDocuments

DMGBRSQUERYDOCUMENTS

DmgBrsQueryDocuments searches for Documents across Folders.

Usage

```
function = dmg_brw32->DmgBrsQueryDocuments
```

Attachment inputs

Variable	Description
CABINET	The name of the Cabinet to search. Required. Note that this is the only required input attachment variable for this rule, unless the Cabinet requires Cabinet filters to be searched. See next variable.
CAB_FILTER_variablename	A set of variables containing Cabinet filter terms, where variablename is the Bridge encoded name of a particular Cabinet filter variable, as defined for the Cabinet in the Documange Administrator application. Pass in as many as needed, if any, to open the Cabinet. Optional, unless required by the Cabinet definition.
FULLTEXT_FILTER	A full text filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
FLD_FILTER	An Extended Folder Property filter for the search. Optional. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini.
DOC_FILTER	A Basic Document Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
XDOC_FILTER	An Extended Document Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
VER_FILTER	A Document Version Basic Property filter for the search. This filter has no effect if the input attachment variable QUERY_VERSIONS contains zero. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
FLD_ORDERBY	A Folder order by SQL clause for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
DOC_ORDERBY	A Document order by SQL clause for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
DOC_CATEGORY	The Category Property for the Extended Document Property filter. Required if there is an Extended Document Property filter, else ignored.

QUERY_DOC_...	SQL Builder variables describing a Basic Document Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DOC_FILTER, and will be ignored if DOC_FILTER is passed in. Optional.
QUERY_VER_...	SQL Builder variables describing a Document Version Basic Property filter. The syntax for these variables is identical to the syntax for the QUERY_DOC_... variables. This is an alternative to VER_FILTER, and will be ignored if VER_FILTER is passed in. Optional.
QUERY_XDOC_...	SQL Builder variables describing an Extended Document Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to XDOC_FILTER, and will be ignored if XDOC_FILTER is passed in. Optional.
QUERY_XFLD_...	SQL Builder variables describing an Extended Folder Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to FLD_FILTER, and will be ignored if FLD_FILTER is passed in. Optional.
OBn_XFLD_...	SQL Builder variables describing a Folder Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to FLD_ORDERBY, and will be ignored if FLD_ORDERBY is passed in. Optional.
OBn_DOC_...	SQL Builder variables describing a Document Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to DOC_ORDERBY, and will be ignored if DOC_ORDERBY is passed in. Optional.
LEVEL_LIMITER	A Level number from 1 to the number of Levels in the Cabinet. Limits hits returned to that Level. Optional. When missing, LEVEL_MASK is used.
LEVEL_MASK	A bit mask with a bit set for each level for which hits are to be returned. Use LEVEL_MASK when you want to limit Document hits to a list of Levels. DmgBrsQueryDocuments handles up to 10 levels. For example, a value of 1 (first bit) would limit returned hits to Level 1. A value of 2 (second bit) would limit returned hits to Level 2. A value of 3 (first and second bits) would limit returned hits to Levels 1 and 2. A value of 4 (third bit) would limit returned hits to Level 3. And so on. Optional. Ignored if LEVEL_LIMITER exists. The default is 1023, for all 10 levels.
PS_LIST	A Property Set List to limit any Basic or Extended Document Properties to return for each Document found and any Extended Folder Properties to return for each Folder found. Uses PS_DOC, PS_XDOC, and PS_XFLD Property Sets. Optional. If absent, DmgBrsQueryDocuments will use a Property Set List called "QueryDocumentsDefault" if one exists.

QUERY_VERSIONS	Set this variable to 1 to cause this rule to apply the criteria to all Documents and return any Document Versions that match, rather than just the current version. Optional. Default is the INI option [QueryDocuments] QueryVersions.
MAXHITS	The total number of Document hits allowed to be returned in one transaction. Limited by the INI option [QueryDocuments] MaxHits. Optional. Default is the INI option [QueryDocuments] MaxHits.
MAXFLDCOUNT	The maximum number of Level 1 Folders to retrieve on each Query of Documanager. Multiple Queries of Documanager may be required to accumulate all (MaxHits) of the results for DmgBrsQueryDocuments. Infrequently needed. Can be used to tune performance. Optional. Default is 30.
STARTAT	The spot at which to begin returning results. Typically you will pass in the value of the STARTAT_NEXT variable passed out. See explanation below. Optional. Default is empty, which causes the hits to be returned starting with the first Document that matches the criteria.
GROUP_HITS_BY_LEVEL	Set this to 1 to cause hits from a multi-Level Cabinet to be returned in separate hits lists grouped by Level. Set it to 0 to cause all hits to appear in one hits list, regardless of Level. Optional. Default is [QueryDocuments] GroupHitsByLevel

Attachment outputs

Variable	Description
HITS_EXIST	Contains 1 if Documents were found, else contains 0.
HITS_EXIST_Lx	A set of variables, one for each Level of the Cabinet, that show whether hits exist at that Level, where x is the Level. Each variable contains 1 if Documents were found at that level, else contains 0.
HITS	The total number of hits across all Levels. Exists only if HITS_EXIST contains 1.
HITSx.field	A record set describing the Documents that were found, where x is the record number, and field is one of the following fields. Exists only if HITS_EXIST is 1 and GROUP_HITS_BY_LEVEL contains 0.
.DOC_ID	The Id Property of the Document.
.DOC_MAJORVERSION	The MajorVersion Property of the Document.
.DOC_MINORVERSION	The MinorVersion Property of the Document.
.DOC_CATEGORY	The Category Property of the Document.
.DOCSPEC	The DOCSPEC of the Document.
.DOC_ISCOMPOUND	Contains 1 if the Document is a Compound Document, else contains 0.

<i>.DOC_propertyname</i>	Variables containing other Basic Document Properties of the Document listed in the PS_DOC Property Set of the Property Set List, if any, or all these Properties if no Property Set List was used. These will be absent if none are listed.
<i>.XDOC_propertyname</i>	Variables containing Extended Document Properties of the Document listed in the PS_XDOC Property Set of the Property Set List, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name) , or all these Properties if no Property Set List was used.
.CONVNAME	A URL Tail with the Document's Name and converted FileType.
.ORIGNAME	A URL Tail with the Document's Name and FileType.
.EVEN	Contains 1 if the number of this hit record is even, else contains 0. Can be useful in formatting a table of results when using CGI templates.
<i>.XFLD_Lx_propertyname</i>	A set of variables containing the values of the Extended Folder Properties requested in the PS_XFLD_Lx Property Set (if any) for the Folder in which this Document was found, where x is the level and propertyname is the Bridge-encoded Folder Property Name (not DB Field Name) , or all these Properties if no Property Set List was used. The variables for Level x will be empty if x is greater than FLD_LEVELNUMBER.
.FLD_ID	The Id Property of the Folder in which this Document was found.
.FLD_KEYSTRING	The KeyString Property of the Folder in which this Document was found.
.FLD_LABEL	The Label Property of the Folder in which this Document was found.
.FLD_LEVELNUMBER	The LevelNumber Property of the Folder in which this Document was found.
HITS_Lx_	A set of variables, one for each Level of the Cabinet, containing the number of hits found at each Level, where x is the Level. These variables exist only if GROUP_HITS_BY_LEVEL contains 1. A variable for any given Level exists only if the corresponding HITS_EXIST_Lx contains 1.
HITS_Lx_y.field	A record set describing the Documents that were found, where x is the record number, and field is one of the following fields. Exists only if the corresponding HITS_Lx_ variable exists.
.DOC_ID	The Id Property of the Document.
.DOC_MAJORVERSION	The MajorVersion Property of the Document.
.DOC_MINORVERSION	The MinorVersion Property of the Document.

.DOC_CATEGORY	The Category Property of the Document.
.DOCSPEC	The DOCSPEC of the Document.
.DOC_ISCOMPOUND	Contains 1 if the Document is a Compound Document, else contains 0.
.DOC_propertyname	Variables containing other Basic Document Properties listed in the PS_DOC Property Set of the Property Set List, if any, or all these Properties if no Property Set List was used.
.XDOC_propertyname	Variables containing Extended Document Properties listed in the PS_XDOC Property Set of the Property Set List, if any, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name) , or all these Properties if no Property Set List was used.
.CONVNAME	A URL Tail with the Document's Name and converted FileType.
.ORIGNAME	A URL Tail with the Document's Name and FileType.
.EVEN	Contains 1 if the number of this hit record is even, else contains 0. Can be useful in formatting a table of results when using CGI templates.
.XFLD_Lx_propertyname	A set of variables containing the values of the Extended Folder Properties requested in the PS_XFLD_Lx Property Set (if any) for the Folder in which this Document was found, where x is the level and propertyname is the Bridge-encoded Folder Property Name (not DB Field Name) , or all these Properties if no Property Set List was used. The XFLD_Lx_propertyname variables for Level x will not exist if x is greater than x in HITS_Lx_y.field.
.FLD_ID	The Id Property of the Folder in which this Document was found.
.FLD_KEYSTRING	The KeyString Property of the Folder in which this Document was found.
.FLD_LABEL	The Label Property of the Folder in which this Document was found.
FLD_FILTER	The Extended Folder Property filter. May be empty. If QUERY_XFLD_ variables were used to create a filter, this will contain that filter.
DOC_FILTER	The Basic Document Property filter. May be empty. If QUERY_DOC_ variables were used to create a filter, this will contain that filter.
XDOC_FILTER	The Extended Document Property filter. May be empty. If QUERY_XDOC_ variables were used to create a filter, this will contain that filter.

LEVEL1_FLD_COUNT	The total number of Folders at level 1 of the Cabinet that match the search criteria, as returned by DmgResultGetTopLevelFolderCount().
QUERY_STEM	A variable that provides convenience in calling DmgBrsQueryDocuments again with the same settings and filters. See the explanation under Remarks, and the examples.
MOREHITS	Contains 1 if there are more hits after the ones that have been returned in this transaction, else contains 0.
REACHED_FLD_FETCH_LIMIT	Exists only if MOREHITS contains 0. Contains 1 if MOREHITS is 0 because the Folder Fetch Limit has been reached, else contains 0. The Folder Fetch Limit is a system option that may be set with the Documanager Administrator.
STARTAT_NEXT	A STARTAT value to get more hits. To get more hits, combine the value of BRIDGEURL and QUERY_STEM with STARTAT = STARTAT_NEXT. Exists only if MOREHITS contains 1.

Remarks

DmgBrsQueryDocuments searches for Documents across Folders. This is in contrast to DmgBrsQuery, which searches for Folders and Documents within a Folder. See the remarks for DmgBrsQuery for further discussion of the different search methods available in the Bridge to Documanager.

When DmgBrsQueryDocuments searches for Documents, it looks for Documents that fit the Document filters within Folders that fit the Folder filter. It looks for Documents in the first Level 1 Folder and all of its Folder descendants, if any, then in the second Level 1 Folder and all of its Folder descendants, if any, and continues in this fashion until it finds MaxHits Documents or runs out of Documents or Folders that match the filters.

DmgBrsQueryDocuments can return its results in either of two formats: one list, or multiple lists grouped by Level. The fields returned in the record sets for the two formats are identical with two exceptions. First, FLD_LEVELNUMBER does not appear if hits are grouped by Level. Second, the Extended Folder Properties (the XFLD_Lx_ propertyname variables) are different. When hits are grouped by Level, the Extended Folder Properties for Levels with higher numbers than the Level the Document was found in do not exist. When hits are not grouped by Level, they exist, but they are empty.

For each Document DmgBrsQueryDocuments finds, it returns Basic and Extended Document Properties, Basic and Extended Folder Properties for the Folder in which the Document was found, variables that make it convenient to create references to the Document's file or converted version, and variables that make it easy to get the next batch of hits, if any, that match the filters.

STARTAT

DmgBrsQueryDocuments returns a variable called STARTAT_NEXT that can be used to get the next batch of hits, if any. You typically won't need to be concerned with how the value of a STARTAT is constructed. In case you want to formulate your own STARTAT variable, the format is as follows:

```
[StartFolder[,StartingDocIndex
[,StartingFldIndexL1[,StartingFldIndexL2[...]]]]]
```

The fields are comma delimited.

- Field 1: StartFolder. The StartFolder of the Documanage Query, i.e., the zero-based index of the Level 1 Folder at which to start searching.
- Field 2: StartingDocIndex. The zero-based index of the Document to start at within the starting folder designated in the following fields.
- Field 3: StartingFldIndexL1. The zero-based starting Folder index for Level 1 relative to StartFolder. This is essentially a placeholder and is always 0 since StartFolder indicates the starting Level 1 Folder.
- Fields 4 and up: StartingFldIndexLx. The zero-based starting Folder index within the Folder at the previous Level, where x is the Level. There can be 0 or more of these, up to Level 10. The highest one indicates the Level to start at.

For example, STARTAT = 0,20,0 means “start with the 21st Document (index 20) of the first Folder found at Level 1”. STARTAT = 7,0,0,4,2 means “start the Query at top Level Folder 7, and at the first Document (index 0) in the third Level 3 Folder in the fifth Level 2 Folder in the first Level 1 Folder”.

QUERY_STEM

The QUERY_STEM that DmgBrsQueryDocuments returns is for convenience in building URLs to call DmgBrsQueryDocuments again to continue navigating through results with the same settings and filters. The QUERY_STEM contains all non-empty inputs listed below in the format of an http query. Any filters built with SQL Builder are included in the QUERY_STEM. BRIDGEURL plus this stem and REQTYPE is a valid URL that points to the start of a search with DmgBrsQueryDocuments. BRIDGEURL plus this stem and REQTYPE and any valid STARTAT value begins returning results from DmgBrsQueryDocuments at the spot indicated by the STARTAT. All text parts of QUERY_STEM are URL encoded.

The inputs included in QUERY_STEM for DmgBrsQueryDocuments are:

```
CABINET
CAB_FILTER_variablename variables
FLD_FILTER
DOC_FILTER
XDOC_FILTER
FLD_ORDERBY
DOC_ORDERBY
FULLTEXT_FILTER
DOC_CATEGORY
PS_LIST
MAXHITS
MAXFLDCOUNT
LEVEL_LIMITER
LEVEL_MASK
GROUP_HITS_BY_LEVEL
```

For example, the value of a QUERY_STEM when the PS_LIST and MAXHITS variables were passed in might be as follows:

```
CABINET=My%20Cabinet&PS_LIST=QueryDocsPSList&MAXHITS=10
```

Examples

In addition to the examples below, see Example 6 for DmgBrsQuery for various ways that the results of DmgBrsQueryDocuments can be used to link to other rules. Although the examples are for DmgBrsQuery, the principles and even the names of the variables are the same.

- 1 In this example, we'll search a Cabinet with 3 Levels, and with GroupHitsByLevel turned off, so that all of the hits are returned in one list. Assume that [QueryDocuments] MaxHits is 40. The input attachment variables might look like the following.

```
CABINET = OEM Parts Explosion
PS_LIST = OEM_PARTS_EXPLOSION
GROUP_HITS_BY_LEVEL = 0
```

The output attachment variables might look like the following.

```
DOC_FILTER =
FOLDER_FILTER =
HITS = 40
HITS1.CONVNAME = /BLUEBALL.pdf
HITS1.DOC_CATEGORY = Drawing
HITS1.DOC_DATE =
HITS1.DOC_ID = 829
HITS1.DOC_MAJORVERSION = 1
HITS1.DOC_MINORVERSION = 0
HITS1.DOC_NAME = BLUEBALL
HITS1.DOCSPEC =
CABINET=OEM%20Parts%20Explosion&DOC_ID=829&DOC_VERS=1.0
HITS1.EVEN = 0
HITS1.FLD_ID = 142
HITS1.FLD_KEYSTRING = TMSDocs.dbo.Machines.MachineID = 'M3 '
HITS1.FLD_LABEL = M3
HITS1.FLD_LEVELNUMBER = 1
HITS1.ORIGNAME = /BLUEBALL.GIF
HITS1.XFLD_L1_Machineid = M3
HITS1.XFLD_L1_Machinename = Chassis builder
HITS1.XFLD_L2_Assemblyid =
HITS1.XFLD_L2_Assemblyname =
HITS1.XFLD_L2_Machineid =
HITS1.XFLD_L3_Assemblyid =
HITS1.XFLD_L3_Partid =
HITS1.XFLD_L3_Partname =
... [plus records for HITS_2 through HITS_40]
HITS_EXIST = 1
HITS_EXIST_L1 = 1
HITS_EXIST_L2 = 1
HITS_EXIST_L3 = 1
MOREHITS = 1
QUERY_STEM=CABINET=OEM%20Parts%20Explosion&PS_LIST=OEM_PARTS_EXPLOSION&GROUP_HITS_BY_LEVEL=0
STARTAT_NEXT = 1,2,0,0,1
XDOC_FILTER =
```

We can infer from the above variables that the Property Set List OEM_PARTS_EXPLOSION contains PS_XFLD_L1, PS_XFLD_L2, and PS_XFLD_L3 Property Sets because of the presence of the variables whose names begin with HITS1.XFLD_L. Since the Document shown above is in a Folder at Level 1, as can be seen from the variable HITS1.FLD_LEVELNUMBER = 1, the variables whose names begin with HITS1.XFLD_L2 and HITS1.XFLD_L3 are empty. Notice that the variables HITS_EXIST_L2 and HITS_EXIST_L3 are always present even though the hits are not grouped by level. In this case, they indicate that hits were found at Levels 2 and 3.

