opentext™

OpenText™ Documentum™ Content Services for SAP® Solutions

Administration Guide

Configure connections to SAP, HTTP archiving services, agent component, and view component using the WebAdmin tool.

EDCCOSAPCS250400-AGD-EN-01

OpenText™ Documentum™ Content Services for SAP® Solutions
Administration Guide
EDCCOSAPCS250400-AGD-EN-01
Rev.: 2025-Oct-21

This documentation has been created for OpenText™ Documentum™ Content Services for SAP® Solutions CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Introducing Documentum Content Services for SAP Solutions

1.1 Overview

The Documentum Content Services for SAP Solutions (CS SAP) application integrates the OpenText™ Documentum™ Content Management system with the SAP R/3 or ECC system. It is an interface to the SAP R/3 or ECC system based on SAP standard Document Management System (DMS) and ArchiveLink interfaces.

Documentum Content Services for SAP Solutions provides the following functions:

- Enables users to easily access and display documents stored in a OpenText Documentum Content Management (CM) repository from a variety of SAP modules.
- Links documents stored in OpenText Documentum CM to a variety of SAP documents.
- Archives SAP data, reports, and documents through ArchiveLink certified interfaces in OpenText Documentum CM.

  Note: To use Documentum Content Services for SAP Solutions with HTTP archiving services, install , as described in the *OpenText Documentum Archive Services for SAP Solutions - Installation Guide (EDCCOSAPAR250400-IGD)*.

- Stores and manages reports and outgoing documents in OpenText Documentum CM.

1.2 Intended audience

In order to address the manuals to the correct audience, the roles of the users of the manuals have been defined as follows:

- System Administrator – This role covers users who install and configure Documentum Content Services for SAP Solutions. Documentum Content Services for SAP Solutions integrates OpenText Documentum CM and the SAP R/3 system.
- Content Services Administrator – This role covers users who manage Documentum Content Services for SAP Solutions using WebAdmin.
- Document Controller – This role covers those who release documents to SAP and maintain the links to those documents.

- Standard User – This role covers users who view documents using SAPGUI and the View component of Documentum Content Services for SAP Solutions.

This document is intended for Documentum Content Services for SAP Solutions administrators.

This document forms part of a documentation suite designed to support those who install, configure, and use Documentum Content Services for SAP Solutions. The product and documentation suite can be found on OpenText My Support (https://support.opentext.com/).

1.3 Documentum Content Services for SAP Solutions architecture

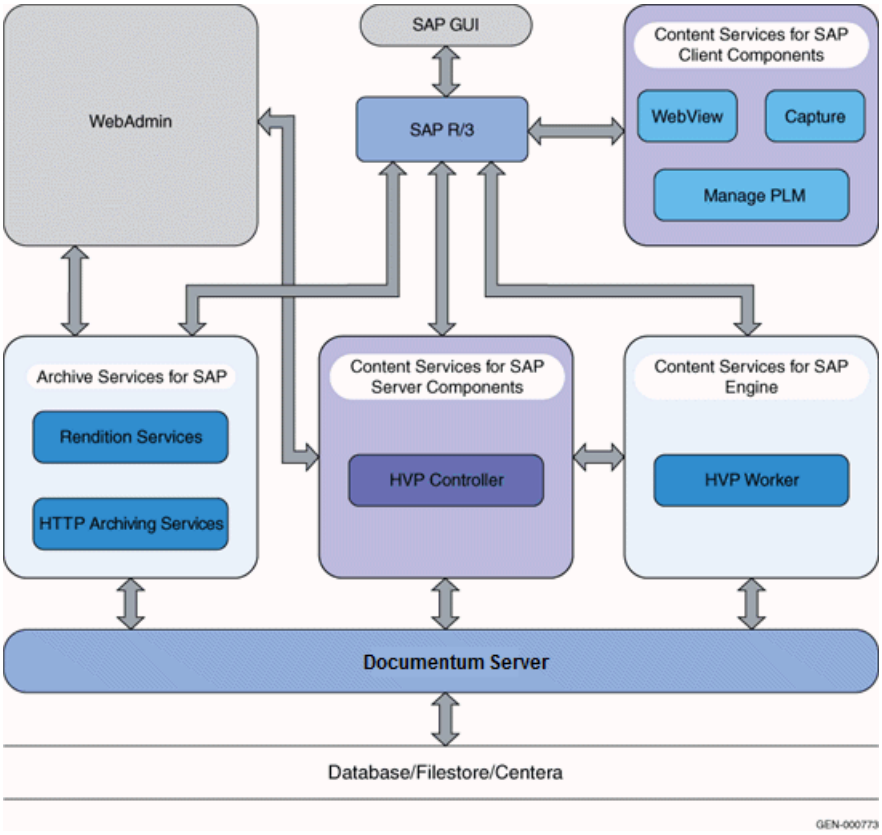


Figure 1-1: Documentum Content Services for SAP Solutions and SAP

Note: Content Services for SAP capture component is not present in the 7.2 and later releases.