- 2 In this example, we'll repeat the search from Example 1, but with one important difference. We'll group hits by level this time. The input attachment variables would look like the following.

```
CABINET = OEM Parts Explosion
PS_LIST = OEM_PARTS_EXPLOSION
GROUP_HITS_BY_LEVEL = 1
```

The output attachment variables might look like the following.

```
DOC_FILTER =
FOLDER_FILTER =
HITS = 40
HITS_EXIST = 1
HITS_EXIST_L1 = 1
HITS_EXIST_L2 = 1
HITS_EXIST_L3 = 1
HITS_L1_ = 10
HITS_L1_1.CONVNAME = /BLUEBALL.pdf
HITS_L1_1.DOC_CATEGORY = Drawing
HITS_L1_1.DOC_DATE =
HITS_L1_1.DOC_ID = 829
HITS_L1_1.DOC_MAJORVERSION = 1
HITS_L1_1.DOC_MINORVERSION = 0
HITS_L1_1.DOC_NAME = BLUEBALL
HITS_L1_1.DOCSPEC =
CABINET=OEM%20Parts%20Explosion&DOC_ID=829&DOC_VERS=1.0
HITS_L1_1.EVEN = 0
HITS_L1_1.FLD_ID = 142
HITS_L1_1.FLD_KEYSTRING = TMSDocs.dbo.Machines.MachineID = 'M3'
HITS_L1_1.FLD_LABEL = M3
HITS_L1_1.ORIGNAME = /BLUEBALL.GIF
HITS_L1_1.XFLD_L1_Machineid = M3
HITS_L1_1.XFLD_L1_Machinename = Chassis builder
... [plus records for HITS_L1_2 through HITS_L1_10]
HITS_L2_ = 6
HITS_L2_1.CONVNAME = /overhead%20conveyer.pdf
HITS_L2_1.DOC_CATEGORY = Drawing
HITS_L2_1.DOC_DATE =
HITS_L2_1.DOC_ID = 741
HITS_L2_1.DOC_MAJORVERSION = 1
HITS_L2_1.DOC_MINORVERSION = 0
HITS_L2_1.DOC_NAME = overhead conveyer
HITS_L2_1.DOCSPEC =
CABINET=OEM%20Parts%20Explosion&DOC_ID=741&DOC_VERS=1.0
HITS_L2_1.EVEN = 0
HITS_L2_1.FLD_ID = 181
HITS_L2_1.FLD_KEYSTRING = TMSDocs.dbo.Assemblies.AssemblyID = 'A7'
HITS_L2_1.FLD_LABEL = A7
HITS_L2_1.ORIGNAME = /overhead%20conveyer.GIF
HITS_L2_1.XFLD_L1_Machineid = M3
HITS_L2_1.XFLD_L1_Machinename = Chassis builder
HITS_L2_1.XFLD_L2_Assemblyid = A7
HITS_L2_1.XFLD_L2_Assemblyname = overhead conveyer
HITS_L2_1.XFLD_L2_Machineid = M3
HITS_L2_2.CONVNAME = /PM%20PROCEDURE.pdf
HITS_L2_2.DOC_CATEGORY = Instructions
```

```
HITS_L2_2.DOC_DATE =
HITS_L2_2.DOC_ID = 758
HITS_L2_2.DOC_MAJORVERSION = 1
HITS_L2_2.DOC_MINORVERSION = 0
HITS_L2_2.DOC_NAME = PM PROCEDURE
HITS_L2_2.DOCSPEC =
CABINET=OEM%20Parts%20Explosion&DOC_ID=758&DOC_VERS=1.0
HITS_L2_2.EVEN = 1
HITS_L2_2.FLD_ID = 181
HITS_L2_2.FLD_KEYSTRING = TMSDocs.dbo.Assemblies.AssemblyID = 'A7 '
HITS_L2_2.FLD_LABEL = A7
HITS_L2_2.ORIGNAME = /PM%20PROCEDURE.DOC
HITS_L2_2.XFLD_L1_Machineid = M3
HITS_L2_2.XFLD_L1_Machinename = Chassis builder
HITS_L2_2.XFLD_L2_Assemblyid = A7
HITS_L2_2.XFLD_L2_Assemblyname = overhead conveyer
HITS_L2_2.XFLD_L2_Machineid = M3
... [plus records for HITS_L2_3 through HITS_L2_6]
HITS_L3_ = 24
HITS_L3_1.CONVNAME = /test.pdf
HITS_L3_1.DOC_CATEGORY = test
HITS_L3_1.DOC_DATE = 2003-06-03 13:20:44
HITS_L3_1.DOC_ID = 1219
HITS_L3_1.DOC_MAJORVERSION = 1
HITS_L3_1.DOC_MINORVERSION = 0
HITS_L3_1.DOC_NAME = test
HITS_L3_1.DOCSPEC =
CABINET=OEM%20Parts%20Explosion&DOC_ID=1219&DOC_VERS=1.0
HITS_L3_1.EVEN = 0
HITS_L3_1.FLD_ID = 265
HITS_L3_1.FLD_KEYSTRING = TMSDocs.dbo.Parts.partID = 'P31 ' and
TMSDocs.dbo.Parts.SubAssemblyID = 'A8 '
HITS_L3_1.FLD_LABEL = P31 in A8
HITS_L3_1.ORIGNAME = /test.TXT
HITS_L3_1.XFLD_L1_Machineid = M3
HITS_L3_1.XFLD_L1_Machinename = Chassis builder
HITS_L3_1.XFLD_L2_Assemblyid = A8
HITS_L3_1.XFLD_L2_Assemblyname = AddedAssembly
HITS_L3_1.XFLD_L2_Machineid = M3
HITS_L3_1.XFLD_L3_Assemblyid = A8
HITS_L3_1.XFLD_L3_Partid = P31
HITS_L3_1.XFLD_L3_Partname = AddedPart
... [plus records for HITS_L3_2 through HITS_L3_24]
MOREHITS = 1
QUERY_STEM=CABINET=OEM%20Parts%20Explosion
&PS_LIST=OEM_PARTS_EXPLOSION&GROUP_HITS_BY_LEVEL=0
STARTAT_NEXT = 1,2,0,0,1
XDOC_FILTER =
```

Note that, for the Document in record HITS_L1_1, the field variables whose names begin with XFLD_L2 and XFLD_L3 are absent. Compare with records HITS_L2_1 and HITS_L3_1. Also note that the field variable FLD_LEVELNUMBER isn't returned since the Level is obvious from the name of the variables.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options	[QueryDocuments]
Validations	Session and Reqtype
Platform restrictions	None.
See also	DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsLogPropertySets DmgBrsPresentDocumentPage DmgBrsPresentProject DmgBrsQueryDiaries DmgBrsQuery DmgBrsReturnConvertedDocument DmgBrsReturnDocument

DMGBRSQUERYPROJECTS

DmgBrsQueryProjects searches for a user's Projects in a Workflow Cabinet, without regard to Task.

Usage

```
function = dmg_brw32->DmgBrsQueryProjects
```

Attachment inputs

Variable	Description
CABINET	The name of the Workflow Cabinet to be searched. This must be a Workflow Cabinet, or the rule returns an error. Required.
PRJ_FILTER	An Extended Project Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
PRJ_ORDERBY	A Project order by SQL clause for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional.
QUERY_PRJ_...	SQL Builder variables describing a Standard Extended Project Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to PRJ_FILTER, and will be ignored if PRJ_FILTER is passed in. Optional. This filter can be used in combination with a QUERY_XPRJ_... filter.
QUERY_XPRJ_...	SQL Builder variables describing an Extended Project Property filter. See the description of SQL Builder syntax in the Bridge General reference. This is an alternative to PRJ_FILTER, and will be ignored if PRJ_FILTER is passed in. Optional. This filter can be used in combination with a QUERY_PRJ_... filter.
OBn_PRJ_...	SQL Builder variables describing a Project Orderby filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to PRJ_ORDERBY, and will be ignored if PRJ_ORDERBY is passed in. Optional.
MAXHITS	The number of Project hits allowed to be returned in one transaction. Limited by the INI option [QueryProjects] MaxHits. Optional. Default is the INI option [QueryProjects] MaxHits.
PROJECTS_INDEX	The index to start displaying Projects. Zero-based. Optional. Default is 0.
CHECKEDOUT_STATE	A filter that will be ANDed with any other filters. Optional. The default is the INI option [QueryProjects] CheckedoutState. This has three recognized values, case independent: ANY—return Projects without regard to their checkedout state. CHECKED_OUT—return only those Projects that are checked out. NOT_CHECKED_OUT—return only those Projects that are not checked out.

PROJECT_STATE	A filter that will be ANDed with any other filters. Optional. The default is the INI option [QueryProjects] ProjectState. This has three recognized values, case independent: ANY -- return Projects without regard to their Project state. PENDING—return only those Projects that are Pending. SUSPENDED—return only those Projects that are Suspended.
PS_LIST	The name of a Property Set List to limit what Extended Project Properties are returned with each Project hit. Optional. If absent, DmgBrsQueryProjects will use a Property Set List called "QueryProjectsDefault" if one exists.
SHOW_DOCS_IND	If this is 1, then each Project record will include DOCS_IND to indicate whether the Project contains Documents. This has a performance cost. 0 means no DOCS_IND. Optional. Default is the INI option [DocsIndicator] ShowDocsIndicator.
SHOW_NOTES_IND	If this is 1, then each Project record will include NOTES_IND to indicate whether the Project contains Notes. 0 means no NOTES_IND. Optional. Default is the INI option [Notes] ShowNotesIndicator.

Attachment outputs

Variable	Description
PROJECTS_EXIST	Contains 1 if any Projects were found, else contains 0. Note that this does not indicate whether Projects exist for the user. It merely indicates whether any were found given any filters and the PROJECTS_INDEX that were passed in. Projects may exist for the user even if none were found after applying the filters.
PROJECTS	The number of Projects found. Exists only if PROJECTS_EXIST contains 1.
PROJECTSx.field	A record set describing the Projects that were found, where x is the record number and field is one of the following fields. Exists only if PROJECTS is greater than 0.
.XPRJ_propertyname	A set of variables containing the values of the Extended Project Properties requested in the PS_XPRJ Property Set, if any, on the Property Set list in PS_LIST where propertyname is the Bridge-encoded Project Property Name (not DB Field Name), or all these Properties if no Property Set List was used.
.DOCS_IND	Contains 1 if there are Documents in this Project, else contains 0. Exists only if SHOW_DOCS_IND contains 1.
.NOTES_IND	Contains 1 if there are Notes associated with this Project, else contains 0. Exists only if SHOW_NOTES_IND contains 1.
.PRJ_LABEL	The Label Property of this Project.

.PRJ_ID	The ID Property of this Project.
.PRJ_KEYSTRING	The Keysting Property of this Project.
.PRJ_TASKTYPE	The numeric TaskType Property of this Project.
.PRJ_TASKTYPE_TEXT	The TaskType Property of this Project, in text form, where the text matches the enum DmgTaskTypes in dmgs_api_types.h.
.PRJ_standardextendedproperty	A set of variables containing the values of the 19 Standard Extended Project Properties described in the Introduction.
MOREPROJECTS	Contains 1 if there are Projects after the ones that have been returned in this transaction, else contains 0.
PROJECTS_INDEX_NEXT	The index of the next batch of Projects. Exists only if MOREPROJECTS contains 1.
PREVPROJECTS	Contains 1 if there are Projects previous to the ones that have been returned in this transaction, else contains 0.
PROJECTS_INDEX_PREV	The index of the previous batch of Projects. Exists only if PREVPROJECTS contains 1.
QUERY_STEM	A variable that provides convenience in calling DmgBrsQueryProjects again with the same settings and filters. See the explanation under Remarks.
REACHED_FLD_FETCH_LIMIT	Exists only if MOREPROJECTS contains 0. Contains 1 if MOREPROJECTS is 0 because the Folder Fetch Limit has been reached, else contains 0. The Folder Fetch Limit is a system option that may be set with the Documanager Administrator.

Remarks DmgBrsQueryProjects searches for a user's Projects in a Workflow Cabinet, without regard to Task.

This rule is similar to DmgBrsQuery. It provides paging through results in a similar manner. Bear in mind that the Project list is likely to be more volatile than other types of lists. For this reason, paging by index might cause items to be skipped or repeated more often than it would for paging through other types of items.

Note that the only required input attachment variable is CABINET. You may pass in filters, and you may request automatic filtering by Suspended or Pending, and/or checked out or not checked out Projects.

DmgBrsQueryProjects doesn't return lists of Documents in the Projects. To see the Documents in a given Project, use DmgBrsPresentProject.

The QUERY_STEM for DmgBrsQueryProjects works like that for DmgBrsQueryDocuments. See the description of this variable for DmgBrsQueryDocuments. The inputs included in QUERY_STEM for DmgBrsQueryProjects are:

CABINET

```

PRJ_FILTER
PRJ_ORDERBY
MAXHITS
CHECKEDOUT_STATE
PROJECT_STATE
PS_LIST
SHOW_DOCS_IND
SHOW_NOTES_IND

```

Examples Using the following attachment inputs would cause up to the first 10 Projects to be returned that were checked out to the user and were pending. Assume that the INI option [QueryProjects] MaxHits has a value of 20.

```

CABINET = MyWorkflowCabinet
MAXHITS = 10
CHECKEDOUT_STATE = Checked_Out
PROJECT_STATE = Pending

```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [QueryProjects]
 [DocsIndicator] ShowDocsIndicator
 [Notes] ShowNotesIndicator

Validations Session and Reqtype

Platform restrictions None.

See also DmgBrsQueryAllProjectsEx
 DmgBrsGetTaskQueue
 DmgBrsPresentProject
 DmgBrsQuery
 DmgBrsQueryDocuments

DMGBRSREMOVEDOCUMENTASRENDITION

DmgBrsRemoveDocumentAsRendition removes a Document Version from all the Rendition relationships of which it is a member.

Usage

```
function = dmg_brw32->DmgBrsRemoveDocumentAsRendition
```

Attachment inputs

Variable	Description
A Version Document Specification	The Document Version to delete from the set of Renditions. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Cabinet is a Workflow Cabinet, else 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsRemoveDocumentAsRendition removes a Document Version from all the Rendition relationships of which it is a member.

Note that this doesn't delete the Document Version itself. After this rule has executed, the Document Version will still exist, but it will no longer be listed as a Rendition of another Document Version.

This rule will fail if the Bridge is not connected to Documange Service Release 2 or higher.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

DmgBrsPresentDocument
DmgBrsPresentFolder
DmgBrsPresentProject

See also

DmgBrsAddDocumentAsRendition
DmgBrsQueryDocumentRenditions

DmgBrsDeleteDocument

DMGBRSRESUMEPROJECT

DmgBrsResumeProject resumes a suspended Project.

Usage `function = dmg_brw32->DmgBrsResumeProject`

Attachment inputs

Variable	Description
A Project Specification	The Project to resume. Required.

Attachment outputs None.

Remarks DmgBrsResumeProject resumes a suspended Project. To suspend a project, use DmgBrsSuspendProject.

Examples See the example for DmgBrsGetTaskQueue.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsPresentProject

See also DmgBrsGetTaskQueue
DmgBrsPresentProject
DmgBrsSuspendProject

DMGBRSRETURNANNOTATIONS

DmgBrsReturnAnnotations returns an annotation file containing all the annotations for a Document.

Usage

```
function = dmg_brw32->DmgBrsReturnAnnotations
```

Attachment inputs

Variable	Description
Annotations File Info	The outputs of DmgBrsCacheAnnotationsFile for the Document whose Annotations are to be returned. Required.

Attachment outputs

Variable	Description
HTTP_STATUS	The status for creating an HTTP header. If the rule fails, this is set to "404 Not Found". Otherwise, the default status set by DmgBrsCopyAttachment is left unchanged.
HTTP_MIMETYPE	The MIME type for creating an HTTP header. "application/octet-stream"
HTTP_CONTENTLENGTH	The length of the file in bytes.
TARGET_FILE	The path of the file.
TARGET_FILE_NAME	The name (including extension) from the end of TARGET_FILE.
TEMP_FILE	The path of a temp file for direct download by a web server. Exists only if the INI option [TempCache] TempFiles is ON. The file will be in the directory configured in the INI option [TempCache] DirPath.
TEMP_FILE_NAME	The name (including extension) from the end of TEMP_FILE.

Remarks

DmgBrsReturnAnnotations returns an annotation file containing the annotations for a Document. The rule DmgBrsCacheAnnotationsFile must precede this rule on the rules list. Use DmgBrsReturnDocument to return the Document contents file.

If an annotation file is being downloaded for the purpose of being updated, be sure to download the current version of the file. To do this, set the Major Version in the Version Document Specification passed to DmgBrsCacheAnnotationsFile to -1.

This rule requires Documange Server 6.4 service release 7 or later.

Examples

See the example for DmgBrsLockAnnotations.

ReqType changes

[ReqTypes] Failure if the rule fails.

[ReqTypes] Redirection if TEMP_FILE exists.

INI options

[TempCache]

Validations

Session and Reqtype

Platform restrictions	None.
See also	DmgBrcDumpFile DmgBrsLockAnnotations DmgBrsReturnConvertedDocument DmgBrsReturnDocument DmgBrsUnlockAnnotations DmgBrsUpdateAnnotations DmgBrsCacheAnnotationsFile

DMGBRSRETURNCONVERTEDDOCUMENT

DmgBrsReturnConvertedDocument returns a Document contents file converted to the specified type.

UsageUsage

```
function = dmg_brw32->DmgBrsReturnConvertedDocument
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document to convert. Required.
SUBDOC_NUM	The one-based Subdocument number for the Subdocument to convert. Optional if the Document is Compound, else ignored. For a Subdocument to be converted, the Document must be Compound and SUBDOC_NUM must exist. If the Document is Compound and SUBDOC_NUM does not exist, this rule will attempt to convert the entire Compound Document to one PDF file. The rule will fail if this is not possible.
CONVNAME	A URL Tail for the converted Document. The extension on the value of this variable indicates the requested converted file type. If there is no extension, the INI option [Rendering] DocumentConversion is used to determine the converted FileType. Required. If this variable is missing, the rule does not fail. It simply does nothing.
CACHE_FILE_SUFFIX	This variable may be empty or absent, else it must contain one value from the suffix list configured in [ReturnConvertedDocument]CacheFileSuffixes. If it contains a value that does not match any of the configured values, the rule DmgBrsReturnConvertedDocument will fail. If it contains a valid value, then the value will be appended to the file name of the converted file in the Bridge cache. This allows the same Documanager document converted multiple times to the same type, but with different characteristics, to be distinguished in the Bridge cache. For example, if you convert a document to PDF with a watermark, and then convert the same document to PDF without a watermark, you would use CACHE_FILE_SUFFIX to distinguish the two PDFs in the Bridge cache.

If CONTENTS_DECOMPRESSED_TYPE is MET or AFP (i.e., Printstream):

Variable	Description
PRINTSTREAM_TO_IMAGE_PDF	Set this to 1 to cause this rule to convert PrintStream contents files to Image PDF when the desired converted file type is PDF. Set it to 0 to cause this rule to convert PrintStream contents files to text PDF when the desired converted file type is PDF. Optional. Default is the INI option [ReturnConvertedDocument] PrintstreamToImagePDF. Use the INI option [ReturnConvertedDocument] PrintstreamToImagePDFColor to control whether Printstreams can be converted to color PDF. See the caution below.

PRINTSTREAM_CONFIG	The value for CONFIG, which contains the Printstream config in DAP.INI to use for the conversion. Optional if the contents file is a Printstream file and PRINTSTREAM_TO_IMAGE_PDF is 0, else ignored. The default is the INI option [ReturnConvertedDocument] PrintstreamConfig. If the INI option is empty, the value of the attachment variable TEMP_CONFIG will be used. If TEMP_CONFIG is not found or empty, the value of the attachment variable CONFIG will be used. If CONFIG is not found or empty, the value of the attachment variable SESS_CONFIG will be used. See DmgBrsValidateSession.
--------------------	---

If CONTENTS_DECOMPRESSED_TYPE is DPA:

Variable	Description
DMGBR_FORMFILTEREXISTS	DmgBrsReturnConvertedDocument will make appropriate adjustments to the name of the converted version of a DPA file in the Bridge File Cache if it might be made with different Formsets on different requests. It ordinarily detects the possibility of different Formsets by the presence of the inputs for the rule DPRFilterFormsetForms on the input attachment. Use DMGBR_FORMFILTEREXISTS to over-ride this detection. If DMGBR_FORMFILTEREXISTS exists on the input attachment and contains 1, then DmgBrsReturnConvertedDocument will behave as if filtered Formsets are in use. For example, this could be useful if a custom rule for Formset filtering is in use and the inputs for DPRFilterFormsetForms are absent. If DMGBR_FORMFILTEREXISTS exists on the input attachment and contains 0, then DmgBrsReturnConvertedDocument will behave as if filtered Formsets are not in use. For example, this could be useful if the same Formset filter is always used. Optional.

If the converted Document will be an Image PDF:

Variable	Description
----------	-------------

PAGE_RANGE	<p>The range of pages to convert. The value of this variable is one or more pages or page ranges separated by commas. Use a hyphen to indicate a page range. If the first page of a range is not specified, the rule uses the first page of the document for the first page of the range. If the last page of the range is not specified, the rule uses the last page of the document for the last page of the range.</p> <p>The following example values assume a 10 page document.</p> <p>Example 1: "1,3,8, 9".</p> <p>Example 2: "5-7,2".</p> <p>Example 3: "4, 3, 7-".</p> <p>Example 4: "-5, 9, 4".</p> <p>Extra white space, extra commas, inverted page ranges, and characters that are not digits, hyphens, or commas are all ignored. The pages are always converted in the order they appear in the Document, and not in the order they appear in PAGE_RANGE. Duplicate pages are allowed in PAGE_RANGE, but the page will appear only once in the output. Optional. Default is empty, which means that all pages will be converted.</p>
------------	---

Attachment outputs

Variable	Description
HTTP_STATUS	The status for creating an HTTP header. If the rule fails, this is set to "404 Not Found". Otherwise, the default status set by DmgBrsCopyAttachment is left unchanged.
HTTP_MIMETYPE	The MIME type for creating an HTTP header. The MIME type will be based on the extension of the converted file. If no MIME type for the extension of the converted file is listed in DMG_BRMIME.INI or in the Bridge's internal list of MIME types, a default MIME type of "application/octet-stream" is used. See the description of the Bridge configuration file DMG_BRMIME.INI.
HTTP_CONTENTLENGTH	The length of the converted file in bytes.
TARGET_FILE	The path of the converted file.
TARGET_FILE_NAME	The name (including extension) from the end of TARGET_FILE.
TEMP_FILE	The path of a temp file for direct download by a web server. Exists only if the INI option [TempCache] TempFiles is ON. The file will be in the directory configured in the INI option [TempCache] DirPath.
TEMP_FILE_NAME	The name (including extension) from the end of TEMP_FILE.

If CONTENTS_DECOMPRESSED_TYPE is MET or AFP, and if the conversion is not to Image PDF:

Variable	Description
METACODEFILE	A required input for MTCLoadFormset, output when the Document contents file is of type MET. The path to the contents file to be converted. The value will be the same as that of CONTENTS_DECOMPRESSED_PATH. This variable is written to the Input Queue.
AFPPFILE	A required input for MTCLoadFormset, output when the Document contents file is of type AFP. The path to the contents file to be converted. The value will be the same as that of CONTENTS_DECOMPRESSED_PATH. This variable is written to the Input Queue.
PRINTFILE	A required input for DPRPrint. The path for the converted file. The value will be the same as that of TARGET_FILE. This variable is written to the Input Queue.
CONFIG	See the description of the input variable PRINTSTREAM_CONFIG. This variable is written to the Input Queue.
USERID	A required input for MTCLoadFormset. Contains “DMG_BR”. This variable is written to the Input Queue unless a variable of that name already exists on the Input Queue, in which case the existing variable is left unchanged.

If CONTENTS_DECOMPRESSED_TYPE is DCD:

Variable	Description
CPYINFILE	A required input for ExecISICpy. The path to the contents file to be converted. The value will be the same as that of CONTENTS_DECOMPRESSED_PATH. This variable is written to the Input Queue.
CPYOUTFILE	A required input for ExecISICpy. The path for the converted file. The value will be the same as that of TARGET_FILE. This variable is written to the Input Queue.
CPYCONFIGFILE	A required input for ExecISICpy. The config file to be used for the conversion, which determines the type of the converted file. The value used is determined by the INI option [ReturnConvertedDocument] CpyConfigFile_x, if an option exists where x is the converted file type specified by CONVNAME, else by [ReturnConvertedDocument] CpyConfigFile. This variable is written to the Input Queue unless a variable of that name already exists on the Input Queue, in which case the existing variable is left unchanged.

If CONTENTS_DECOMPRESSED_TYPE is DPA:

Variable	Description
----------	-------------

DMSARCFIELD	A required input for DPRRetrieveDPA. The path to the contents file to be converted. The value will be the same as that of CONTENTS_DECOMPRESSED_PATH. This variable is written to the Input Queue.
PRINTFILE	A required input for DPRPrint. The path for the converted file. The value will be the same as that of TARGET_FILE. This variable is written to the Input Queue.
PRTTYPE	A required input for DPRRetrieveDPA. The type of the converted file specified by CONVNAME. This variable is written to the Input Queue.
USERID	A required input for DPRRetrieveDPA. Contains "DMG_BR". This variable is written to the Input Queue unless a variable of that name already exists on the Input Queue, in which case the existing variable is left unchanged.

If the type of the converted file specified by CONVNAME is PDF, if the Server INI option [ReturnConvertedDocument]UseTPDForImagePDF is ON, and if CONTENTS_DECOMPRESSED_TYPE is a type not otherwise covered (Windows only):

Variable	Description
PDFNAME	A required input for TPDCreateFormset. The path for the converted file. The value will be the same as that of TARGET_FILE. This variable is written to the Input Queue.
SRCTYPE	A required input for TPDCreateFormset. This will contain "JPG" if the Document contents file contains any pages in color, else it will contain "TIF". This variable is written to the Input Queue.
TIFFNAME	A required input for TPDCreateFormset. The number of page files for the PDF. This variable is written to the Input Queue.
TIFFNAMEx.NAME	A required input for TPDCreateFormset. A record set listing the paths to each of the page files for the PDF, where x is the record number. The paths will be in the Bridge File Cache. The files will be of type JPG if the Document contents file contains any pages in color, else they will be of type TIF. These variables are written to the Input Queue.

Remarks

DmgBrsReturnConvertedDocument returns a Document contents file converted to the specified type. The rule DmgBrsCacheContentsFile must precede this rule on the rules list.

DmgBrsReturnConvertedDocument determines the type of the converted file based on the input attachment variable CONVNAME, as described above. Some conversions are not possible, and some conversions available on Windows are not available on other platforms. See the table below for more information.

DmgBrsReturnConvertedDocument doesn't do the actual conversion, but instead relies upon other rules and libraries to carry out the conversion. The particular rules and libraries it uses depend on the type of the Document contents file, and on the type of the converted file. On its reverse run, DmgBrsReturnConvertedDocument checks for the requested converted file and fails if it does not exist.

If you are doing conversions that use ExecISICpy, DPRRetrieveDPA, or MTCLoadFormset, you must customize all the necessary INI files and font cross-reference files they require. See

the documentation on font cross-reference files, on the DocuMerge Bridge, and on those rules for more information.

When converting Printstream contents files (files of type MET or AFP) to PDF, which type of PDF will be produced—image PDF or text PDF—is controlled by the variable PRINTSTREAM_TO_IMAGE_PDF and its associated INI options. A text PDF represents the file's contents with characters, and may be searched as text. However, it is not guaranteed to look exactly like the source document for Printstream files. An image PDF contains a bit mapped image of the file's contents, and cannot be searched as text. For PrintStream files, the bit mapped image provides a high fidelity representation of the document, but it will generally produce larger files and will take longer to convert than text PDF.

When converting a contents file to image PDF on Windows, DmgBrsReturnConvertedDocument uses the Shuttle library to create a file of type JPG for each page of a color contents file to be converted to image PDF, or a file of type TIF for each page of a black and white contents file to be converted. The Shuttle library in turn uses the Printstream library to derive the individual pages from the Printstream file. You must properly archive the required fonts and other resources for your Printstream documents into Documanage with Docusave Server. For information on configuring the Printstream library, please see the Using Docusave Workstation LFS manual. These libraries analyze each Printstream file and determine the resources needed to render it. Those resources are then retrieved from Documanage and used to draw each page of the requested document as an image of the page. If DmgBrsReturnConvertedDocument cannot create a particular page file for a black and white contents file, it will substitute the file configured by the INI option [ReturnConvertedDocument] MissingBlackAndWhitePageFile for that file. If it cannot create a particular color page, it will substitute the file configured by the INI option [ReturnConvertedDocument] MissingColorPageFile for it. If the INI option [ReturnConvertedDocument]UseTPDForImagePDF is ON on Windows, TPDCreateFormset assembles the TIF or JPG page images into a Formset and produces a PDF.

CAUTION:On Windows, converting a large color document to image PDF in its entirety with TPDCreateFormset uses a lot of system resources, including CPU time and memory. If converting such a document exceeds system capacity, you may need to add more memory, set the INI option [ReturnConvertedDocument]PrintstreamToImagePDFColor to OFF (which causes the document to be converted in black and white), or limit the number of pages to convert using the PAGE_RANGE input attachment variable. Alternatively, if your application does not require customizations available with the TPD rules, you can set the INI option [ReturnConvertedDocument]UseTPDForImagePDF to OFF. This will cause this rule to use its built-in facility for writing image PDFs instead of setting up the TPD rules. That facility does not have the limitations of the TPD rules.

When converting a contents file to image PDF on Unix platforms, only files of type TIF or JPG can be converted.

If the type of the Document contents file is the same as the requested converted file type, then no conversion is necessary and DmgBrsReturnConvertedDocument behaves like DmgBrsReturnDocument. Otherwise, DmgBrsReturnConvertedDocument attempts to convert the contents file to the requested type depending on the type of the contents file, as described in the following table.

Contents File Type	Converted File Type	Description
--------------------	---------------------	-------------

DCD	types convertible with ExecISICpy	DmgBrsReturnConvertedDocument provides the necessary inputs for the rule ExecISICpy to carry out the conversion. If ExecISICpy cannot handle the requested type, then conversion will fail.
DPA	types convertible with DPRRetrieveDPA / DPRPrint	DmgBrsReturnConvertedDocument provides the necessary inputs for the rules DPRRetrieveDPA and DPRPrint to carry out the conversion. If DPRRetrieveDPA and DPRPrint cannot handle the requested type, then conversion will fail.
MET or AFP	types convertible with MTCLoadFormset / DPRPrint	DmgBrsReturnConvertedDocument provides the necessary inputs for the rules MTCLoadFormset and DPRPrint to carry out the conversion. If MTCLoadFormset and DPRPrint cannot handle the requested type, then conversion will fail. If the requested converted type is PDF and PRINTSTREAM_TO_IMAGE_PDF contains 1, then MTCLoadFormset, which creates a text PDF, is not used for the conversion. Instead, DmgBrsReturnConvertedDocument provides the necessary inputs for an image PDF conversion with TPDCreateFormset.
FPP	XML	DmgBrsReturnConvertedDocument uses the DMFPXML.DLL library to carry out the conversion. Windows only. If conversion is requested for an FPP file on a non-Windows platform, or if a converted file type other than XML is requested on Windows, DmgBrsReturnConvertedDocument will fail.
Other types not covered above, and MET or AFP when PRINTSTREAM_TO_IMAGE_PDF contains 1.	PDF	On Windows: If the INI option[ReturnConvertedDocument]UseTPDForImagePDF is ON, DmgBrsReturnConvertedDocument uses the Shuttle and Printstream libraries to provide the necessary inputs for the rule TPDCreateFormset to carry out the conversion. If the INI option is OFF, DmgBrsReturnConvertedDocument uses Shuttle and the rule's built-in facility for writing image PDF to carry out the conversion. If this type of conversion is requested on a non-Windows platform, DmgBrsReturnConvertedDocument will fail unless the file or files being converted are of type JPG or TIF.
Other types not covered above	Types other than PDF	DmgBrsReturnConvertedDocument will fail.

- Examples** 1 A rules list to do conversions of all types could be set up as follows.

```
[ ReqType:CONVERTDOC ]  
function = atcw32->ATCLogTransaction  
function = atcw32->ATCLoadAttachment  
function = atcw32->ATCUnloadAttachment  
function = dmg_brw32->DmgBrsCopyAttachment  
function = dmg_brw32->DmgBrsValidateSession  
function = dmg_brw32->DmgBrsCacheContentsFile  
function = dmg_brw32->DmgBrsReturnConvertedDocument  
function = cpyrule->ExecISICpy  
function = dprw32->DPRRetrieveDPA  
function = Tpdw32->TPDCreateFormset  
function = dprw32->DPRSetConfig  
function = mtcw32->MTCLoadFormset  
function = dprw32->DPRPrint
```

You can, of course, remove any conversion rules that you don't need. For example, if your Bridge installation deals only with contents files of type DCD, then you need the rule ExecISICpy, but you can remove the rules DPRRetrieveDPA, TPDCreateFormset, DPRSetConfig, MTCLoadFormset, and DPRPrint. TPDCreateFormset is not needed if the INI option [ReturnConvertedDocument]UseTPDForImagePDF is OFF.

- 2 Some rules lists to do particular types of conversions could be set up as follows. In general, separating rules lists by their purpose is clearer than having rules lists that try to do too many different things.

```
[ ReqType:CONVERTFPP ]  
function = atcw32->ATCLogTransaction  
function = atcw32->ATCLoadAttachment  
function = atcw32->ATCUnloadAttachment  
function = dmg_brw32->DmgBrsCopyAttachment  
function = dmg_brw32->DmgBrsValidateSession  
function = dmg_brw32->DmgBrsCacheContentsFile  
function = dmg_brw32->DmgBrsReturnConvertedDocument  
  
[ ReqType:CONVERTDCD ]  
function = atcw32->ATCLogTransaction  
function = atcw32->ATCLoadAttachment  
function = atcw32->ATCUnloadAttachment  
function = dmg_brw32->DmgBrsCopyAttachment  
function = dmg_brw32->DmgBrsValidateSession  
function = dmg_brw32->DmgBrsCacheContentsFile  
function = dmg_brw32->DmgBrsReturnConvertedDocument  
function = cpyrule->ExecISICpy  
  
[ ReqType:CONVERTDPA ]
```

```

function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsReturnConvertedDocument
function = dprw32->DPRRetrieveDPA
function = dprw32->DPRPrint
[ ReqType:CONVERTPRINTSTREAM ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsReturnConvertedDocument
function = dprw32->DPRSetConfig
function = mtcw32->MTCLoadFormset
function = dprw32->DPRPrint
[ ReqType:CONVERTTOIMAGEPDF ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsReturnConvertedDocument
function = Tpdw32->TPDCreateFormset (not needed if the INI option
[ReturnConvertedDocument]UseTPDForImagePDF is OFF)

```

- 3 The following is a rules list that can be used to do various kinds of Document contents file conversions, and can also return a contents file, a page of a contents file, or a thumbnail image of a page of a contents file.

```

[ ReqType:RETURNFILE ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsReturnDocument

```

```
function = dmgb32->DmgbReturnConvertedDocument
function = dmgb32->DmgbReturnDocumentPage
function = dmgb32->DmgbReturnThumbnail
function = cpyrule->ExecISICpy
function = dprw32->DPRRetrieveDPA
function = Tpdw32->TPDCreateFormset (not needed if the INI option
[ReturnConvertedDocument]UseTPDForImagePDF is OFF)
function = dprw32->DPRSetConfig
function = mtcw32->MTCLoadFormset
function = dprw32->DPRPrint
```

Notice that although CONVNAME is a required input attachment variable, DmgbReturnConvertedDocument does not fail if the variable is missing. Instead, it does nothing. All of the DmgbReturn... rules that deal with the contents file (as opposed to the annotations file) work this way, so they can all be put on a single rules list. All of these rules require the outputs of DmgbCacheContentsFile and a URL Tail with a unique variable name, and they all do nothing if their specific URL Tail is missing. Thus, you can simply run the rules list and provide the necessary URL Tail, and the desired rule will be activated.

We do not necessarily recommend that you set up a rules list like that shown above, since it is usually clearer to separate rules lists by their specific purpose. The example simply illustrates that such a rules list is possible.

ReqType changes	[ReqTypes] Failure if the rule fails.
	[ReqTypes] Redirection if TEMP_FILE exists.
INI options	[ReturnConvertedDocument]
	[Rendering] DocumentConversion
	[TempCache]
Validations	Session and Reqtype
Platform restrictions	Conversion of FPP to XML or of any file type except JPG or TIF to image PDF is Windows only.
See also	DmgbDumpFile
	DmgbReturnAnnotations
	DmgbReturnDocument
	DmgbCacheContentsFile

DMGBRSRETURNDOCUMENT

DmgBrsReturnDocument returns a Document contents file.

Usage

```
function = dmg_brw32->DmgBrsReturnDocument
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document to return. Required.
SUBDOC_NUM	The one-based Subdocument number for the Subdocument to return. Optional if the Document is Compound, else ignored. For a Subdocument to be returned, the Document must be Compound and SUBDOC_NUM must be supplied. Otherwise, the entire Document contents file is returned.
DECOMPRESS	Set this to 1 to cause the decompressed version of a Document contents file of type DCZ to be returned, except when the DCZ contains FPP data. Set it to 0 to cause a Document contents file of type DCZ to be returned as DCZ. Decompression is available only on Windows. Optional. Default is the INI option [ReturnDocument] Decompress.
ORIGNAME	A URL Tail for the Document contents file. The content of this variable isn't relevant to the operation of this rule, but its existence is. Required. If this variable is missing, the rule does not fail. It simply does nothing.

Attachment outputs

Variable	Description
HTTP_STATUS	The status for creating an HTTP header. If the rule fails, this is set to "404 Not Found". Otherwise, the default status set by DmgBrsCopyAttachment is left unchanged.
HTTP_MIMETYPE	The MIME type for creating an HTTP header. Depends on the type of the contents file.
HTTP_CONTENTLENGTH	The length of the file in bytes.
TARGET_FILE	The path of the file.
TARGET_FILE_NAME	The name (including extension) from the end of TARGET_FILE.
TEMP_FILE	The path of a temp file for direct download by a web server. Exists only if the INI option [TempCache] TempFiles is ON. The file will be in the directory configured in the INI option [TempCache] DirPath.
TEMP_FILE_NAME	The name (including extension) from the end of TEMP_FILE.

Remarks

DmgBrsReturnDocument returns a Document contents file in its original filetype. The rule DmgBrsCacheContentsFile must precede this rule on the rules list. To return a converted

version of the Document contents file, use DmgBrsReturnConvertedDocument. To return the Document's ANO Annotations file, use DmgBrsReturnAnnotations.

If the Document file is compressed into a DCZ file, use DECOMPRESS to determine whether it will be decompressed before being returned. DECOMPRESS does not apply to compressed FPP files, and is available only on Windows.

Examples	See the example for DmgBrsReturnConvertedDocument that shows DmgBrsReturnDocument.
ReqType changes	[ReqTypes] Failure if the rule fails. [ReqTypes] Redirection if TEMP_FILE exists.
INI options	[ReturnDocument] [TempCache]
Validations	Session and Reqtype
Platform restrictions	DECOMPRESS = 1 is Windows only.
See also	DmgBrcDumpFile DmgBrsReturnAnnotations DmgBrsReturnConvertedDocument DmgBrsReturnDocumentPage DmgBrsReturnThumbnail DmgBrsCacheContentsFile

DMGBRSRETURNDOCUMENTPAGE

DmgBrsReturnDocumentPage returns a page of a Document contents file, converted to JPG, black and white TIF, image PDF, EMF, or DCZ (compressed EMF).