Documentum Content Services for SAP Solutions consists of the following components:

- WebAdmin

  An administrative console that allows you to:

  – Perform archiving services.

    Using WebAdmin, you can perform the following archiving services:

    ○ Create, configure, and manage archives.

    ○ Manage certificates for the archive.

    - Configure document archival for Documentum Content Services for SAP Solutions.
  - Perform Document Management System related services. You can:
    - Configure batch jobs for linking and replication tasks.
    - Schedule and manage the execution of linking and replication jobs.

- High Volume Processing Server (HVPS)

  A server component that allows you to automate the linking between SAP objects and documents, as well as maintain those links. This automation improves productivity, information integrity, and information availability. This replaces the Documentum Content Services for SAP Solutions Agent component present in 5.3 and 6.0 releases. Unlike the previous Agent, HVPS is completely a Java-based component. The following are the two main components:

  - HVP Controller: This server component is packaged as the `HVPS.dar` file. HVP Controller runs in the realm of the method server and delegates the jobs executed from WebAdmin to the registered HVP Workers.
  - HVP Worker: The Worker is a .war file which can be installed on any supported application server. For more information about supported application servers, see *Release notes*. The jobs delegated by the controller are executed by the HVP Worker. The HVP Worker packages DFC libraries that is used to communicate with the repository. It communicates with SAP using the SAP JCo v3.1.x libraries. The JCo libraries have to be downloaded from the SAP marketplace into the lib directory of HVP Worker.

- View

  View is a thick client which must be installed on every client computer that needs to access reports, drawings, and other documents stored in a OpenText Documentum CM repository.

  The View component supports Windows 10.

- WebView

  This is a Web-based View application which could be installed on any supported application server. The advantage of using WebView over View is that the user need not have to install the View application on every client computer and the documents reports are rendered on to the browser window. Unlike View, WebView supports Linux-based operating systems also. WebView also has Daeja viewer plugin which enables users to open TIFF and PDF documents inside the browser window and provides tools to annotate.

- Manage

  Enables you to release OpenText Documentum CM documents to SAP, and to maintain those released documents.

Note: The *OpenText Documentum Content Services for SAP Release Notes* provides detailed information about specific Operating System versions and supported software environments.

1.4 Supported SAP document formats

OpenText Documentum CM supports the following SAP document classes/formats:

- Incoming or scanned documents (FAX class, TIFF format)
- Outgoing documents (OTF class, PDF format)
- Archived data (REO class, REO format)
- Reports or print lists (ALF class, ALF format)

1.5 SAP and Documentum Content Services for SAP Solutions terms

The following table presents the SAP and Documentum Content Services for SAP Solutions terms and their definitions used in this guide.

Table 1-1: Terms and definitions

Term	Definition
CS SAP	Product that connects OpenText Documentum CM to and from SAP.
ArchiveLink	Cross-functional interface that is part of the SAP Basis System. ArchiveLink handles the storing and retrieval of documents and data to and from a repository external to SAP.
WebAdmin	An administrative console that allows you to perform: • Archiving services: – Create, configure, and manage archives. – Manage certificates for the archive. – Configure document archival for Documentum Content Services for SAP Solutions. • Document Management Services-related services: – Configure batch jobs for linking and replication tasks. – Schedule and manage the execution of linking and replication jobs.

Term	Definition
HVPS	A server component that allows you to automate the linking between SAP objects and documents, as well as maintain those links. This automation improves productivity, information integrity, and information availability. This replaces the Documentum Content Services for SAP Solutions Agent component present in 5.3 and 6.0 releases. Unlike the previous Agent, HVPS is completely a Java-based component.
SAP Document Management System (DMS)	Document Management System that is part of the SAP Basis System. Presents a logical layer to integrate with external systems such as AutoCAD or OpenText Documentum CM. Not directly related to ArchiveLink.
SAP Product Lifecycle Management (PLM)	SAP Product Lifecycle Management (PLM) provides an integrated environment that ensures all personnel involved in product development, manufacturing, and service have quick and secure access to current information. It provides a set of BAPI calls that can be used by external systems. For example, OpenText Documentum CM.
SAP Master Record	A set of master data, such as customer or vendor data, which is used in the creation of SAP documents.
SAP GUI	SAP graphical user interface is a graphical menu/screen tool that connects a client to the SAP server.
Original document	Paper-based version of a document. For example, an invoice can consist of two sheets of paper received from a supplier. Paper documents are scanned in and stored as electronic documents in OpenText Documentum CM.
SAP document	An electronic transactional record of header data and line items in SAP.