Usage

```
function = dmg_brw32->DmgBrsReturnDocumentPage
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document whose page will be returned. Required.
SUBDOC_NUM	The one-based Subdocument number for the Subdocument whose page is to be returned. Required if the Document is Compound, else ignored.
PAGENAME	A URL Tail for the converted Document page. The extension on the value of this variable indicates the desired converted FileType. Parts of the URL Tail other than the extension are ignored. If there is no extension, the INI option [Rendering] PageConversion is used to determine the converted FileType. Required. If this variable is missing, the rule does not fail. It simply does nothing. If the converted FileType is not one of those listed, then the rule fails.
PAGE	The page for which the image is to be returned. When the Document is a Compound Document, this is the page in Subdocument SUBDOC_NUM. One-based. If missing, defaults to 1.
ROT	The rotation state of the page image. Optional. Default is no rotation. Valid values are 0, 1, 2, or 3 for 0, 90, 180, or 270 degrees clockwise rotation.

Attachment outputs

Variable	Description
HTTP_STATUS	The status for creating an HTTP header. If the rule fails, this is set to "404 Not Found". Otherwise, the default status set by DmgBrsCopyAttachment is left unchanged.
HTTP_MIMETYPE	The MIME type for creating an HTTP header, appropriate for the type of file: "image/jpeg" for JPG, "image/tiff" for TIF, "application/pdf" for PDF, and "application/octet-stream" for EMF or DCZ.
HTTP_CONTENTLENGTH	The length of the file in bytes.
TARGET_FILE	The path of the file.
TARGET_FILE_NAME	The name (including extension) from the end of TARGET_FILE.
TEMP_FILE	The path of a temp file for direct download by a web server. Exists only if the INI option [TempCache] TempFiles is ON. The file will be in the directory configured in the INI option [TempCache] DirPath.

TEMP_FILE_NAME	The name (including extension) from the end of TEMP_FILE.
PDFNAME	Appears only when the page is being converted to PDF. A required input for TPDCreateFormset. The path to which TPDCreateFormset should write the PDF. The path will be in the Bridge File Cache. This variable is written to the Input Queue.
SRCTYPE	Appears only when the page is being converted to PDF. A required input for TPDCreateFormset. This will contain "JPG". This variable is written to the Input Queue.
TIFFNAME	Appears only when the page is being converted to PDF. A required input for TPDCreateFormset. Contains "1". This variable is written to the Input Queue.
TIFFNAME1.NAME	Appears only when the page is being converted to PDF. A required input for TPDCreateFormset. A record set with one record listing the path to the page file for the PDF. The path will be in the Bridge File Cache. This variable is written to the Input Queue.

On Windows, if the INI option [ReturnDocumentPage]UseTPDForPDF is ON and the page is being converted to PDF, the following additional outputs will appear:

Variable	Description
PDFNAME	A required input for TPDCreateFormset. The path to which TPDCreateFormset should write the PDF. The path will be in the Bridge File Cache. This variable is written to the Input Queue.
SRCTYPE	A required input for TPDCreateFormset. This will contain "JPG". This variable is written to the Input Queue.
TIFFNAME	A required input for TPDCreateFormset. Contains "1". This variable is written to the Input Queue.
TIFFNAME1.NAME	A required input for TPDCreateFormset. A record set with one record listing the path to the page file for the PDF. The path will be in the Bridge File Cache. This variable is written to the Input Queue.

Remarks

DmgBrsReturnDocumentPage returns a page of a Document contents file, converted to JPG, black and white TIF, image PDF, EMF, or DCZ (compressed EMF). The typical FileType is JPG. The others are used more rarely. The rule DmgBrsCacheContentsFile must precede this rule on the rules list.

See DmgBrsReturnConvertedDocument for a description of the difference between image and text PDF.

EMF and DCZ, which will be a compressed EMF, may optionally be used when a page of a DCD contents file is to be presented in vector format. See DmgBrsPresentDocument and DmgBrsPresentDocumentPage.

For bitmapped FileTypes, which are all the FileTypes other than EMF and DCZ, the INI options [ReturnDocumentPage] MaxWidthInPixels and MaxHeightInPixels limit the width and height of the image.

The outputs of DmgBrsPresentDocument or DmgBrsPresentDocumentPage can easily be used to create the inputs for DmgBrsReturnDocumentPage.

DmgBrsReturnDocumentPage uses the Shuttle library, plus libraries Shuttle depends on such as the Printstream library or the INSO library, to create page images. If a page image cannot be created for some reason, then the file configured in the INI option [ReturnDocumentPage] MissingFile is returned.

Examples A rules list for DmgBrsReturnDocumentPage might look like the following if PDF was a possible page conversion type.

```
[ ReqType:CONVERTTOIMAGEPDF ]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmgb32->DmgBrsCopyAttachment
  function = dmgb32->DmgBrsValidateSession
  function = dmgb32->DmgBrsCacheContentsFile
  function = dmgb32->DmgBrsReturnDocumentPage
  function = Tpdw32->TPDCreateFormset
```

Notice that the rule TPDCreateFormset is configured on the above rules list to do the conversion to PDF after DmgBrsReturnDocumentPage runs. If your Bridge application does not convert pages to PDF, or if the INI option [ReturnDocumentPage]UseTPDForPDF is OFF, then you would not need the rule TPDCreateFormset.

See also the example for DmgBrsReturnConvertedDocument, and the examples for DmgBrsPresentDocumentPage.

ReqType changes [ReqTypes] Failure if the rule fails.

[ReqTypes] Redirection if TEMP_FILE exists.

INI options [ReturnDocumentPage]
[Rendering] PageConversion
[TempCache]

Validations Session and Reqtype

Platform restrictions Windows only.

See also DmgBrcDumpFile
DmgBrsPresentDocument
DmgBrsPresentDocumentPage
DmgBrsReturnDocument
DmgBrsCacheContentsFile
DmgBrsReturnConvertedDocument

DMGBRSRETURNTHUMBNAIL

DmgBrsReturnThumbnail returns a thumbnail image of a Document page.

Usage

```
function = dmg_brw32->DmgBrsReturnThumbnail
```

Attachment inputs

Variable	Description
Contents File Info	The outputs of DmgBrsCacheContentsFile for the Document containing the page for which the thumbnail is to be returned. Required.
SUBDOC_NUM	The one-based Subdocument number for the Subdocument whose page is to be thumbnailed. Required if the Document is Compound, else ignored.
PAGE	The one-based page number for which the thumbnail is to be returned. When the Document is a Compound Document, this is the page in Subdocument SUBDOC_NUM. Required.
USE_MISSING	Set this to 1 to return the thumbnail file configured in the INI option [Thumbnails] MissingFile. If the option is empty or not 1, an ordinary thumbnail will be created and returned.
THUMBNAME	A URL Tail for the thumbnail image. The content of this variable isn't relevant to the operation of this rule, but its existence is. Required. If this variable is missing, the rule does not fail. It simply does nothing.

Attachment outputs

Variable	Description
HTTP_STATUS	The status for creating an HTTP header. If the rule fails, this is set to "404 Not Found". Otherwise, the default status set by DmgBrsCopyAttachment is left unchanged.
HTTP_MIMETYPE	The MIME type for creating an HTTP header. "image/jpeg"
HTTP_CONTENTLENGTH	The length of the file in bytes.
TARGET_FILE	The path of the file.
TARGET_FILE_NAME	The name (including extension) from the end of TARGET_FILE.
TEMP_FILE	The path of a temp file for direct download by a web server. Exists only if the INI option [TempCache] TempFiles is ON. The file will be in the directory configured in the INI option [TempCache] DirPath.
TEMP_FILE_NAME	The name (including extension) from the end of TEMP_FILE.

Remarks

DmgBrsReturnThumbnail returns a thumbnail image of a Document page. The rule DmgBrsCacheContentsFile must precede this rule on the rules list.

If USE_MISSING is 1, this rule returns the file configured in the INI option [Thumbnails] MissingFile. DmgBrsReturnThumbnail should not be used when the system is not configured for thumbnails.

Thumbnails cannot be made from Document files that are DPA files or from DCZ files that contain compressed FPP files. If a thumbnail cannot be made for some reason or if the rule fails, the file configured in the INI option [Thumbnails] MissingFile will be returned.

Thumbnail images are always of type JPG. The size of thumbnails produced is controlled by the INI options [Thumbnails] MaxWidthInPixels and MaxHeightInPixels. The darkness can be adjusted with [Thumbnails] Darkness.

ReqType changes [ReqTypes] Redirection if TEMP_FILE exists.

INI options

- [Thumbnails] Thumbnails
- [Thumbnails] MissingFile
- [Thumbnails] MaxWidthInPixels
- [Thumbnails] MaxHeightInPixels
- [Thumbnails] Darkness
- [Rendering] DPAExtension
- [TempCache]

Validations Session and Reqtype

Platform restrictions Windows only.

See also

- DmgBrsPresentDocument
- DmgBrsCacheContentsFile

DMGBRSRUNDOCUMANAGEDIAGNOSTIC

DmgBrsRunDocumanageDiagnostic runs a Documanage Diagnostic Test and executes the designated failure action if the test detects a problem.

Usage

```
function = dmg_brw32->DmgBrsRunDocumanageDiagnostic
```

Attachment inputs

Variable	Description
DMGDIAG_NAME	The name of the Documanage Diagnostic Test to run. This rule does not attempt to run any test if this variable is missing or empty, or if the test named in this variable does not exist in Documanage. Optional.
IDSACTIONONFAIL	The action to take if the attempt to run the test in DMGDIAG_NAME fails, or if the test runs and detects a problem. Valid values are "Pause", to cause IDS to go into Pause mode, "Stop", to cause IDS to stop, and "Restart", to cause IDS to restart, all case insensitive. Any other value is interpreted to mean "NoAction". Optional. Default is "NoAction".

Attachment outputs

Variable	Description
BRIDGEINITIALIZED	Contains 1 if the Bridge is initialized, else contains 0. No other outputs appear if this contains 0. This variable is output even if no input attachment variables were supplied.
SESS_ROUTERIP	The IP address of the Documanage router, to identify the Documanage system. Based on the INI option [Documanage] Router.
DMGDIAG_NAMEDOESNTEXTST	Contains 1 if the Documanage Diagnostic test named in DMGDIAG_NAME does not exist in Documanage. In this case, all of the following variables whose names begin with DMGDIAG_ will contain 0.
DMGDIAG_RUNATTEMPTED	Contains 1 if an attempt was made to run the test, else contains 0.
DMGDIAG_RUNSUCCEEDED	Contains 1 if the test could be run, else contains 0.
DMGDIAG_TESTSUCCEEDED	Exists only if DMGDIAG_RUNSUCCEEDED contains 1. Contains 1 if the test succeeded, that is, if it detected no problem. Contains 0 if the test failed, that is, it detected some kind of problem.
DMGDIAG_TESTFAILUREMESSAGE	Exists only if DMGDIAG_TESTSUCCEEDED contains 0. A message that describes the nature of the test failure.

IDS ACTIONONFAIL	The IDS action to take if the test could not be run or ran and detected a problem. Will be "Pause", "Stop", "Restart", or "NoAction".
IDS ACTIONONFAIL_RAN	Contains 1 if IDS ACTIONONFAIL ran, else contains 0. If the test could not be run or ran and detected a problem, and if IDS ACTIONONFAIL was not "NoAction", then IDS ACTIONONFAIL will run.

- Remarks**
- DmgBrsRunDocumangeDiagnostic runs a Documange Diagnostic Test and executes the designated failure action if the test detects a problem.
- DmgBrsRunDocumangeDiagnostic is primarily for administrative use, especially for testing the configuration of the system when a test fails (see the discussion of [DMGDiagnosics] TestFailDiagnostic below), but it may also be used for periodic monitoring and reporting of the condition of the system. DmgBrsIdle can provide periodic monitoring, but it doesn't provide reporting.
- See the Documange Administrator's Guide for information on how to set up and name Documange Diagnostic Tests.
- In interpreting the output attachment variables of this rule, it is important to distinguish whether the Documange Diagnostic Test named in DMGDIAG_NAME was able to be run from whether it detected a problem. The test may not be able to be run due to loss of a network connection with Documange, or other such conditions. If it is able to be run, then it may or may not detect a problem. If the test couldn't be run, then the question of whether it detected a problem is irrelevant.
- The Documange Diagnostic Test named in DMGDIAG_NAME may be the test configured in the INI option [DMGDiagnosics] TestFailDiagnostic. In that case, DmgBrsRunDocumangeDiagnostic will always respond as if the test ran and detected a problem. TestFailDiagnostic is not an actual Documange Diagnostic Test, but merely a dummy test name to use to test what would happen if an actual Documange Diagnostic Test were to detect a problem.

- Examples**
- The following rules list may be used to run DmgBrsRunDocumangeDiagnostic.

```
[ ReqType:RUN_DIAGS ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsRunDocumangeDiagnostic
```

If your Bridge application uses CONFIG, then you will want to provide a CONFIG variable or add DmgBrsValidateSession to the above rules list. See DmgBrsValidateSession.

- ReqType changes**
- [ReqTypes] Failure if the rule fails, although this rule has no expected failure conditions.

- INI options**
- [Documange] Router
[DMGDiagnosics] StartupDiagnostic
[DMGDiagnosics] TestFailDiagnostic

- Validations**
- None.

- Platform restrictions**
- None.

See also DmgBrsIdle
 DmgBrsInit
 DmgBrsCheckPausedServer

DMGBRSSETDOCUMENTRETENTION

DmgBrsSetDocumentRetention sets the retention date of a Document.

Usage function = dmg_brs32->DmgBrsSetDocumentRetention

Attachment inputs

Variable	Description
A Version Document Specification	The Document whose retention date is to be set. Required.
DOC_RETENTIONDATE	The new retention date of the Document Version. The date must be in a format the Bridge can recognize or the rule will fail. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The ID Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks DmgBrsSetDocumentRetention sets the retention date of a Document. In setting the date, you must follow the Documange requirements for setting the retention date of a Document, or the rule will fail.

This rule is available only if the Bridge is logged into Documange 6.4 Service Release 4 or higher and is using Dmg_Api 1.4.1 or later.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and ReqType

Platform restrictions None.

May be followed by DmgBrsPresentDocument
DmgBrsPresentFolder
DmgBrsPresentProject

See also `DmgBrsPresentDocument`

DMGBRSSETINDICATORS

DmgBrsSetIndicators replaces the 1 or 0 value of an attachment variable with a designated new value.

Usage

```
function = dmg_brw32-
>DmgBrsSetIndicators,varname,truevalue,falsevalue[,...]
```

Parameters

Parameter	Description
varname	The name of an attachment variable or a RecordSet Wildcard on the output queue. Case-independent.
truevalue	The new value for varname if varname contains 1. May be empty.
falsevalue	The new value for varname if varname contains 0. May be empty.

The parameters for this rule come in sets of three: varname,truevalue,falsevalue. Any number of these sets may be passed to the rule, separated by commas. As a practical matter, if there are several indicators to be set, the rules list may be more readable if DmgBrsSetIndicators is called multiple times, with one or a few indicators passed to it each time.

Attachment inputs

Variable	Description
varname(s)	The variable or variables designated in the parameter(s) varname.

Attachment outputs

None.

Remarks

DmgBrsSetIndicators replaces the 1 or 0 value of an attachment variable with a designated new value. DmgBrsSetIndicators does its work on the reverse run of the rules list, and operates only on variables on the output queue.

The word “indicator” in this context is used to refer to an attachment variable that has a value of 1 or 0 to indicate whether some condition is true or false. For displaying the value of an indicator to a user, values other than 1 or 0 may be preferable.

DmgBrsSetIndicators ignores any attachment variable that it cannot find, and ignores any value that is not 1 or 0.

Notice that DmgBrsSetIndicators replaces the values of attachment variables. In some cases, a Bridge Application may depend on having a value of 1 or 0 in an indicator. In such cases, use DmgBruCopyAttachmentVar to make a copy of the indicator under a different name for display to the user, and call DmgBrsSetIndicators for the copy. See the examples below.

Examples

In this example, we'll use DmgBrsSetIndicators to set the values of some output attachment variables of DmgBrsQueryDocuments.

```
[ ReqType:QUERY_DOCS ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsSetIndicators,HasMore,More Hits
Exist,,HITS*.DOC_APPROVED,A
```

```
function = dmg_brw32-  
>DmgBruCopyAttachmentVar,runr,output,MOREHITS,output,HASMORE  
function = dmg_brw32->DmgBrsQueryDocuments
```

In the above example, we use `DmgBrsSetIndicators` to set the value of a single attachment variable and of a field in a record set.

The rule `DmgBrsQueryDocuments` outputs a variable called `MOREHITS`, that contains 1 or 0 to indicate whether all of the Document hits have been returned or not. Assume that our Bridge application depends on the fact that `MOREHITS` will contain 1 or 0, but that we also want to use a variable to display an indication to the user that more hits exist. In the example, we use `DmgBruCopyAttachmentVar` to copy `MOREHITS` to another variable called `HASMORE`. We then pass `HASMORE` to `DmgBrsSetIndicators`. The parameters of `DmgBrsSetIndicators` will cause `HASMORE`'s new values to be "More Hits Exist" if it contained 1, and empty if it contained 0.

In the hits record set output by `DmgBrsQueryDocuments`, there will be a Document Property called `DOC_APPROVED`, which will contain 1 or 0 to indicate whether the Document is approved. In the example, we use `DmgBrsSetIndicators` to set `DOC_APPROVED` to "A" if it contains 1 and empty if it contains 0, for every record in the record set.

Notice that, since `DmgBrsSetIndicators` works on the reverse run of the rules list, it may be placed above `DmgBrsQueryDocuments` on the rules list since `DmgBrsQueryDocuments` works on the forward run.

See also the example for `DmgBrsQuery` that uses `DmgBrsSetIndicators`.

ReqType changes	[ReqTypes] Failure if the rule fails, although this rule has no expected failure conditions.
INI options	None.
Validations	None.
Platform restrictions	None.
See also	<code>DmgBruCopyAttachmentVar</code>

DMGBRSSETUPCOMPOUNDDOCUMENT

DmgBrsSetupCompoundDocument allows the construction of a transaction that will return either a file that can be viewed in a browser or, if that is not possible, the results of DmgBrsPresentCompoundDocument.

Usage

```
function = dmg_brw32->DmgBrsSetupCompoundDocument
```

Attachment inputs

Variable	Description
Contents file info	The outputs of DmgBrsCacheContentsFile. Required.

Attachment outputs

Variable	Description
RETURNING_FILE	Contains 1 if DmgBrsSetupCompoundDocument is setting up a rule that returns a file (DmgBrsReturnDocument or DmgBrsReturnConvertedDocument). Contains 0 if DmgBrsSetupCompoundDocument is setting up the rule DmgBrsPresentCompoundDocument.

If the Compound Document contains a single Subdocument file that is viewable in a browser, DmgBrsSetupCompoundDocument outputs the following attachment variables to allow DmgBrsReturnDocument to run and to suppress DmgBrsPresentCompoundDocument:

Variable	Description
ORIGNAME	The trigger variable for DmgBrsReturnDocument.
SUBDOC_NUM	Causes DmgBrsReturnDocument to return the Subdocument file.
RUN_PRESENTCOMPOUNDDOCUMENT	Contains 0. This prevents DmgBrsPresentCompoundDocument from running.

If all of the Subdocuments in the Compound Document can be converted to image PDF with DmgBrsReturnConvertedDocument, DmgBrsSetupCompoundDocument outputs the following attachment variables to allow DmgBrsReturnConvertedDocument to run and to suppress DmgBrsPresentCompoundDocument:

Variable	Description
CONVNAME	The trigger variable for DmgBrsReturnConvertedDocument.
RUN_PRESENTCOMPOUNDDOCUMENT	Contains 0. This prevents DmgBrsPresentCompoundDocument from running.

If neither of the above conditions (which allow returning a file) apply, DmgBrsSetupCompoundDocument outputs the following attachment variables to direct DmgBrsPresentCompoundDocument to run.

Variable	Description
RUN_PRESENTCOMPOUNDDOCUMENT	Contains 1.

Remarks

DmgBrsSetupCompoundDocument allows the construction of a transaction that will return either a file that can be viewed in a browser or, if that is not possible, the results of DmgBrsPresentCompoundDocument. This allows a Bridge application to use a single transaction for handling a request to get a Compound Document.

DmgBrsSetupCompoundDocument will attempt to setup rules that return a file that is viewable in a browser. JPEG and PDF files are considered to be viewable in a browser. If it cannot setup such rules, it sets up DmgBrsPresentCompoundDocument to return information about the Compound Document.

If the document is not a compound document, this rule will fail.

Note that this rule deletes the variables SUBDOC_NUM, ORIGNAME and CONVNAME except in those cases where it explicitly sets one or more of them.

Typically this rule is on a rules list followed by DmgBrsReturnConvertedDocument, DmgBrsReturnDocument, and DmgBrsPresentCompoundDocument. It is important to make sure that DmgBrsPresentCompoundDocument is after DmgBrsReturnConvertedDocument and DmgBrsReturnDocument on the rules list, or it will cause the latter two rules to run because it outputs their trigger variables. See the example.

Examples

The following is a typical rules list for the use of DmgBrsSetupCompoundDocument. Note that DmgBrsPresentCompoundDocument is after DmgBrsReturnConvertedDocument and DmgBrsReturnDocument so that its outputs do not cause the latter two rules to run.

```
[ ReqType:GET_CMPND_DOC ]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsSetupCompoundDocument
function = dmg_brw32->DmgBrsReturnDocument
function = dmg_brw32->DmgBrsReturnConvertedDocument
function = dmg_brw32->DmgBrsPresentCompoundDocument
```

- ReqType changes** [ReqTypes] Failure if the rule fails
- INI optionsINI Options** None
- Validations** Session and Reqtype
- Platform restrictions** None
- May be followed by** DmgBrsReturnDocument
DmgBrsReturnConvertedDocument
DmgBrsPresentCompoundDocument
- See also** DmgBrsReturnDocument
DmgBrsReturnConvertedDocument

DmgBrsPresentCompoundDocument

DmgBrsPresentDocument

DmgBrsPresentSubdocument

DmgBrsCacheContentsFile

DMGBRSSETUPFOLDER

DmgBrsSetupFolder finds a Folder and puts its Folder Specification on the attachment, in preparation to run other rules such as DmgBrsCreateDocument.

Usage

```
function = dmg_brw32->DmgBrsSetupFolder
```

Attachment inputs

Variable	Description
A Folder Specification	If DmgBrsSetupFolder finds a Folder Specification, then it ignores all other inputs—whether otherwise required or not—and doesn't produce any outputs. It has no need to do anything if a Folder Specification already exists. Optional.
FLD_LEVELNUMBER	The Folder Level of the Folder to find. Optional. Defaults to 1.
CABINET	The Cabinet of the Folder to find. Required.
UNIQUE_FLD	Set this to 1 to cause DmgBrsSetupFolder to fail if it finds more than one Folder that matches the search criteria. Set it to 0 to cause DmgBrsSetupFolder to report the Folder Specification of the first Folder that it finds that matches the search criteria. Optional. Default is [SetupFolder] UniqueFld.
FLD_FILTER	An Extended Folder Property filter for the search. The use of this variable must be enabled by turning on the corresponding option for this rule in dmg_brs.ini. Optional, but search criteria must be provided by this or either of the following two methods, or DmgBrsSetupFolder will fail.
QUERY_XFLD_...	SQL Builder variables describing an Extended Folder Property filter. See the description of SQL Builder syntax in the Bridge General Reference. This is an alternative to FLD_FILTER, and will be ignored if FLD_FILTER exists. Optional.
XFLD_Lx_ propertyname	Extended Properties of a Folder to find, where x is FLD_LEVELNUMBER, and propertyname is the Bridge-encoded Folder Property Name. This is an alternative to FLD_FILTER and QUERY_XFLD_... and will be ignored if either of those exists. Optional.

Attachment outputs

Variable	Description
FLD_LEVELNUMBER	The LevelNumber Property of the Folder that was found.
FLD_ID	The Id Property of the Folder that was found.
FLD_KEYSTRING	The KeyString Property of the Folder that was found.
HFLD	A Dmg_API handle to the Folder that was found. The reverse run of DmgBrsSetupFolder removes this from the attachment. It serves as an input to DmgBrsCreateDocument when that rule follows this one on a rules list. This variable is written to the Input Queue.

Remarks DmgBrsSetupFolder finds a Folder and puts its Folder Specification on the attachment. This is typically in preparation to run DmgBrsCreateDocument in a situation where information about the Folder in which the Document will be created is known, but a Folder Specification isn't available. One application is in automated systems that create Documents and select the Document's Folder based on some known Extended Properties of the Folder. However, DmgBrsSetupFolder can precede any rule that requires a Folder Specification as its major input, such as DmgBrsPresentFolder.

See DmgBrsCreateDocument for a discussion of when to use DmgBrsSetupFolder before DmgBrsCreateDocument.

There are three ways to pass search criteria to DmgBrsSetupFolder: A Folder filter, SQL Builder variables, and Extended Folder Properties. DmgBrsSetupFolder looks for the method of passing criteria in that order. One of these three must be provided, or the rule fails.

Examples In this example, we will search for a Folder and create a Document in one transaction. Here is what the rules list might look like.

```
[ ReqType:CREATE_DOC]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsSetupFolder
  function = dmg_brw32->DmgBrsCreateDocument
```

Here's what our input attachment variables might be.

```
CABINET = Customers
XFLD_L1_Last$20Name = Doe
XFLD_L1_First$20Name = Jane
DOC_FILEPATH = e:\importfiles\myfile.jpg
DOC_CATEGORY = Photo
```

Notice that we supplied two Extended Folder Properties to identify the Folder, and DmgBrsSetupFolder took care of finding the folder and setting up the Folder Specification for DmgBrsCreateDocument.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	[SetupFolder]
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsCreateDocument DmgBrsDeleteFolder DmgBrsPresentFolder
See also	DmgBrsCreateDocument DmgBrsCreateFolder DmgBrsLogPropertySets

DMGBRSUSPENDPROJECT

DmgBrsSuspendProject suspends a Project.

Usage `function = dmgb_brw32->DmgBrsSuspendProject`

Attachment inputs

Variable	Description
A Project Specification	The Project to suspend.
I_SUSPENDFOR	A comment that describes why the Project was suspended. Required.
I_SUSPENDTILL	A date on which the Project's suspension will expire unless it has been resumed before then. The date must be in a format that the Bridge can recognize, or DmgBrsSuspendProject will fail. Required.

Attachment outputs None.

Remarks DmgBrsSuspendProject suspends a Project. The Project will remain suspended until the date in I_SUSPENDTILL or until it is resumed (with DmgBrsResumeProject).

Examples The following is an example of a rules list that could be used to suspend a Project and then return a list of pending Tasks, all in one transaction, so that the user could immediately proceed to picking the next Task after suspending a Project.

```
[ ReqType:SUSPEND_PRJ]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmgb_brw32->DmgBrsCopyAttachment
  function = dmgb_brw32->DmgBrsValidateSession
  function = dmgb_brw32->DmgBrsSuspendProject
  function = dmgb_brw32->DmgBrsGetTasks
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsGetTasks

See also DmgBrsResumeProject

DMGBRSUNDODOCUMENTCHECKOUT

DmgBrsUndoDocumentCheckOut undoes a Document checkout.

Usage

```
function = dmg_brw32->DmgBrsUndoDocumentCheckOut
```

Attachment inputs

Variable	Description
A Current Document Specification	The checked out copy of a Document or the Document. Required.

Attachment outputs

Variable	Description
CABINET	The Cabinet of the Document.
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
DOC_VERS	The DOC_VERS of the Document.
DOCSPEC	A DOCSPEC for the Document.
DOC_MAJORVERSION	The MajorVersion Property of the Document.
DOC_MINORVERSION	The MinorVersion Property of the Document.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsUndoDocumentCheckOut undoes a Document checkout. The Specification passed in may be for the main Document or for the checked out copy of the Document.

After the undo, the checked out copy of the Document will no longer exist, so all of the output variables refer to the main Document. Because of this, the values of the Document Specification variables may be different when they are output. For example, if the Specification for the checked out copy of the Document was passed in, then CABINET will be the Cabinet of the checked out Document on input and will be the CABINET of the main Document, which may be a different Cabinet, on output.

Only the user who checked out the Document may undo the checkout. Otherwise, the rule will fail.

Examples The following rules list will undo a Document checkout, and then present the Document. The only inputs required are those required by DmgBrsUndoDocumentCheckOut and DOC_MAJORVERSION = -1, to complete a current Version Document Specification for DmgBrsCacheContentsFile.

```
[ ReqType:UNDO_DOC_CO]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsUndoDocumentCheckOut
  function = dmg_brw32->DmgBrsCacheContentsFile
  function = dmg_brw32->DmgBrsPresentDocument
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsPresentDocument

DmgBrsPresentFolder

DmgBrsPresentProject

All of the above will, of course, operate relative to the Document and not to the checked out copy, which no longer exists after DmgBrsUndoDocumentCheckOut runs.

See also DmgBrsCheckOutDocument

DMGBRSUNLOCKANNOTATIONS

DmgBrsUnlockAnnotations unlocks locked Annotations.

Usage

```
function = dmg_brw32->DmgBrsUnlockAnnotations
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to unlock Annotations. Required.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsUnlockAnnotations unlocks locked Annotations. After Annotations have been updated, they should be unlocked so that they're available for others to modify.

Only the user that locked the Annotations can unlock them. Otherwise, the rule will fail.

Examples

The following rules list will unlock the Annotations for a Document, and then present the Document. DmgBrsUnlockAnnotations requires a Current Document Specification, but DmgBrsCacheContentsFile requires a Version Document Specification. To satisfy the latter, simply put DOC_MAJORVERSION = -1 on the output attachment to indicate the current Document.

```
[ ReqType:UNLOCK_ANNOTS]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsUnlockAnnotations
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsPresentDocument
```

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsPresentProject
See also	DmgBrsLockAnnotations DmgBrsReturnAnnotations DmgBrsUpdateAnnotations

DMGBRSUPDATEANNOTATIONS

DmgBrsUpdateAnnotations updates the Annotations File of a Document.

Usage

```
function = dmg_brw32->DmgBrsUpdateAnnotations
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to update Annotations. If the Document is checked out, this must be the checked out copy. Required.
ANNOTS_UPLOADERR	Except for the name of the variable, identical to DOC_UPLOADERR in DmgBrsCreateDocument. See DmgBrsCreateDocument. Optional.
ANNOTS_FILEPATH	The path to the Annotations file. The name of this file is unimportant, so it may be a temporary name. Required.
ANNOTS_DELFIL	Set this to 1 to cause the file in ANNOTS_FILEPATH to be deleted after DmgBrsUpdateAnnotations runs, whether it succeeds or not. Set this to 0 to prevent file deletion. Optional. Default is the INI option [UpdateAnnotations] DeleteFile.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document was in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.

Remarks

DmgBrsUpdateAnnotations updates the Annotations File of a Document. The Annotations File of the Document is completely replaced by the file in ANNOTS_FILEPATH, so be certain that the file contains correct ANO Annotations for all pages of the Document that have Annotations.

If the user has not checked out the Document or locked the Annotations, DmgBrsUpdateAnnotations will fail. If the Annotations were locked, then the change becomes effective immediately. If the Document was checked out, then the change becomes effective when the Document is checked in.

This rule requires Documanager Server 6.4 service release 7 or later.

Examples The following rules list will update the Annotations for a Document and unlock them in one transaction. The Current Document Specification that DmgBrsUpdateAnnotations requires suffices as inputs for DmgBrsUnlockAnnotations.

```
[ ReqType:UNLOCK_ANNOTS]
  function = atcw32->ATCLogTransaction
  function = atcw32->ATCLoadAttachment
  function = atcw32->ATCUnloadAttachment
  function = dmg_brw32->DmgBrsCopyAttachment
  function = dmg_brw32->DmgBrsValidateSession
  function = dmg_brw32->DmgBrsUpdateAnnotations
  function = dmg_brw32->DmgBrsUnlockAnnotations
```

ReqType changes [ReqTypes] Failure if the rule fails.

INI options [UpdateAnnotations]

Validations Session and Reqtype

Platform restrictions None.

May be followed by DmgBrsUnlockAnnotations
DmgBrsPresentDocument
DmgPrsPresentFolder
DmgPrsPresentProject

See also DmgBrsLockAnnotations
DmgBrsReturnAnnotations
DmgBrsUnlockAnnotations
DmgBrsValidateSession, particularly the parameter

DMGBRSUPDATEDIARY

DmgBrsUpdateDiary modifies the Properties of a Diary.

Usage

```
function = dmg_brw32->DmgBrsUpdateDiary
```

Attachment inputs

Variable	Description
DIA_KEYSTRING	The Keysting property of the Diary to update, typically obtained from DmgBrsQueryDiaries. Required.
Diary Properties	The names of any Diary Properties to update. See the list of Properties in the attachment outputs for information on which Properties can be edited. Any that are not editable are ignored. Any number of Diary Properties may be passed in. Optional.

Attachment outputs

Variable	Description
DIARY_TITLE	The Title Property of the Diary (editable).
DIA_DESCRIPTION	The Description Property of the Diary (editable).
DIA_PRIORITY	The Priority Property of the Diary (editable).
DIA_CREATEDON	The CreatedOn Property of the Diary (not editable).
DIA_STARTON	The StartOn Property of the Diary (editable).
DIA_DUEON	The DueOn Property of the Diary (editable).
DIA_COMPLETEDON	The CompletedOn Property of the Diary (editable).
DIA_LASTMODIFIED ON	The LastModifiedOn Property of the Diary (not editable).
DIA_NEXTALERTTI ME	The NextAlertTime Property of the Diary (editable).
DIA_ASSIGNEDBY	The AssignedBy Property of the Diary (not editable).
DIA_OWNER	The Owner Property of the Diary (not editable).
DIA_REFERENCE	The Reference Property of the Diary (editable).
DIA_STATUS	The Status Property of the Diary (editable).
DIA_ID	The Id Property of the Diary (not editable).
DIA_KEYSTRING	The Keysting Property of the Diary (not editable).

Remarks

DmgBrsUpdateDiary modifies the Properties of a Diary. All Properties of the Diary are returned on the Output Queue whether they were changed or not and whether they are editable or not.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
See also	DmgBrsCreateDiary DmgBrsQueryDiaries

DMGBRSUPDATEDOCUMENT

DmgBrsUpdateDocument modifies the Basic and Extended Properties of the checked out copy of a Document.

Usage

```
function = dmg_brw32->DmgBrsUpdateDocument
```

Attachment inputs

Variable	Description
A Current Document Specification	A checked out copy of a Document. Required. If the Specification is not for a checked out copy, the rule will fail.
DOC_...	Any Basic Document Properties to be changed. Any of the following may be set. DOC_NAME, DOC_CATEGORY, DOC_DESCRIPTION, DOC_SUBCATEGORY, DOC_STATUS, DOC_KEYWORD1, DOC_KEYWORD2, DOC_FLAG1, DOC_FLAG2, DOC_DATE, DOC_FILETYPE. Any others are ignored. Optional.
XDOC_propertyname	Any Extended Document Properties to be changed, where propertyname is the Bridge-encoded Extended Document Property Name (not DB Field Name). These must be appropriate for the Document's Category. If Category is being changed, the new Category is set first and is in effect. Any read-only Properties are ignored. Optional.

Attachment outputs

Variable	Description
ORIG_CABINET	The Cabinet of the original Document, not the checked out copy.
ORIG_DOC_VERS	The DOC_VERS of the original Document.
ORIG_DOCSPEC	The DOCSPEC of the original Document. Note that the variable names in the DOCSPEC are not prefixed with "ORIG_".
ORIG_DOC_ID	The Id Property of the original Document.
ORIG_DOC_MAJORVERSION	The MajorVersion Property of the original Document.
ORIG_DOC_MINORVERSION	The MinorVersion Property of the original Document.
ORIG_IS_WORKFLOW	Contains 1 if the original Document was in a Workflow Cabinet, else contains 0.
ORIGNAME	A URL Tail with the original Document's Name and FileType.
ORIG_FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the original Document. 1 is the first Level, that is, the Level nearest the root. Exists only if ORIG_IS_WORKFLOW contains 0.

ORIG_FLD_KEYSTRING	The KeyString Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_FLD_ID	The Id Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_PRJ_KEYSTRING	The Keystring Property of the Project of the original Document. Exists only if ORIG_IS_WORKFLOW contains 1.
IS_WORKFLOW	Contains 1 if the checked out copy is in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the checked out copy. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the checked out copy. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the checked out copy. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the checked out copy. Exists only if IS_WORKFLOW contains 1.
COPYNAME	A URL Tail with Name and FileType of the checked out copy.

Remarks DmgBrUpdateDocument modifies the Basic and Extended Properties of the checked out copy of a Document. The changes do not become a permanent part of the Document until the Document is checked in (DmgBrCheckInDocument).

If no Basic and no Extended Document Properties are provided, then DmgBrUpdateDocument does not change the Document and merely reports its outputs.

Only the user that checked out the Document may update it. Otherwise, the rule will fail.

To change the Category of a Document and set Extended Document Properties for the new Category, you must call DmgBrUpdateDocument twice, the first time to change the Category, and the second time to set the Extended Document Properties for the new Category.

Use DmgBrUpdateDocumentFile to update the contents file of a checked out copy of a Document.

Examples 1 The following rules list updates the Properties of the checked out copy of a Document and then presents the Folder or Project that contains that checked out copy.

```
[ ReqType:UPDATE_DOC]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmgb32->DmgBrCopyAttachment
function = dmgb32->DmgBrValidateSession
function = dmgb32->DmgBrUpdateDocument
function = dmgb32->DmgBrPresentFolder
function = dmgb32->DmgBrPresentProject
```

In the above, DmgBrsPresentFolder or DmgBrsPresentProject will run, depending on the value of IS_WORKFLOW. Thus, the same rules list can be used to handle Documents checked out in Workflow Projects or in ordinary Folders.

- 2 The following rules list updates the Properties of the checked out copy of a Document and then presents the checked out copy.