Chapter 2

Introducing WebAdmin

2.1 Overview

WebAdmin is a browser-based tool hosted within the Documentum Administrator that you can use to configure and administer Documentum Content Services for SAP Solutions.

Note: Make sure that you have deployed Documentum Administrator to access the WebAdmin tool. For deployment instructions, see *OpenText Documentum Content Management - Server and Server Extensions Installation Guide (EDCSY250400-IGD)*.

You can perform the following tasks in WebAdmin:

- "Configuring connections to SAP" on page 17
- "Configuring HTTP Archiving Services" on page 19
- "Configuring the Agent component" on page 31

2.2 Logging in to WebAdmin through Documentum Administrator

Log in to Documentum Administrator to administer Documentum Content Services for SAP Solutions.

Note: The *OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)* has complete information about using Documentum Administrator.

To connect to WebAdmin through Documentum Administrator:

1. Start a web browser on a client machine.
2. Connect to the following URL, where *host* is the host where Documentum Administrator is installed and *portnumber* is a port number provided during application server installation:

   ```
   http://<host>:<portnumber>/da/
   ```

3. Type your login name and password.
4. Select a repository from the list box.

   If you change the repository, retype your password.

5. In the **Location** list (if available), select the location on your organization's network from which you are accessing Documentum Administrator.

   This allows you to access content from the nearest storage area in the network. Depending on your organization's setup, this location might be a fixed value.

6. To view additional options, click **More Options**:

   a. To connect to the repository using a particular server, select that server from the **Server** list box.

      The default is **Any Running Server**.

   b. If the repository is running in domain-required mode, type the domain name.

   c. To set the session locale to another language, select the language from the list box.

      Note: Do not click the **Additional Accessibility Options** link on the login page. Documentum Administrator does not support the accessibility options.

   d. To change your password in a repository, click **Change Password**, select a repository and type your old and new passwords, and then click **Change Password**.

7. Click **Login**.

8. The **System Information** page appears with information about the system.

   The *OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)* has complete information about using Documentum Administrator and logging in to repositories available in Documentum Administrator.

2.3 Introducing the Content Services for SAP Solutions node of the WebAdmin GUI

After you have logged in to Documentum Administrator and the System Information page appears, you can select and expand the Content Services for SAP Solutions node located under the Administration node on the left-hand tree.

The *OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)* has complete information about using Documentum Administrator and logging in to the repositories available in Documentum Administrator.

To use the following subnodes, you must have completed the installation of Documentum Content Services for SAP solutions. For more information, see *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)*.

- Actions

- ArchiveLink
- Auto Manage
- Clients
- Documentum
- SAP

Each subnode contains additional subnodes used to perform Documentum Content Services for SAP Solutions functions:

- The *Actions* subnode lets you create Content Services Actions which perform document linking, data replication, and integrity checking functions.
- The *Archive Link* subnode lets you configure archives, Barcodes for Archive Link, and certificate management.
- The *Auto Manage* subnode lets you set up agent configurations, register and de-register HVP Workers, configure jobs to run agents, and monitor job progress.
- The *Clients* subnode lets you configure Content Services for the Content Services View client application.
- The *Documentum* subnode lets you define OpenText Documentum CM Queries.
- The *SAP* subnode lets you define SAP Queries and configure SAP Servers and Users used by Content Services.

These functions are described in the following table.

Table 2-1: Functions in Documentum Content Services for SAP Solutions (WebAdmin interface)

Subnode	Functions	Procedures
Documentum	Query	"Documentum queries" on page 34.
SAP	Query	"SAP queries" on page 32.
	Server	"Creating, viewing, and editing connections to a SAP server" on page 17.
	User	"Creating, viewing, and editing an SAP user" on page 18.
Actions	Link SAP	"Creating, viewing, and editing SAP to OpenText Documentum CM links" on page 41.