```
[ ReqType:UPDATE_DOC]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = ATCW32->ATCAppend2Attachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsUpdateDocument
function = dmg_brw32->DmgBrsCacheContentsFile
function = dmg_brw32->DmgBrsPresentDocument

[Attach: UPDATE_DOC]
DOC_MAJORVERSION = -1
```

The Current Document Specification that DmgBrsUpdateDocument requires provides all the elements of the Version Document Specification that DmgBrsCacheContentsFile needs except the Document version Properties. For this reason, we have used ATCAppend2Attachment to add DOC_MAJORVERSION = -1 to the attachment to tell DmgBrsCacheContentsFile to use the current version when setting up DmgBrsPresentDocument.

- 3 The following rules list updates the Properties of the checked out copy of a Document and then presents the Folder that contains the original Document, not the checked out copy. In this example, we'll assume that we are handling only Folders, not Projects.

```
[ ReqType:UPDATE_DOC]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmg_brw32->DmgBrsCopyAttachment
function = dmg_brw32->DmgBrsValidateSession
function = dmg_brw32->DmgBrsUpdateDocument
function = dmg_brw32->
DmgBruCopyAttachmentVar, runf, output, ORIG_CABINET, output, CABINET
function = dmg_brw32->
DmgBruCopyAttachmentVar, runf, output, ORIG_FLD_LEVELNUMBER, output,
FLD_LEVELNUMBER
function = dmg_brw32->
DmgBruCopyAttachmentVar, runf, output, ORIG_FLD_KEYSTRING, output, FLD_
KEYSTRING
function = dmg_brw32->
DmgBruCopyAttachmentVar, runf, output, ORIG_FLD_ID, output, FLD_ID
function = dmg_brw32->DmgBrsPresentFolder
```

In the above, notice that we have overwritten the values of CABINET, FLD_LEVELNUMBER, etc., for the checked out copy of the Document so that DmgBrsPresentFolder could run for the original Document.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options None.

Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	<p>DmgBrsPresentDocument</p> <p>DmgBrsPresentFolder</p> <p>DmgBrsPresentProject</p> <p>These will all use the outputs from DmgBrsUpdateDocument for the checked out copy of the Document unless you use DmgBruCopyAttachmentVar to alter the outputs so the original Document will be used. See examples above.</p>
See also	<p>DmgBrsCheckInDocument</p> <p>DmgBrsCheckOutDocument</p> <p>DmgBrsLogPropertySets</p> <p>DmgBrsUpdateAnnotations</p> <p>DmgBrsUpdateDocumentCM</p> <p>DmgBrsUpdateDocumentFile</p>

DMGBRSUPDATEDOCUMENTCM

DmgBrsUpdateDocumentCM modifies and/or presents the Content Management Properties of a Document.

Usage

```
function = dmg_brw32->DmgBrsUpdateDocumentCM
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to update the Content Management Properties. Required.
DOC_OBSOLETE	The new Obsolete flag for the Document. Set this to 0 to clear the flag. 1, or any other value other than 0, sets the flag. Optional. If missing, the Obsolete flag for the document is not updated.
DOC_RELEASED	The new Released flag for the Document. Set this to 0 to clear the flag. 1, or any other value other than 0, sets the flag. Optional. If missing, the Released flag for the document is not updated.
DOC_APPROVED	The new Approved flag for the Document. Set this to 0 to clear the flag. 1, or any other value other than 0, sets the flag. Optional. If missing, the Approved flag for the document is not updated.

Attachment outputs

Variable	Description
IS_WORKFLOW	Contains 1 if the Document is in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the Document. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the Document. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keysting Property of the Project of the Document. Exists only if IS_WORKFLOW contains 1.
DOC_OBSOLETE	The value of the Obsolete flag for the Document, after giving effect to any update requested. Will contain 1 or 0.
DOC_RELEASED	The value of the Released flag for the Document, after giving effect to any update requested. Will contain 1 or 0.
DOC_APPROVED	The value of the Approved flag for the Document, after giving effect to any update requested. Will contain 1 or 0.

Remarks

DmgBrsUpdateDocumentCM modifies and/or presents the Content Management Properties of a Document. If no Content Management Properties are passed in for modification, then this rule simply presents the Content Management Properties.

The Document must not be checked out. If the Document is checked out, this rule will fail. Changes to the Content Management Properties take effect immediately.

Note that the Documange Server enforces various rules about when the Content Management Properties may be set or cleared. If the values of the Content Management Properties passed to DmgBrsUpdateDocumentCM violates those Documange Server rules, DmgBrsUpdateDocumentCM will fail. See the Documange documentation for more information.

Use DmgBrsUpdateDocument to modify the other Properties of a Document.

ReqType changes	[ReqTypes] Failure if the rule fails.
INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder
See also	DmgBrsSetIndicators DmgBrsUpdateDocument

DMGBRSUPDATEDOCUMENTFILE

DmgBrsUpdateDocumentFile updates the contents file of a Document.

Usage

```
function = dmg_brw32->DmgBrsUpdateDocumentFile
```

Attachment inputs

Variable	Description
A Current Document Specification	The Document to update the contents file. This must be the checked out copy of a Document, or the rule will fail. Required.
DOC_UPLOADERR	Identical to DOC_UPLOADERR in DmgBrsCreateDocument. See DmgBrsCreateDocument. Optional.
DOC_FILEPATH	The path to the new contents file. See the remarks below for more information about this variable. Required.
DOC_DELFILE	Set this to 1 to cause the file in DOC_FILEPATH to be deleted after DmgBrsUpdateDocumentFile runs, whether it succeeds or not. Set this to 0 to prevent file deletion. Optional. Default is the INI option [CreateDocument] DeleteFile.
DOC_FILETYPE	The FileType Property of the Document. If this variable exists, DmgBrsUpdateDocumentFile will set the Document's FileType Property to the variable's value. If it does not exist, DmgBrsUpdateDocumentFile will set the Document's FileType Property to the extension on DOC_FILEPATH. If there is no extension on DOC_FILEPATH, DmgBrsUpdateDocumentFile will leave the Document's FileType Property unchanged. Optional.
DOC_NAME	The Name Property of the Document. If this variable exists, DmgBrsUpdateDocumentFile will set the Document's Name Property to the variable's value. Otherwise, the Name property will remain as it was before the file was updated. Optional.

Attachment outputs

Variable	Description
ORIG_CABINET	The Cabinet of the original Document, not the checked out copy.
ORIG_DOC_VERS	The DOC_VERS of the original Document.
ORIG_DOCSPEC	The DOCSPEC of the original Document. Note that the variable names in the DOCSPEC are not prefixed with "ORIG_".
ORIG_DOC_ID	The Id Property of the original Document.
ORIG_DOC_MAJORVERSION	The MajorVersion Property of the original Document.
ORIG_DOC_MINORVERSION	The MinorVersion Property of the original Document.
ORIG_IS_WORKFLOW	Contains 1 if the original Document was in a Workflow Cabinet, else contains 0.

ORIGNAME	A URL Tail with the original Document's Name and FileType.
ORIG_FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the original Document. 1 is the first Level, that is, the Level nearest the root. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_FLD_KEYSTRING	The KeyString Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_FLD_ID	The Id Property of the Folder of the original Document. Exists only if ORIG_IS_WORKFLOW contains 0.
ORIG_PRJ_KEYSTRING	The Keystring Property of the Project of the original Document. Exists only if ORIG_IS_WORKFLOW contains 1.
IS_WORKFLOW	Contains 1 if the checked out copy is in a Workflow Cabinet, else contains 0.
FLD_LEVELNUMBER	The LevelNumber Property of the Folder of the checked out copy. 1 is the first Level, that is, the Level nearest the root. Exists only if IS_WORKFLOW contains 0.
FLD_KEYSTRING	The KeyString Property of the Folder of the checked out copy. Exists only if IS_WORKFLOW contains 0.
FLD_ID	The Id Property of the Folder of the checked out copy. Exists only if IS_WORKFLOW contains 0.
PRJ_KEYSTRING	The Keystring Property of the Project of the checked out copy. Exists only if IS_WORKFLOW contains 1.
COPYNAME	A URL Tail with Name and FileType of the checked out copy.

Remarks DmgBrsUpdateDocumentFile updates the contents file of a Document. The contents file of the Document is completely replaced by the file in DOC_FILEPATH.

DOC_FILEPATH may be a local directory path, beginning with the drive, or a relative path, although relative paths are not recommended due to their ambiguity. The path must, as always, be in a format readable by the operating system on which this rule is running. Alternatively, this may be a UNC path. In any case, the path must end with a filename and must include the correct extension of the file if you do not supply DOC_FILETYPE. To avoid name collisions while uploading to some directory on a busy system, you may want to give the file a temporary name. Do not change the extension unless you supply DOC_FILETYPE.

Be sure that the FileType Property of the Document, which can be set with DOC_FILETYPE, matches the type of the data in the file. If it does not, the Bridge will not be able to correctly determine the file's MIME type when sending it to a browser, and some operating systems will have difficulty opening the file with the correct application.

Examples The output attachment variables for DmgBrsUpdateDocumentFile are the same as those for DmgBrsUpdateDocument. See the examples for DmgBrsUpdateDocument.

ReqType changes [ReqTypes] Failure if the rule fails.

INI options	[CreateDocument] DeleteFile
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentDocument DmgBrsPresentFolder DmgBrsPresentProject These will all use the outputs from DmgBrsUpdateDocumentFile for the checked out copy of the Document unless you use DmgBruCopyAttachmentVar to alter the outputs so the original Document will be used. See examples for DmgBrsUpdateDocument.
See also	DmgBrsCheckOutDocument DmgBrsUpdateAnnotations DmgBrsUpdateDocument DmgBrsValidateSession, particularly the parameter

DMGBRSUPDATEFOLDER

DmgBrsUpdateFolder modifies the Extended Properties of a Folder.

Usage

```
function = dmgb32->DmgBrsUpdateFolder
```

Attachment inputs

Variable	Description
A Folder Specification	The Folder to modify. Required.
XFLD_propertyname	The Extended Properties to be modified, where propertyname is the Bridge-encoded Folder Property Name. None, any or all may be passed in. Non-editable Properties will be ignored. As always, date values must be in a format the Bridge can recognize, or the rule will fail. Optional.

Attachment outputs

Variable	Description
FLD_LEVELNUMBER	The one-based Cabinet Level of the Folder.
FLD_ID	The ID Property of the Folder. May be zero.
FLD_KEYSTRING	The Keysting Property of the Folder. If keys were changed, this will over-write the value of the FLD_KEYSTRING that was input, if any, with a new value.

Remarks

DmgBrsUpdateFolder modifies the Extended Properties of a Folder. Any changes take effect immediately.

If no editable Properties are supplied, the Folder will not be updated.

Examples

Suppose we have a Cabinet called Customers with one Level. We want to correct the First Name Property of a particular customer, and then present the updated Folder.

The input attachment variables might look like the following.

```
CABINET = Customers
FLD_KEYSTRING = DB.dbo.Table.CustID = 123
XFLD_First$20Name = Sylvia
```

The rules list might look like the following.

```
[ReqType:UPDATE_FLD]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dmgb32->DmgBrsCopyAttachment
function = dmgb32->DmgBrsValidateSession
function = dmgb32->DmgBrsUpdateFolder
function = dmgb32->DmgBrsPresentFolder
```

DmgBrsPresentFolder uses the same Folder Specification, made up of CABINET and FLD_KEYSTRING, that applied to DmgBrsUpdateFolder.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options	None.
Validations	Session and Reqtype
Platform restrictions	None.
May be followed by	DmgBrsPresentFolder
See also	DmgBrsCreateFolder DmgBrsLogPropertySets DmgBrsPresentFolder DmgBrsUpdateDocument DmgBrsUpdateProject

DMGBRSUPDATEPROJECT

DmgBrsUpdateProject modifies the Extended and certain Basic Properties of a Project.

Usage

```
function = dmg_brw32->DmgBrsUpdateProject
```

Attachment inputs

Variable	Description
A Project Specification	The Project to modify. Required.
XPRJ_propertyname	The Extended Properties to be modified, where propertyname is the Bridge-encoded Project Property Name. None, any or all may be passed in. Non-editable Properties will be ignored. As always, date values must be in a format the Bridge can recognize, or the rule will fail. Do not set the Standard Extended Project Properties whose names begin with "I_" or "P_". Optional.
PRJ_DESCRIPTION	The Description of the Project. The value passed in here will be truncated to 100 characters. Optional.
PRJ_DUEDATE	The DueDate of the Project. Optional.
PRJ_PRIORITY	The Priority of the Project. This must be a number from 1 through 10 inclusive or the rule will fail. Optional.

Attachment outputs

Variable	Description
PRJ_KEYSTRING	The Keystring Property of the Project. If keys were changed, this will over-write the value of the PRJ_KEYSTRING that was input with a new value.

Remarks

DmgBrsUpdateProject modifies the Extended and certain Basic Properties of a Project. Any changes take effect immediately.

If no editable Properties are supplied, the Project will not be updated.

Examples

See the example under DmgBrsUpdateFolder. The same principles apply, substituting DmgBrsUpdateProject for DmgBrsUpdateFolder, and DmgBrsPresentProject for DmgBrsPresentFolder.

ReqType changes

[ReqTypes] Failure if the rule fails.

INI options

None.

Validations

Session and Reqtype

Platform restrictions

None.

May be followed by

DmgBrsPresentProject

See also

DmgBrsCreateProject
DmgBrsGetTaskQueue
DmgBrsLogPropertySets

DmgBrsPresentProject

DmgBrsUpdateDocument

DmgBrsUpdateFolder

DMGBRSVALIDATESSESSION

DmgBrsValidateSession validates the user's session and manages Bridge cookie data and UI CONFIG.

Usage

```
function = dmgb_brw32-  
>DmgBrsValidateSession,UserSessionTimeoutInMinutes
```

Parameters

Parameter	Description
UserSessionTimeoutInMinutes	Used to over-ride the INI option [ValidateSession] UserSessionTimeoutInMinutes. A value of 0 means no timeout. A positive number indicates a timeout in minutes. Optional. If the parameter does not exist or cannot be interpreted as 0 or a positive number, then the INI option is used. See discussion under remarks below.

NOTE: DmgBrsValidateSession does work on its forward and reverse runs, and the input and output attachment variables for those runs are described separately below.

Attachment inputs, forward run

Variable	Description
COOKIE.DMGBRSESS	The Bridge browser cookie. Note the dot in the name. This variable is ignored when the INI option [ValidateSession] UseCookies is OFF, and required when it is ON. Web servers will typically generate this as an environment variable if a cookie named DMGBRSESS exists.
CONFIG	The UIConfig. Pass this in to update the UIConfig to a new value. This will be passed out in SESS_CONFIG. Optional.
SESS_COOKIE_RESET	Pass this variable in (value is ignored) to cause the cookie data to be reset to default values, including [ValidateSession] DefaultUIConfig. If the INI option [UserCredentials] Source is Login, the user will be required to login again before proceeding. Because of this, SESS_COOKIE_RESET is generally passed in only during a login transaction when [UserCredentials] Source is Login. Optional.
SESS_USERDATA	Pass this in to cause the User Data in the cookie to be replaced with this new value. UserData may be up to 16 k bytes. Optional.
Trusted Credentials Variables	If the INI option [UserCredentials] Source is Trusted, then the variables configured in [UserCredentials] for supplying the trusted name and domain must be present. These variables are ignored if Source is not Trusted.
REMOTE_ADDR	A standard web server environment variable containing the IP address of the originator of the transaction. Required when Source is Login, otherwise optional.

HTTPS	Web servers frequently include this as an environment variable. Contains “on” or “off” to indicate if secure HTTP protocol is in use for this transaction. The protocol is needed when DmgBrsValidateSession builds BRIDGEURL from attachment variables because the INI options [BridgeURL] SecuredServerURL and UnsecuredServerURL haven’t been configured. If this variable is not present when the protocol is needed, DmgBrsValidateSession uses the INI option [BridgeURL] ProtocolSpecifier instead. Optional.
SERVER_NAME	A standard web server environment variable. Used when DmgBrsValidateSession builds BRIDGEURL from attachment variables because the INI options [BridgeURL] SecuredServerURL and UnsecuredServerURL haven’t been configured. Required in this circumstance, else ignored.
SERVER_PORT	A standard web server environment variable. Used when DmgBrsValidateSession builds BRIDGEURL from attachment variables because the INI options [BridgeURL] SecuredServerURL and UnsecuredServerURL haven’t been configured. Required in this circumstance if the INI option [BridgeURL] SuppressPortInServerURL is OFF, else ignored.
SCRIPT_NAME	A standard web server environment variable. Used when DmgBrsValidateSession builds BRIDGEURL from attachment variables because the INI options [BridgeURL] SecuredServerURL and UnsecuredServerURL haven’t been configured. Required in this circumstance, else ignored.
QUERY_STRING	A standard web server environment variable. Required when the INI option [UserCredentials] Source is Login. It is used in building an ad hoc login, described under remarks.

Attachment outputs

Forward run

UserData: There are two ways to update userdata.

Variable	Description
BRS_SESSION	The Bridge session handle, suitable for use in calls to dmg_api functions that require hSession. It is exposed for use by custom rules. Not present if the Bridge was not initialized, or if there was an error. Appropriate impersonation for this transaction (effective user, effective groups) will have been applied to the session. If a custom rule sets the EffectiveUsername Property or calls the dmg_api function DmgSessionSetEffectiveGroups, then it must also restore them to the state they were in before it set them. Do not call the dmg_api function DmgHandleClose on this value.

BRIDGEURL	<p>The URL for accessing the Bridge, for use in building links to the Bridge in web applications. The INI options [BridgeURL] SecuredServerURL and UnsecuredServerURL are used to create BRIDGEURL, if they exist and depending on the setting of SecureServerUsage. Otherwise, the input attachments HTTPS, SERVER_NAME, SERVER_PORT, and SCRIPT_NAME are used.</p> <p>BRIDGEURL contains the protocol, server, port (if [BridgeURL] SuppressPortInServerURL is OFF), and script (client application). Example: https://10.20.30.20:23/doc-prog/dcltw32.exe</p>
SESS_VALID	Contains 1 if DmgBrsValidateSession determines that the session is valid, else contains 0. Subsequent rules use this variable to validate the session. See remarks below for more information.
SESS_SOURCE	Will contain "Login", "Trusted", or "Bridge", depending on the setting of the INI option [UserCredentials] Source
SESS_LOGINTIME	The numeric time when the user whose cookie this is logged in with DmgBrsLogin, if SESS_SOURCE is Login, or the time at which the cookie's data was set, or the current time if cookies are not in use.
SESS_ROUTERIP	The IP address of the Documange Router the user whose cookie this is logged into with DmgBrsLogin, if SESS_SOURCE is Login, or the Router at the time the cookie's data was set, or the current Router if cookies are not in use.
SESS_USERNAME	The user name for this transaction.
SESS_USERDOMAIN	The user domain for this transaction.
SESS_GROUPACCESS	<p>The user's groups for this transaction. The existence of this variable is controlled by the option [ValidateSession]ShowGroupAccess, whose default is OFF. This variable will be empty if SESS_USERNAME is empty or contains '~~Invalid'</p>
SESS_REMOTEADDR	The value of REMOTE_ADDR for the user whose cookie data this is. This may be 0.0.0.0 if [UserCredentials] Source is not Login, since the input variable REMOTE_ADDR is not required in that case.
SESS_ACESSTIME	The time at which the user whose cookie this is last accessed the Bridge, or the current time if cookies are off.
SESS_CONFIG	The UI CONFIG found in the cookie, or the value of CONFIG, if CONFIG was passed in, or the value of [ValidateSession] DefaultUIConfig if SESS_COOKIE_RESET was passed in or if the cookie data was invalidated or is new.
SESS_USERDATA	Contains the user data from the cookie, if any. There are no Documange Bridge rules that use this data. It's exclusively for the use of the Bridge application. If SESS_USERDATA was passed in, its value will be passed out in this variable.