Subnode	Functions	Procedures
	Link OpenText Documentum CM	"Creating, viewing, and editing OpenText Documentum CM to SAP links" on page 44.
	Replicate SAP	"Replicating SAP objects" on page 52.
	Replicate OpenText Documentum CM	"Replicating OpenText Documentum CM objects" on page 55.
	Check Document Info Records	"Checking the integrity of linked documents" on page 50.
	Unlink Document Info Record	"Unlink Document Info Record action" on page 58
Archive Link	Archives	"Configuring, viewing, and editing archives" on page 19.
	Barcodes	"Configuring HTTP barcodes for archive linking" on page 28.
	Certificates	"Configuring HTTP certificates for archive linking" on page 29
	Repositories	"Configuring HTTP repositories for archive linking" on page 29
Auto Manage	Agents	"Creating, viewing, and editing an Agent" on page 71.
	HVPS	"Creating, viewing, and editing HVPS" on page 72
	Jobs	"Creating, viewing, and editing Documentum Content Services for SAP Solutions jobs" on page 73.
Clients	View	"Configuring the View component" on page 77.

Chapter 3

Configuring connections to SAP

Before you can use Content Services Archive or Agent functionality, configure the SAP server and user information in Documentum Content Services for SAP Solutions.

In order to communicate with both SAP and OpenText Documentum CM, Documentum Content Services for SAP Solutions must know the server and user login details for each system. The OpenText Documentum CM login parameters are specified when the Archive or Agent services are created, as described in "Configuring, viewing, and editing archives" on page 19 and "Creating, viewing, and editing an Agent" on page 71. When Documentum Content Services for SAP Solutions connects to OpenText™ Documentum™ Content Management Server, it reads the SAP server and user configuration parameters from the repository.

Documentum Content Services for SAP Solutions was designed so that you can configure multiple SAP servers and users. This allows Documentum Content Services for SAP Solutions to be used across multiple SAP application servers.

Create a specific user in your SAP system for use with Documentum Content Services for SAP Solutions.

The procedures in this chapter describe how to configure SAP servers and SAP users which the WebAdmin application uses to access SAP.

3.1 Creating, viewing, and editing connections to a SAP server

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **SAP** subnode and select the **Server** subnode.

   The **Server** screen appears.
3. Select **File** > **New** > **SAP Server** from the menu at the top of the **Server** screen.

   The **New SAP Server** screen appears.
4. Type a name for the Server in the **New Server Name:** field.
5. To log in to an SAP server, type the hostname or IP address for the server. When an SAP router is used, fill in the complete SAP router string in the following format:

   ```
   /H/router1/H/<host_name_or_IP_address>
   ```

6. If you want to log in to an SAP group, which is associated with an SAP R/3 server, select **Enable load balancing**, and type the <*SAP_group*> in the SAP Logon group field.

   Note: For SAP Server, enable load balancing configuration for Windows Operating System , add an entry "sapms<SID>3600/tcp # SAP System Message Port in C:\WINDOWS\system32\drivers\etc-> services file. Do not add this entry at the end of the file, instead add it somewhere in between just before sapdp00 3200/tcp.

   Edit the file on the system where Documentum Content Services for SAP Solutions WebAdmin is running.

   Also, try restarting the system if the changes do not work.

7. Type the system name and number in the appropriate fields.

8. Click **OK** to save the SAP server configuration.

3.2 Creating, viewing, and editing an SAP user

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.

2. Click to expand the **SAP** subnode and select the **User** subnode.

   The **SAP User** screen appears.

   Note: To enable worklist and links creation in Documentum Content Services for SAP Solutions WebAdmin, the recommended authorization profiles for SAP users are:

   - SAP_ALL
   - SAP_NEW

3. Select **File** > **New** > **SAP User** from the menu at the top of the **SAP User** screen.

   The **New SAP User** screen appears.

4. Type the new username in the **New User Name:** field.

5. Type the SAP username the **User ID:** field.

6. Type a password for the user.

7. Type the client number.

8. Select the language for the user from the **Language:** list box.

9. Click **OK** to save the SAP user configuration.

Chapter 4

Configuring HTTP Archiving Services

4.1 Overview

Note: Documentum Content Services for SAP Solutions does not include the HTTP archiving services component; this component has been moved to Documentum Archive Services for SAP (AS SAP). The instructions contained in this section are relevant only if you have parallel installations of Documentum Content Services for SAP Solutions and in your environment.

SAP must be configured to work with Documentum Content Services for SAP Solutions. Information about configuring SAP using SAPGUI is in the *OpenText Documentum Content Services for SAP Solutions - Configuration Guide (EDCCOSAPCS250400-CGD)*.

4.2 Configuring, viewing, and editing archives

SAP uses named “logical archives” as a mechanism to specify target storage. Installations have a number of “archives” relating to different types of information which gets archived. For example:

- Archive AA may be used to archive printlists from SAP. As an administrator, you may want to configure the system to store printlists within the OpenText Documentum Content Management (CM) Server folder /SAP/Printlists.
- Archive BB may be used to archive outgoing documents from SAP. As an administrator, you may want to configure the system to store outgoing documents within the Documentum CM Server folder /SAP/Outgoing.

WebAdmin allows you to specify rules for how to handle archived documents/data from SAP. As shown previously, this may be simply to store different types of information in different locations for better housekeeping. However, it may also be desirable to specify access permissions, initiate workflows, or define whether received documents should be rendered into formats such as HTML and PDF.

All configuration objects created in WebAdmin are stored within the Documentum CM Server. For example, each archive configuration, such as AA, BB, can be found in the Documentum CM Server folder /System/DocLink/SAP/Archive.

Before configuring an archive in WebAdmin, you must first create the archive in SAP. When this has been done, WebAdmin can be used to mirror the SAP configuration and define Documentum CM Server specific configuration options.

To configure, view, or edit archives:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **ArchiveLink** subnode and select the **Archive** subnode.

   The **Archive** screen appears.
3. Select **File** > **New** > **Archive** from the menu at the top of the Archive screen.

   The **New Archive** screen appears.
4. Type the archive name in the **Archive Name:** field.

   You can use names up to 30 character in length for archives, when supported by SAP.
5. The following parameters can be configured.

Table 4-1: Valid entries

Fieldname	Description
Archive ID	Name of the SAP archive using a two-letter string. The installation script creates a sample archive named 'AA.'
SAP Document Type	Set to NONE (HTTP provided).
Documentum Type	Specifies the OpenText Documentum CM document type, as described in "Configuring the repository document type" on page 21.
Workflow	Set to No Workflow.
Attribute Map	The attribute map is used to define the OpenText Documentum CM attributes of an archived document. There is a special attribute "FOLDER" that can be configured. If nothing is specified, the document is stored in the default cabinet. To specify the folder path, use the same format string as for the Agent attribute maps. Example: "FOLDER="/SAP/Archive/AA".
Filtering	Custom Filter Specifies a server method that is executed when a document is stored. This allows you to filter attributes and to do additional tasks when a document is saved, as described in "Specifying a custom filter" on page 22.

Fieldname	Description
	Built-in Filter Allows you to specify what filters are applied to convert the ALF format into XML for output to PDF, ASCII, or HTML. Requires the document type to be sap_print_list, as described in "Specifying a built-in filter" on page 22.
	Service-based business objects (SBO) Allows you to customize archived object behavior, as described in Specifying a custom filter.

These items are further explained in the following sections.

6. Click **OK** to save the archive configuration.

4.2.1 Deleting archived and linked documents

In a repository, if you delete version 1.0 of a document that is linked to SAP or archived from SAP, the link to SAP is also deleted. This is because the dm_relation object which creates the link to SAP is deleted when the parent object (which is always version 1.0) is deleted.

OpenText recommends to not delete the original version of objects that are linked to SAP if you want to maintain their link to SAP. If you need to delete version 1.0 of a document, but want to keep the link to SAP, after deleting the document, you must relink the object to SAP, outside of Documentum Content Services for SAP Solutions.

4.2.2 Configuring the repository document type

The value of the Document Type field defines the object type used to store the document in the repository. The default format is sap_archive. This object type must be a subtype of dm_document, for example, dm_doc_ebr.

If you use filters that extract additional attributes, this parameter must be set to the corresponding OpenText Documentum CM document type.

4.2.3 Specifying a custom filter

Optional: Type the name of a custom filter here.

A custom filter is usually a Docbasic or Java program that is stored as content of a specific method (dm_method) or an SBO. For example, a custom filter may parse the archived file and extract attributes from the document content. The attributes are then passed back to the Content Services Archive software and stored as custom attributes. Or, a custom filter can create queries to attach other documents (such as SOPs) as virtual components to the archived document.

Custom filters have to be marked with a leading exclamation mark if they are external executables and not dm_methods. The complete path to the executable has to be provided after the exclamation mark.

For example:

```
!C:\production\extract_keys.exe
```

SBO custom filters must be marked with a leading exclamation mark and pound sign (!#).

For example:

```
!#mySBOName
```

Customizing archives using service-based business objects provides more information.

4.2.4 Specifying a built-in filter

Using existing OpenText Documentum CM filters, you can define additional actions performed when a PrintList is archived. The following filters are currently implemented:

- *make_pdf*: A PDF rendition is generated by the Content Services software and added to the archived PrintList. To create a PDF rendition, you may want to define parameters to control how the rendition is formatted.
- *make_text*: An ASCII text rendition is generated by the Content Services Archive software and added to the archived PrintList.
- *make_html*: An HTML rendition is generated by the Content Services Archive software and added to the archived PrintList.