BRS_DESTINATION	Exists only if there is an ad hoc login, explained in Remarks.
BRSLOGIN_ERRORPROMPT	Exists only if there is an ad hoc login. Will be a number from 1 through 5 indicating the reason for the ad hoc login: Invalid session, session timed out, session logged out, invalid credentials, or a generic "Authentication failed".
BRSLOGIN_ERRORCODE	Exists only if there is an ad hoc login. A Documange error code related to the reason for the ad hoc login.
BRSLOGIN_ERRORMSG	Exists only if there is an ad hoc login. A standard message that goes with the value of BRSLOGIN_ERRORPROMPT.

Attachment inputs

Reverse run

Variable	Description
COOKIE_DATE_FORMAT	May contain either "HTTP" or "ASP" to over-ride the INI option [ValidateSession] CookieDateFormat for this transaction. Optional.
TEMP_CONFIG	Over-rides the value of SESS_CONFIG for this transaction only. See the output variable CONFIG below. Optional.
SESS_SOURCE	This, and all of the following SESS_... variables, are the outputs of the forward run. They provide the data that will be written to the cookie if the INI option [ValidateSession] UseCookies is ON. All are required when that option is ON, and ignored otherwise, except for SESS_CONFIG.
SESS_LOGINTIME	As output by the forward run.
SESS_ROUTERIP	As output by the forward run.
SESS_USERNAME	As output by the forward run.
SESS_USERDOMAIN	As output by the forward run.
SESS_GROUPACCESS	As output by the forward run.
SESS_REMOTEADDR	As output by the forward run.
SESS_ACESSTIME	As output by the forward run.
SESS_CONFIG	As output by the forward run.
SESS_USERDATA	As output by the forward run.

Attachment outputs

Reverse run

Variable	Description
HTTP_COOKIE_NAME	The name of the Bridge cookie, for use by the client application in setting the cookie in the HTTP header. Exists only if the INI option [ValidateSession] UseCookies is ON.

HTTP_COOKIETEXT	The text of the Bridge cookie, for use by the client application in setting the content of the cookie. Scrambled if the INI option [ValidateSession] ScrambleCookies is ON. Exists only if the INI option [ValidateSession] UseCookies is ON.
HTTP_COOKIEPERSIST	Contains 1 if the INI option [ValidateSession] CookiesPersist is ON, else contains 0. For use by the client application in determining if the cookie should have an expiration date. Cookies without an expiration date expire when the browser is closed. This has the effect of logging a user out if the cookie is being used for security. Exists only if the INI option [ValidateSession] UseCookies is ON.
HTTP_COOKIE_DATE	The expiration date of the Bridge cookie, for use by the client application in setting the expiration date of the cookie. The ASP cookie collection takes a different date format than the actual HTTP cookie, so COOKIE_DATE_FORMAT and its associated INI option may be used to set this date to the format required for your Bridge application. Exists only if the INI options [ValidateSession] UseCookies and [ValidateSession] CookiesPersist are both ON.
CONFIG	The UI Config. Exists only if the INI option [ValidateSession] DefaultUIConfig is not empty. This will be the value of TEMP_CONFIG if it exists, else the value of SESS_CONFIG. If SESS_CONFIG can't be found, the value of the INI option [ValidateSession] DefaultUIConfig is used.

Remarks

DmgBrsValidateSession validates the user's session and manages Bridge cookie data and UI CONFIG.

DmgBrsValidateSession must be on the rules list for every Server rule that validates Session, or you must provide other means to validate the session. See the discussion of SESS_VALID below.

The Bridge cookie maintains user-specific data between Bridge transactions. This is particularly important when the INI option [UserCredentials] Source is Login. In that case, the Bridge will not initialize successfully if [ValidateSession] UseCookies is OFF. This is because the user's name and domain are stored in the cookie. They are an important part of validating the cookie, and provide the basis for correct user impersonation when [UserCredentials] Source is Login. The Bridge cookie also maintains the UI CONFIG and any UserData between sessions. When the INI option [UserCredentials] Source is not Login, [ValidateSession] UseCookies may be set to OFF, but in that case, the UI CONFIG and any UserData will not be preserved between sessions unless the Bridge application takes care of them.

There are two ways to update userdata. The variable SESS_USERDATA may be set prior to the forward run of DmgBrsValidateSession. This allows the client application to update the data. Alternatively, SESS_USERDATA may be set after the forward but prior to the reverse run of DmgBrsValidateSession. This allows a custom server rule to set it. A typical place to initialize userdata is during the transaction that runs the rule DmgBrsLogin, if Login is in use.

FORWARD RUN

On its forward run, DmgBrsValidateSession reads the Bridge cookie, if any, unscrambles it if the INI option [ValidateSession] ScrambleCookies is ON, validates the session, and outputs session data for use by other rules and the reverse run of DmgBrsValidateSession.

The forward run fails if the Bridge is not initialized.

The forward run checks for minimum disk space. The required minimum is about 2 megabytes. It fails if disk space is inadequate.

If the INI option [UserCredentials] Source is set to Bridge, the forward run will always set SESS_VALID to 1. It will set the SESS_... variables to appropriate values for the Bridge. If the INI option [ValidateSession] UseCookies is ON, the forward run preserves the UIConfig and UserData from the cookie in the variables SESS_CONFIG and SESS_USERDATA, unless these are over-ridden in the ways described above for the input attachments.

If the INI option [UserCredentials] Source is set to Trusted, the forward run will attempt to find the trusted name variables described in [UserCredentials]. If it can't find them or if they are empty, it will set SESS_VALID to 0. It looks first for TrustedNameVar. If that's missing or empty, it looks for TrustedQualifiedNameVar. If it can find the proper variables, it will set SESS_VALID to 1 and set SESS_USERNAME and SESS_USERDOMAIN to the trusted values. Like Source Bridge, if UseCookies is ON, it always preserves the UIConfig and UserData from the cookie.

If the INI option Source in [UserCredentials] is set to Login, the forward run will validate the cookie. Validations are:

- Check the username is not ~~LOGGEDOUT (which means the session is logged out) or ~~INVALID (which means the cookie couldn't be read or was reset).
- Check that the source of credentials currently in effect matches that in the cookie.
- If the INI option [ValidateSession] UserSessionTimeoutInMinutes is greater than 0, check that the session hasn't timed out. See description of the parameter for important details.
- If the INI option [ValidateSession] LogoutUsersOnBridgeRestart is ON, checks that the session is younger than the last Bridge restart.
- Check that the last access time is not prior to login time, and that neither time is zero.
- Check that the cookie does not appear to be stolen. It does this by comparing the SESS_REMOTEADDR, which is read from the cookie, to the input variable REMOTE_ADDR. They must match.

If the cookie data fails any of the above validations, the rule sets SESS_USERNAME to ~~LOGGEDOUT (if it isn't already ~~INVALID). It then sets up an "ad hoc" login.

In ad hoc login is the process of asking a user to log in again, and then directing them back to the rules list where the request to log in again originated.

DmgBrsValidateSession does the following to set up an ad hoc login:

- 1 It sets the output variable SESS_VALID to 0 so that subsequent Bridge rules that depend on a valid session will not run.
- 2 It changes the reqtype to the reqtype configured in the INI option [ReqTypes] LoginUIDisplay. The Bridge application must detect this condition when it handles the result type from this transaction, and should present the user with a web page that solicits the user's credentials, a "login UI display", for the LoginUIDisplay ResType. If [BridgeURL] SecureServerUsage is Login, it will change the reqtype to [ReqTypes] Redirect instead of LoginUIDisplay, and redirect the transaction to the secure server if necessary. Again, the Bridge application must detect this condition when it handles the result type from this transaction and process the Redirect ResType correctly.
- 3 It outputs the variables BRSLOGIN_ERRORPROMPT, BRSLOGIN_ERRORCODE, and BRSLOGIN_ERRORMSG for use in the login UI display web page.

- 4 It creates the variable BRS_DESTINATION, and sets its value to a URL that brings the user back to the current transaction after logging in. The login UI display web page must pass BRS_DESTINATION to the transaction that runs DmgBrsLogin. See DmgBrsLogin. To build the URL in BRS_DESTINATION, DmgBrsValidateSession will use BRIDGEURL and the input attachment variable QUERY_STRING. For an ad hoc login to work, be certain that QUERY_STRING is supplied, and that the INI options [ReqTypes] LoginUIDisplay, LoginExecute, and Redirect and the options in the [BridgeURL] section are configured properly.

SESS_VALID—All Bridge server rules that validate the session do so by checking the value of SESS_VALID. If it is not 1, the session is invalid and a rule that checks the session will not do anything. If you choose not to use DmgBrsValidateSession to validate the session and instead provide your own custom rule to do so, then you must set SESS_VALID to the correct value. You should also set SESS_USERNAME, since it is used by some other rules.

NOTE: Any of the outputs of the forward run of DmgBrsValidateSession may be useful when writing custom rules. In particular, BRS_SESSION may be useful. Knowledge of the specific format of the cookie should not be needed by Bridge rules since the contents of the cookie are exposed through the SESS_... variables. The format of the cookie is not guaranteed to remain the same and should not be relied upon.

REVERSE RUN

On its reverse run, DmgBrsValidateSession writes the Bridge cookie data (HTTP_COOKIE...), if the INI option [ValidateSession] UseCookies is on, and sets CONFIG, if the INI option [ValidateSession] DefaultUIConfig is not empty.

UI Config—There are three ways to update the UI Config. Passing in SESS_COOKIE_RESET before the forward run of DmgBrsValidateSession causes SESS_CONFIG to be set to the default value. Passing in CONFIG before the forward run of DmgBrsValidateSession causes SESS_CONFIG to be updated with the value of CONFIG, even if SESS_COOKIE_RESET was also passed in. And finally, a custom rule may set SESS_CONFIG after the forward run of DmgBrsValidateSession but prior to the reverse run of DmgBrsValidateSession. The reverse run sets the output variable CONFIG to the value of SESS_CONFIG and, if [ValidateSession] UseCookies is ON, puts the value of SESS_CONFIG into the cookie text.

USE OF THE PARAMETER

The optional parameter for DmgBrsValidateSession may be needed on some rules lists if Login is the source of the user credentials. The parameter exists in order to help Bridge application designers, especially those using multiple IDS servers, handle the possibility that a user session may time out during the process of uploading a file if the upload time exceeds the value of the INI option [ValidateSession] UserSessionTimeoutInMinutes. This INI option applies only when Login is the source of the user credentials.

If a session timed out during a long file upload, then DmgBrsValidateSession would discover the timeout on the server after the upload. The file would remain wherever the client application put it while the ad hoc login and redirection back to the transaction occurred. This might work out just fine on a single-IDS system, but on a multi-IDS system, the redirection could come back to a different server that didn't have the uploaded file on it, in which case an error would result because the path to the file might be wrong. For example, on the rules list for DmgBrsCreateDocument, which requires a file to be uploaded, you might set the parameter to 0 to prevent an ad hoc login due to a timeout.

ReqType changes	<p>[ReqTypes] Failure if the rule fails.</p> <p>[ReqTypes] LoginUIDisplay or Redirect if there is an ad hoc login.</p>
INI options	<p>[ValidateSession]</p> <p>[UserCredentials]</p> <p>[BridgeURL]</p> <p>[ReqTypes] Redirect</p> <p>[ReqTypes] LoginUIDisplay</p>
Validations	None.
Platform restrictions	None.
See also	DmgBrsLogin

DMGBRUCOPYATTACHMENTVAR

DmgBruCopyAttachmentVar copies the value of any attachment variable to another attachment variable.

Usage

```
function = dmg_brw32-  
>DmgBruCopyAttachmentVar,run,srcQ,srcVar,dstQ,dstVar
```

Parameters

Parameter	Description
Run	The run of the rules list. May be RUNF for the forward run or RUNR for the reverse run, case insensitive. Required. Interpreted as RUNF if not recognized.
srcQ	The queue for srcVar. May be INPUT or OUTPUT, case insensitive. Required. Interpreted as OUTPUT if not recognized.
srcVar	The source variable. May be an attachment variable name or a RecordSet Wildcard. Required.
dstQ	The queue for dstVar. May be INPUT or OUTPUT, case insensitive. Required. Interpreted as OUTPUT if not recognized.
dstVar	The name of the destination variable. This variable is created if it doesn't exist. If srcVar is a RecordSet Wildcard, then dstVar will be interpreted to mean the name of a field in that RecordSet. Required.

Attachment inputs

Variable	Description
srcVar	The variable or variables designated in the parameter srcVar.

Attachment outputs

Variable	Description
dstVar	The variable or variables designated in the parameter dstVar.

Remarks

DmgBruCopyAttachmentVar copies the value of any attachment variable to another attachment variable. The destination variable will be created if it doesn't exist, and will be over-written if it does.

This rule is controlled through its parameters, described above. It does nothing except to log the problem if it can't find the parameter srcVar, or if the parameters are incorrect.

Examples

DmgBruCopyAttachmentVar can be used in many situations, and the examples under many of the other rules illustrate its use.

- 1 On the forward run, copy the output variable named "MYVAR" to the output variable named "MYCOPY".

```
dmg_brw32->DmgBruCopyAttachmentVar,RUNF,OUTPUT,MyVar,OUTPUT,MyCopy
```
- 2 On the reverse run, copy all of the "Property" fields in the record set "Hits" to variables called "Property_URL". All variables are on the output attachment.

```
dmg_brw32-
>DmgBruCopyAttachmentVar,RunR,Output,Hits*.Property,OutPut,Property
_URL
```

Suppose there were 2 records in the record set, and that Property was the only field in the records. Here's what the output attachment might have looked like before the rule ran:

```
HITS = 2
HITS1.PROPERTY = value 1
HITS2.PROPERTY = value 2
```

After the rule ran, the record set would look like this:

```
HITS = 2
HITS1.PROPERTY = value 1
HITS1.PROPERTY_URL = value 1
HITS2.PROPERTY = value 2
HITS2.PROPERTY_URL = value 2
```

In the above example, DmgBrsEncodeURLField could be run after DmgBruCopyAttachmentVar to URL encode the values of HITS*.PROPERTY_URL.

Platform restrictions None.

See also DmgBrsCopyAttachment
DmgBrsEncodeURLField

DMGBRULOGATTACHMENTS

DmgBruLogAttachments logs the values of all the variables on the designated attachment.

Usage `function = dmg_brw32->DmgBruLogAttachments,run=queue[,...]`

Parameters

Parameter	Description
run=queue	“run” is the run of the rules list and may be the values RUNF for the forward run or RUNR for the reverse run, case insensitive. “queue” is the attachment queue to log and may be INPUT, OUTPUT, or BOTH. There may be as many parameters as needed, or no parameters.

Attachment inputs None.

Attachment outputs None.

Remarks DmgBruLogAttachments logs the values of all the variables on the designated attachment. It is used as a diagnostic aid, to show what the state of either attachment is at any point in the forward or reverse run of a rules list on the server or the client.

If there are no parameters, DmgBruLogAttachments logs the input attachment on the forward run and the output attachment on the reverse run.

Whenever DmgBrsCopyAttachment appears on a server rules list, it logs the state of the output attachment at the beginning and end of the run of the rules list. DmgBruLogAttachments is more flexible, and gives more precise diagnostics, rule by rule if need be.

- Examples**
- 1 The following executes the default behavior, which logs the input attachment on the forward run and the output attachment on the reverse run.
`function = dmg_brw32->DmgBruLogAttachments`
 - 2 The following logs both attachments on each run.
`function = dmg_brw32->DmgBruLogAttachments,RunF=Both,RunR=Both`
 - 3 The following logs the output attachment on the forward run and the input attachment on the reverse run.
`function = dmg_brw32->DmgBruLogAttachments,RUNF=OUTPUT,RUNR=INPUT`
 - 4 The following logs the input attachment on the forward run.
`function = dmg_brw32->DmgBruLogAttachments,RUNF=INPUT`

Platform restrictions None.

See also DmgBrsCopyAttachment

DMGBRUPURGEFILES

DmgBruPurgeFiles deletes files from a directory under the control of various options.

Usage

```
function = dmg_brw32-
>DmgBruPurgeFiles,directory,age_in_minutes,trigger
```

Parameters

Parameter	Description
directory	The directory from which files will be deleted, with or without a trailing directory delimiter. Required.
age_in_minutes	The age in minutes of files to purge, as measured by file creation time. Set this to 0 to purge all files. Optional. If absent, files whose creation time is more than 10 minutes ago will be purged.
trigger	The name (case-independent) of an attachment variable. If the variable is present on either the input or the output queue, the rule will purge files. If the variable is absent, the rule will not purge files. Optional. Default is to purge files.

Attachment inputs

Variable	Description
trigger	The attachment variable configured in the trigger parameter, if any. May be on either the input or the output queue.

Attachment outputs

None.

Remarks

DmgBruPurgeFiles deletes files from a directory under the control of various options specified in the parameters, described above. This rule does its file deletion on the reverse run of the rules list.

This can be particularly useful on the client. See DmgBrcReceiveTempFile.

Do not use this rule to purge files from the directory configured in the INI option [FileCache] DirPath or the directory configured in [TempCache] DirPath. Those are automatically purged by DmgBrsIdle at the proper time.

Examples

See also the example for DmgBrcReceiveTempFile.

- 1 Purge files in e:\temp whose creation time is more than 10 minutes ago.

```
function = dmg_brw32->DmgBruPurgeFiles,e:\temp\
```

Since 10 is the default for the second parameter, it doesn't need to be specified.
- 2 Purge files in e:\temp whose creation time is more than 45 minutes ago.

```
function = dmg_brw32->DmgBruPurgeFiles,e:\temp,45
```
- 3 Purge files in e:\temp whose creation time is more than 20 minutes ago, but only if the attachment variable "MyVar" is present.

```
function = dmg_brw32->DmgBruPurgeFiles,e:\temp,20,MyVar
```
- 4 Purge files in e:\temp, but only if the attachment variable "MyVar" is present. The age of the files is not specified, so it will be the default of 10 minutes.

```
function = dmg_brw32->DmgBruPurgeFiles,e:\temp,,MyVar
```

Platform restrictions None.

See also DmgBrcReceiveTempfile
 DmgBrsIdle

Chapter 8

Quick Reference by Alphabet

The following table contains all of the rules in alphabetical order, with a brief description of each.