4.2.5 Implementing external filters

The filter mechanism allows you to customize Content Services Archive. You can write a filter program that parses the file to be archived and extracts special attributes for storage with the archived document.

The filter can be written in any programming or scripting language, such as Docbasic, C, C++, Perl, and JDK. It must be configured in the document profile with the Custom Filter Method parameter as described in Specifying a custom filter. The filter gets a number of arguments on the command line and it writes the result back to the Content Services Archive process. For performance reasons, the filter does not need to access the repository (but it is possible if really needed).

The filter is called with the following command line parameters:

```
path dm_doc_type dm_archive object_id repository_name dm_ticket
```

The parameters are:

- *path*: Full path of the ASCII text rendition of the file to be archived. Example: '/tmp/S567378.txt.'
- *dm_doc_type*: SAP document type for which this filter is defined. Example: 'ALF.'
- *dm_archive*: SAP archive ID. Example: 'AA.'
- *object_id*: Document ID of the document created in the repository. Example: '09001edc800003af.'
- *repository_name*: Name of the current repository. This parameter is used when the filter has to connect to the repository.
- *dm_ticket*: Encrypted ticket string for the user.

The filter passes the result back simply by writing to the standard output. Additionally, it must return 0 (zero) when the program exits, as shown in this table.

Table 4-2: External filters

Language	Syntax
Docbasic	`print...`
C	`fprintf(stdout,"...")`
C++	`cout << "...")`
Java	

The following parameters allow the filter to pass results back to Content Services Archive:

- *set,<any attribute name>,<value>*: Defines an attribute with a given value. The attribute must exist for the object type used. By default, the object type is 'dm_document.' If additional attributes must be stored, you must define a new

subtype of ‘dm_document' and define the attributes that the filter uses. Use the configuration parameter ‘SAP Obj Type' when using a filter with different object types.

Example: 'set,object_name,PI Sheet 4711'

- *virtual,<obj type> where <qualification>*: Allows you to specify a query that selects documents to attach to the archived document as virtual components.

  Example: ' virtual,dm_document where title is 'SOP 4711%'

- *error,<error message>*: If the filter wants to report an error. OpenText recommends storing the error on the first line of the file. The error message is written to the log file and the operator is notified.

  Example: 'error,Cannot open file'

4.2.5.1 Example: PI sheet

This example creates a custom filter which extracts specific attributes from archived documents. This example uses the PI Sheet filter that was installed with the Content Services Archive software. It assumes that a second filter was installed for Inspection Lots. This filter looks similar to the PI Sheet filter, but is not explained in this section. This example is already installed and configured so it is not required to perform the steps explained in this section.

The purpose of the following customization is to extract some document attributes from an archived PI sheet. These document attributes will enable standard OpenText Documentum CM queries to find the PI sheet again.

The first few lines of the archived PI sheet appear as follows:

```
------------------------------------------------------------
PI sheet            : 100000000000002128
Proc. order         : YMM_14
Plant               : 0001
CntlRecDestin.      : 01
Operating grp.      : GROUP 1
Dest.type           : 1
Test                :
Status              : 00005
Created on          : 05.01.1996
                    : 10:22:36
Changed on          : 05.01.1996
------------------------------------------------------------
```

To create a customized PI sheet filter:

1. Define a new document type named dm_pi_sheet.

   This new document type defines the attributes you wish to extract. The document type is defined with the following DQL statement:

   ```
   CREATE TYPE dm_pi_sheet (
       proc_order char(32),plant char(32),ctrl_rec_dest char(32),
       operating_grp char(32),dest_type char(32),status char(32)
       ) WITH SUPERTYPE dm_document
   ```

2. Create a filter that parses the PI Sheet and defines the attributes in Docbasic:

```
Sub GetMatch(ll As String, match As String, delimiter As String, ByRef res

As String)
If InStr(ll, match) = 1 Then
        pos = InStr(ll, delimiter)
        If pos > 0 Then
            fld$ = Mid$(ll, pos + 2)
            res = Trim$(fld$)
        End If
    End If
End Sub
```

```
Sub Filter(arg_path As String, arg_dm_doc_type As String,_
             arg_dm_archive As String,_
             arg_obj_id As String, arg_docbase As String,_
             arg_user As String, arg_passwd As String)
' open file and get values into variables
    file% = FreeFile
    Open arg_path For Input As file%
    Count = 0
    Do While Not EOF(file%)
' read each line and try to find values
        Line Input #file%, ll$
        GetMatch ll$, "PI sheet", ":", pi_sheet$
        GetMatch ll$, "Proc. order", ":", proc_order$
        GetMatch ll$, "Plant", ":", plant$
        GetMatch ll$, "CntlRecDestin.", ":", ctrl_rec_dest$
        GetMatch ll$, "Operating grp.", ":", operating_grp$
        GetMatch ll$, "Dest.type", ":", dest_type$
        GetMatch ll$, "Status", ":", status$
' definitions must be within the 20 first lines
        Count = Count + 1
        If (Count > 20) Then
            Exit Do
        End If
    Loop
'write attributes and content to stdout
    Print "set,object_name," + pi_sheet$
    Print "set,proc_order," + proc_order$
    Print "set,plant," + plant$
    Print "set,ctrl_rec_dest," + ctrl_rec_dest$
    Print "set,operating_grp," + operating_grp$
    Print "set,dest_type," + dest_type$
    Print "set,status," + status$

    Exit Sub
End Sub
```