Rule	Description
DmgBrcDumpFile	DmgBrcDumpFile writes a file into an HTTP response for a web server to return to a browser.
DmgBrcReceiveTempfile	DmgBrcReceiveTempfile receives Bridge Temp Files sent over the queue from the Bridge Server with ATCSendFile.
DmgBrsAddDocumentAsRendition	DmgBrsAddDocumentAsRendition creates a Rendition relationship between two Document Versions.
DmgBrsAddNote	DmgBrsAddNote adds a Note to a Documanager Folder or Project.
DmgBrsAdvanceProject	DmgBrsAdvanceProject checks in a checked out Project and advances it.
DmgBrsCacheAnnotationsFile	DmgBrsCacheAnnotationsFile puts a Document annotations file into the Bridge Cache for use by other rules.
DmgBrsCacheContentsFile	DmgBrsCacheContentsFile puts a Document contents file and any Subdocument files into the Bridge Cache for use by other rules.
DmgBrsCheckInDocument	DmgBrsCheckInDocument checks in the checked out copy of a Document, committing any changes to permanent storage.
DmgBrsCheckInProject	DmgBrsCheckInProject checks in a checked out Project without advancing it.
DmgBrsCheckOutDocument	DmgBrsCheckOutDocument checks out a Document.
DmgBrsCheckOutProject	DmgBrsCheckOutProject checks out a Project.
DmgBrsCheckPausedServer	DmgBrsCheckPausedServer runs on the PAUSE rules list to monitor Documanager and resume IDS when appropriate.
DmgBrsCopyAttachment	DmgBrsCopyAttachment copies all of the input attachment variables to the output attachment.
DmgBrsCopyDocument	DmgBrsCopyDocument makes a copy of a Document in any specified Folder or Project.
DmgBrsCreateDiary	DmgBrsCreateDiary creates a new Diary.

DmgBrsCreateDocument	DmgBrsCreateDocument creates a new Document in a Folder or Project.
DmgBrsCreateDocumentShortcut	DmgBrsCreateDocumentShortcut creates a Shortcut to a Document.
DmgBrsCreateFolder	DmgBrsCreateFolder creates up to one new Folder at each Level of a Cabinet.
DmgBrsCreateProject	DmgBrsCreateProject creates a new Workflow Project.
DmgBrsCreateProjectFromFolder	DmgBrsCreateProjectFromFolder transforms an existing Folder into a Workflow Project.
DmgBrsDeleteDiary	DmgBrsDeleteDiary deletes a Diary.
DmgBrsDeleteDocument	DmgBrsDeleteDocument deletes all versions of a Document in a Cabinet.
DmgBrsDeleteFolder	DmgBrsDeleteFolder deletes a Folder and, optionally, all of its contents.
DmgBrsEncodeURLField	DmgBrsEncodeURLField URL encodes the values of designated attachment variables.
DmgBrsForwardProject	DmgBrsForwardProject forwards a Project to another member of the Project Team.
DmgBrsGetTaskQueue	DmgBrsGetTaskQueue returns the list of Projects waiting at a particular Task for the current user.
DmgBrsGetTasks	DmgBrsGetTasks returns the list of tasks for the logged-in user.
DmgBrsIdle	DmgBrsIdle runs on the SAR rules list and performs periodic Bridge housekeeping tasks.
DmgBrsInit	DmgBrsInit runs on the INI rules list and initializes and shuts down the Bridge to Documange.
DmgBrsListCabinets	DmgBrsListCabinets lists the Cabinets available on the Documange System.
DmgBrsListCategories	DmgBrsListCategories lists the Categories available on the Documange System.
DmgBrsListGroup	DmgBrsListGroup lists the Groups on the Documange system, or the Groups for a particular User on the Documange system.

DmgBrsListUsers	DmgBrsListUsers lists the Users on the Documanager system, or the Users for a particular Group on the Documanager system.
DmgBrsListWorkflows	DmgBrsListWorkflows lists the Workflow Cabinets available on the Documanager System.
DmgBrsLockAnnotations	DmgBrsLockAnnotations locks the Annotations of a Document to the current user so they can be updated.
DmgBrsLogin	DmgBrsLogin logs in an individual user.
DmgBrsLogout	DmgBrsLogout logs out an individual user.
DmgBrsLogPropertySets	DmgBrsLogPropertySets logs Property Sets as an aid in configuring the Bridge.
DmgBrsMarkNoteCurrent	DmgBrsMarkNoteCurrent sets the Deleted Property of a Note to false.
DmgBrsMarkNoteDeleted	DmgBrsMarkNoteDeleted sets the Deleted Property of a Note to true.
DmgBrsMoveDocument	DmgBrsMoveDocument moves a Document to a different Folder or Project.
DmgBrsPresentCabinetEx	DmgBrsPresentCabinetEx presents information about a particular Cabinet or Workflow Cabinet.
DmgBrsPresentCategory	DmgBrsPresentCategory presents information about a particular Category.
DmgBrsPresentCompoundDocument	DmgBrsPresentCompoundDocument presents information about a Compound Document, including a list of the Subdocuments in it and their characteristics.
DmgBrsPresentDocument	DmgBrsPresentDocument presents information about a Document and its versions.
DmgBrsPresentDocumentPage	DmgBrsPresentDocumentPage presents information about a page of a Document.
DmgBrsPresentFolder	DmgBrsPresentFolder presents information about a Folder and any Notes associated with it.
DmgBrsPresentProject	DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.

DmgBrsPresentSubdocument	DmgBrsPresentSubdocument presents information about a specific Subdocument in a Compound Document.
DmgBrsPresentWorkflow	DmgBrsPresentWorkflow presents information about a Workflow Schema.
DmgBrsQuery	DmgBrsQuery searches for Folders and Documents within a Folder.
DmgBrsQueryAllProjectsEx	DmgBrsQueryAllProjectsEx searches for a user's Projects across all or many Workflow Cabinets, without regard to Task.
DmgBrsQueryDiaries	DmgBrsQueryDiaries searches for a user's Diaries and presents information about them.
DmgBrsQueryDocumentRenditions	DmgBrsQueryDocumentRenditions returns all the Renditions of a Document.
DmgBrsQueryDocuments	DmgBrsQueryDocuments searches for Documents across Folders.
DmgBrsQueryProjects	DmgBrsQueryProjects searches for a user's Projects in a Workflow Cabinet, without regard to Task.
DmgBrsRemoveDocumentAsRendition	DmgBrsRemoveDocumentAsRendition removes a Document Version from all the Rendition relationships of which it is a member.
DmgBrsResumeProject	DmgBrsResumeProject resumes a suspended Project.
DmgBrsReturnAnnotations	DmgBrsReturnAnnotations returns an annotation file containing all the annotations for a Document.
DmgBrsReturnConvertedDocument	DmgBrsReturnConvertedDocument returns a Document contents file converted to the specified type.
DmgBrsReturnDocument	DmgBrsReturnDocument returns a Document contents file.
DmgBrsReturnThumbnail	DmgBrsReturnThumbnail returns a thumbnail image of a Document page.
DmgBrsRunDocumangeDiagnostic	DmgBrsRunDocumangeDiagnostic runs a Documange Diagnostic Test and executes the designated failure action if the test detects a problem.

DmgBrsSetDocumentRetention	DmgBrsSetDocumentRetention sets the retention date of a Document.
DmgBrsSetIndicators	DmgBrsSetIndicators replaces the 1 or 0 value of an attachment variable with a designated new value.
DmgBrsSetupCompoundDocument	DmgBrsSetupCompoundDocument allows the construction of a transaction that will return either a file that can be viewed in a browser or, if that is not possible, the results of DmgBrsPresentCompoundDocument.
DmgBrsSetupFolder	DmgBrsSetupFolder finds a Folder and puts its Folder Specification on the attachment, in preparation to run other rules such as DmgBrsCreateDocument.
DmgBrsSuspendProject	DmgBrsSuspendProject suspends a Project.
DmgBrsUndoDocumentCheckOut	DmgBrsUndoDocumentCheckOut undoes a Document checkout.
DmgBrsUnlockAnnotations	DmgBrsUnlockAnnotations unlocks locked Annotations.
DmgBrsUpdateAnnotations	DmgBrsUpdateAnnotations updates the Annotations File of a Document.
DmgBrsUpdateDiary	DmgBrsUpdateDiary modifies the Properties of a Diary.
DmgBrsUpdateDocument	DmgBrsUpdateDocument modifies the Basic and Extended Properties of the checked out copy of a Document.
DmgBrsUpdateDocumentCM	DmgBrsUpdateDocumentCM modifies and/or presents the Content Management Properties of a Document.
DmgBrsUpdateDocumentFile	DmgBrsUpdateDocumentFile updates the contents file of a Document.
DmgBrsUpdateFolder	DmgBrsUpdateFolder modifies the Extended Properties of a Folder.
DmgBrsUpdateProject	DmgBrsUpdateProject modifies the Extended and certain Basic Properties of a Project.
DmgBrsValidateSession	DmgBrsValidateSession validates the user's session and manages Bridge cookie data and UI CONFIG.

DmgBruCopyAttachmentVar	DmgBruCopyAttachmentVar copies the value of any attachment variable to another attachment variable.
DmgBruLogAttachments	DmgBruLogAttachments logs the values of all the variables on the designated attachment.
DmgBruPurgeFiles	DmgBruPurgeFiles deletes files from a directory under the control of various options.

Chapter 9

Quick Reference by Documanager Object

The following tables group the rules by the Documanager object they deal with, with a brief description of each rule.

Some rules, such as DmgBrsCopyAttachment, will not appear in any of these groups because they do not deal with Documanager objects.

Some rules may appear in more than one group because they deal with more than one type of object. For example, DmgBrsPresentFolder deals with both Folders and Notes.

Documents

Rule	Description
DmgBrsAddDocumentAsRendition	DmgBrsAddDocumentAsRendition creates a Rendition relationship between two Document Versions.
DmgBrsCacheContentsFile	DmgBrsCacheContentsFile puts a Document contents file and any Subdocument files into the Bridge Cache for use by other rules.
DmgBrsCheckInDocument	DmgBrsCheckInDocument checks in the checked out copy of a Document, committing any changes to permanent storage.
DmgBrsCheckOutDocument	DmgBrsCheckOutDocument checks out a Document
DmgBrsCopyDocument	DmgBrsCopyDocument makes a copy of a Document in any specified Folder or Project.
DmgBrsCreateDocument	DmgBrsCreateDocument creates a new Document in a Folder or Project.
DmgBrsCreateDocumentShortcut	DmgBrsCreateDocumentShortcut creates a Shortcut to a Document.
DmgBrsDeleteDocument	DmgBrsCreateProject creates a new Workflow Project.
DmgBrsMoveDocument	DmgBrsMoveDocument moves a Document to a different Folder or Project.
DmgBrsPresentCompoundDocument	DmgBrsPresentCompoundDocument presents information about a Compound Document, including a list of the Subdocuments in it and their characteristics.
DmgBrsPresentDocument	DmgBrsPresentDocument presents information about a Document and its versions.
DmgBrsPresentDocumentPage	DmgBrsPresentDocumentPage presents information about a page of a Document.
DmgBrsPresentProject	DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.
DmgBrsPresentSubdocument	DmgBrsPresentSubdocument presents information about a specific Subdocument in a Compound Document.
DmgBrsQuery	DmgBrsQuery searches for Folders and Documents within a Folder.

DmgBrsQueryDocumentRenditions	DmgBrsQueryDocumentRenditions returns all the Renditions of a Document.
DmgBrsQueryDocuments	DmgBrsQueryDocuments searches for Documents across Folders.
DmgBrsRemoveDocumentAsRendition	DmgBrsRemoveDocumentAsRendition removes a Document Version from all the Rendition relationships of which it is a member.
DmgBrsReturnConvertedDocument	DmgBrsReturnConvertedDocument returns a Document contents file converted to the specified type.
DmgBrsReturnDocument	DmgBrsReturnDocument returns a Document contents file.
DmgBrsReturnDocumentPage	DmgBrsReturnDocumentPage returns a page of a Document contents file, converted to JPG, black and white TIF, image PDF, EMF, or DCZ (compressed EMF).
DmgBrsReturnThumbnail	DmgBrsReturnThumbnail returns a thumbnail image of a Document page.
DmgBrsSetDocumentRetention	DmgBrsSetDocumentRetention sets the retention date of a Document.
DmgBrsSetupCompoundDocument	DmgBrsSetupCompoundDocument allows the construction of a transaction that will return either a file that can be viewed in a browser or, if that is not possible, the results of DmgBrsPresentCompoundDocument.
DmgBrsSetupFolder	DmgBrsSetupFolder finds a Folder and puts its Folder Specification on the attachment, in preparation to run other rules such as DmgBrsCreateDocument.
DmgBrsUndoDocumentCheckOut	DmgBrsUndoDocumentCheckOut undoes a Document checkout.
DmgBrsUpdateDocument	DmgBrsUpdateDocument modifies the Basic and Extended Properties of the checked out copy of a Document.
DmgBrsUpdateDocumentCM	DmgBrsUpdateDocumentCM modifies and/or presents the Content Management Properties of a Document.
DmgBrsUpdateDocumentFile	DmgBrsUpdateDocumentFile updates the contents file of a Document.

Folders

Rule	Description
DmgBrsCreateFolder	DmgBrsCreateFolder creates up to one new Folder at each level of a Cabinet.
DmgBrsDeleteFolder	DmgBrsDeleteFolder deletes a Folder and, optionally, all of its contents.
DmgBrsPresentFolder	DmgBrsPresentFolder presents information about a Folder and any Notes associated with it.
DmgBrsQuery	DmgBrsQuery searches for Folders and Documents within a Folder.
DmgBrsQueryDocuments	DmgBrsQueryDocuments searches for Documents across Folders.
DmgBrsSetupFolder	DmgBrsSetupFolder finds a Folder and puts its Folder Specification on the attachment, in preparation to run other rules such as DmgBrsCreateDocument.
DmgBrsUpdateFolder	DmgBrsUpdateFolder modifies the Extended Properties of a Folder.

Projects

Rule	Description
DmgBrsAdvanceProject	DmgBrsAdvanceProject checks in a checked out Project and advances it.
DmgBrsCheckInProject	DmgBrsCheckInProject checks in a checked out Project without advancing it.
DmgBrsCheckOutProject	DmgBrsCheckOutDocument checks out a Document.
DmgBrsCreateProject	DmgBrsCreateProject creates a new Workflow Project.
DmgBrsCreateProjectFromFolder	DmgBrsCreateProjectFromFolder transforms an existing Folder into a Workflow Project.
DmgBrsForwardProject	DmgBrsForwardProject forwards a Project to another member of the Project Team.
DmgBrsGetTaskQueue	DmgBrsGetTaskQueue returns the list of Projects waiting at a particular Task for the current user.
DmgBrsPresentProject	DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.
DmgBrsQueryAllProjectsEx	DmgBrsQueryAllProjectsEx searches for a user's Projects across all or many Workflow Cabinets, without regard to Task.
DmgBrsQueryProjects	DmgBrsQueryProjects searches for a user's Projects in a Workflow Cabinet, without regard to Task.
DmgBrsResumeProject	DmgBrsResumeProject resumes a suspended Project.
DmgBrsSuspendProject	DmgBrsSuspendProject suspends a Project.
DmgBrsUpdateProject	DmgBrsUpdateProject modifies the Extended and certain Basic Properties of a Project.

Tasks

Rule	Description
DmgBrsGetTaskQueue	DmgBrsGetTaskQueue returns the list of Projects waiting at a particular Task for the current user.
DmgBrsGetTasks	DmgBrsGetTasks returns the list of tasks for the logged-in user.
DmgBrsPresentWorkflow	DmgBrsPresentWorkflow presents information about a Workflow Schema.

Notes

Rule	Description
DmgBrsAddNote	DmgBrsAddNote adds a Note to a Documanager Folder or Project.
DmgBrsMarkNoteCurrent	DmgBrsMarkNoteCurrent sets the Deleted Property of a Note to false.
DmgBrsMarkNoteDeleted	DmgBrsMarkNoteDeleted sets the Deleted Property of a Note to true.
DmgBrsPresentFolder	DmgBrsPresentFolder presents information about a Folder and any Notes associated with it.
DmgBrsPresentProject	DmgBrsPresentProject presents information about a Project, any Notes associated with it, and the Documents it contains.

Diaries

Rule	Description
DmgBrsCreateDiary	DmgBrsCreateDiary creates a new Diary.
DmgBrsDeleteDiary	DmgBrsDeleteDiary deletes a Diary.
DmgBrsQueryDiaries	DmgBrsQueryDiaries searches for a user's Diaries and presents information about them.
DmgBrsUpdateDiary	DmgBrsUpdateDiary modifies the Properties of a Diary.

Annotations

Rule	Description
DmgBrsCacheAnnotations	DmgBrsCacheAnnotationsFile puts a Document annotations file into the Bridge Cache for use by other rules.
DmgBrsLockAnnotations	DmgBrsLockAnnotations locks the Annotations of a Document to the current user so they can be updated.
DmgBrsReturnAnnotations	DmgBrsReturnAnnotations returns an annotation file containing all the annotations for a Document.
DmgBrsUnlockAnnotations	DmgBrsUnlockAnnotations unlocks locked Annotations.
DmgBrsUpdateAnnotations	DmgBrsUpdateAnnotations updates the Annotations File of a Document.

Documanage Diagnostic Tests

Rule	Description
DmgBrsCheckPausedServer	DmgBrsCheckPausedServer runs on the PAUSE rules list to monitor Documanage and resume IDS when appropriate.
DmgBrsIdle	DmgBrsIdle runs on the SAR rules list and performs periodic Bridge housekeeping tasks.
DmgBrsInit	DmgBrsInit runs on the INI rules list and initializes and shuts down the Bridge to Documanage.
DmgBrsRunDocumanageDiagnostic	DmgBrsRunDocumanageDiagnostic runs a Documanage Diagnostic Test and executes the designated failure action if the test detects a problem.

Session

Rule	Description
DmgBrsInit	DmgBrsInit runs on the INI rules list and initializes and shuts down the Bridge to Documange.
DmgBrsIdle	DmgBrsIdle runs on the SAR rules list and performs periodic Bridge housekeeping tasks.
DmgBrsLogin	DmgBrsLogin logs in an individual user.
DmgBrsLogout	DmgBrsLogout logs out an individual user.
DmgBrsValidateSession	DmgBrsValidateSession validates the user's session and manages Bridge cookie data and UI CONFIG.

Chapter 10

Exploration Rules

Most Documanager Bridge applications are customized for a particular Documanager system. The rules listed in the following table, however, let you get the information you need to write a Documanager Bridge application that will work with any Documanager system.

Rule	Description
DmgBrsListCabinets	DmgBrsListCabinets lists the Cabinets available on the Documanager System.
DmgBrsListCategories	DmgBrsListCategories lists the Categories available on the Documanager System.
DmgBrsListGroups	DmgBrsListGroups lists the Groups on the Documanager system, or the Groups for a particular User on the Documanager system.
DmgBrsListUsers	DmgBrsListUsers lists the Users on the Documanager system, or the Users for a particular Group on the Documanager system.
DmgBrsListWorkflows	DmgBrsListWorkflows lists the Workflow Cabinets available on the Documanager System.
DmgBrsPresentCabinetEx	DmgBrsPresentCabinetEx presents information about a particular Cabinet or Workflow Cabinet.
DmgBrsPresentCategory	DmgBrsPresentCategory presents information about a particular Category.