3. Create a method named dm_filter_pisheet with the following DQL statement:

```
        CREATE dm_method OBJECT set object_name='dm_filter_pisheet',
        set method_verb='dmbasic -eFilter',set timeout_min=30,
        set timeout_max=604800,set timeout_default=86400,
        set run_as_server=TRUE,set use_method_content=TRUE,
        set     method_type='dmbasic'
```

4. Use the object ID of the created method and store the Docbasic file with the following API methods.

   The DQL statement in the previous step returned the object ID of the method created:

```
        setfile,c,<ID of dm_method>,<Docbasic path>,crtext
        save,c,<ID of dm_method>
```

5. In WebAdmin, create an archive named PI. Define this archive to use folder / SAP/PI Sheets.

Using this archive from SAP ensures that all PI Sheets are stored in this folder.

6. Configure the archive PI in SAP.

   Make sure PI Sheets are archived to this archive.

7. Create a profile object (dm_al_profile) called ALF-PI.

   This profile is applied when a document of the SAP document type ‘ALF' is archived to the archive ‘PI':

   a. Define Document Type as ‘dm_pi_sheet.'
   b. Define Document Format and SAP Retrieve Format as ‘sap_print_list.'
   c. Activate the Built-In Filter parameter as ‘make_pdf' or ‘make_html' if required.
   d. Define Custom Filter Method as ‘dm_filter_pisheet.'

   Note: This step is very important.

8. Test your customized filter by archiving a PI Sheet.

   Check attributes and renditions to verify that the filter implementation worked correctly.

4.2.6 Customizing archives using service-based business objects

OpenText Documentum CM Business Objects are designed to provide modular business logic to the presentation layer by hiding the underlying repository schema and by using OpenText Documentum CM core services, facilitating customization of object behavior without modifying any existing application built on OpenText Documentum Content Management (CM) Foundation Java API. Service-based business objects (SBOs) are generalized objects that provide a specific service that may operate on different OpenText Documentum CM object types or other business objects, and are not tied to a specific OpenText Documentum CM object type. Each service-based business object provides a generalized interface to a group of related operations that need to be performed. The operations may not need access to a repository; however, content management services are the focus of OpenText Documentum CM Business Objects.

The archiving operation can be customized using a custom filter, such as an SBO. To enable OpenText Documentum CM archiving customization using SBOs, an archive configuration can specify an SBO as a custom filter. Archive Services will dynamically execute a method “doArchive (IDfPersistentObject obj,String archiveID) throws DfException” which should be defined in the SBO:

1. The SBO must have a method void doArchive (IDfPersistentObject pobj, String archiveID) throws DfException.

2. The message returned to SAP HTTP response, in the event of any error while executing the archive customization method, should be returned by doArchive(..) method in the exception message.

3. The call to doArchive(..) runs within the context of an archiving transaction and will do a commit() when SBO doArchive(..) is successfully executed.
4. The SBO module need not handle any function for session management for the SessionManager passed by . For example: Transactions, Session creation, or release. SBOs can obtain session by calling getSessionManager(), getSession(), or just getSession().
5. SBOs should not release the session obtained by session management described in step 4. However, if any session manager or session is created in SBO explicitly, SBO has the responsibility to release it.
6. Pass the IDfPersistentObject corresponding to the archived object to doArchive(..) method of the SBO. In addition, the archive ID will be passed (if it is an archive config object).
7. Set the SessionManager corresponding to credentials specified in repository configuration for the repository to the SBO.

4.2.6.1 Customizing archives using SBOs

To customize platforms using service-based business objects (SBOs):

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **ArchiveLink** subnode and select the **Archive** subnode.

   The **Archive** screen appears.
3. Select **File** > **New** > **Archive** from the menu at the top of the Archive screen.

   The **New Archive** screen appears.
4. Type the SBO service name, prefixed with `!#` in the custom filter text box, and click **OK**.
5. Archive a document to the content repository from SAP.

   The customized functionality implemented in `SBOs doArchive()` method executes.

4.3 Managing temporary disk space in the Documentum Content Services for SAP Solutions host

When an SAP archive file is accessed, Documentum Content Services for SAP Solutions fetches the file to the local disk, and then starts streaming the content back to SAP. These locally cached files are managed by DMCL, and an algorithm implemented in DMCL determines when the files are cleaned up. If dmcl.ini is not configured appropriately, the disk may reach its default maximum capacity at some point. In order to avoid choking the disk space with these temporary files, modify the local_diskfull_limit attribute of the dmcl.ini file.

The local_diskfull_limit attribute specifies the maximum disk space assigned for storing locally cached files, and is expressed as a percentage between 1 and 100.

To configure when OpenText Documentum CM should warn you about an impending shortage of disk space, modify the local_diskfull_warn attribute. This attribute is expressed as a percentage between 1 and 100.

4.4 Configuring HTTP barcodes for archive linking

In the HTTP archive scenario, Agent services process “barcoded documents” and link them to SAP.

A typical scenario for implementing barcode support is in “late archiving with barcodes.”

Note: Ensure that barcodes are available for linking:

- The image is scanned.
- The barcode is recognized (by third-party software).
- The barcode is stored as a number in an object attribute (by third-party software).

To configure HTTP barcodes for archive linking:

1. Connect to WebAdmin. “Logging in to WebAdmin through Documentum Administrator” on page 13 provides information.
2. Click to expand the **ArchiveLink** subnode and select the **Barcodes** subnode.

   The **Barcodes** screen appears.
3. Select **File** > **New** > **Barcode** from the menu at the top of the **Barcode** screen.

   The **New Barcode** screen appears.
4. Select the document type from the **Choose a Document Type:** list box.

5. Select the barcode storage attribute from the **Barcode stored in attribute:** list box.
6. Select the archive for use from the **Archive to use:** list box.
7. Click **OK** to save the barcode configuration.

4.5 Configuring HTTP certificates for archive linking

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **ArchiveLink** subnode and select the **Certificates** subnode.

   The **Certificates** screen appears.
3. Right-click on a certificate and select **Properties** from the submenu.

   Selecting **Delete** removes the Certificate from the Documentum Content Services for SAP Solutions repository.
4. Select **Activate** or **Deactivate** from the **Status:** list box.
5. Select a certificate expiration date from the **Expiration:** calendar menu and list boxes.
6. Click **OK** to save the certificate configuration.

4.6 Configuring HTTP repositories for archive linking

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **ArchiveLink** subnode and select the **Repositories** subnode.

   The **Repositories** screen appears.
3. Select **File** > **New** > **Repository** from the menu at the top of the **Repository** screen.

   The **New Repository** screen appears.
4. Type the connection information for the new repository, as follows:

   Repository Name: Name of the new repository.

   User Name: Username associated with the user of the new repository.

   User Password: User password associated with the username of the user of the new repository.

   Domain: Domain in which the new repository resides.

   Note: Click **Test Connection** to validate the new repository user access credentials.
5. Type the connection information for the global repository associated with the new repository, as follows:

User Name: Username associated with the user of the global repository.

User Password: User password associated with the username of the user of the global repository.

Domain: Domain in which the global repository resides.

6. Click **OK** to save the new repository configuration.

Chapter 5

Configuring the Agent component

The Agent component substantially improves productivity, information integrity, and information availability by automating the linking process between SAP objects and documents, and the maintenance of those links.

For example, attribute information from scanned invoices can be automatically replicated from SAP to OpenText Documentum CM, providing non–SAP users with searchable access to invoices without having to learn SAPGUI.

5.1 Overview

The Agent component provides an automatic means of linking and replicating objects in SAP and OpenText Documentum CM.

There are three parts to the Agent component known as Actions, Agents, and Jobs. Actions define what has to be done.

There are five types of actions:

1. Linking SAP objects to OpenText Documentum CM objects.
2. Linking OpenText Documentum CM objects to SAP objects.
3. Replicating SAP objects into a OpenText Documentum CM repository.
4. Replicating OpenText Documentum CM objects into SAP.
5. Checking the integrity of the linked objects in SAP and OpenText Documentum CM.

   All actions use queries to perform these tasks.

   Agents run the actions. The Agent defines on what SAP system and with which user an action is carried out.

   Jobs are scheduled events that can start Agents. There can be multiple Agents attached to a Job that are run one after the other. The job defines when the Agents have to run, according to a specified schedule.

   The following illustration shows the relationship between these parts.

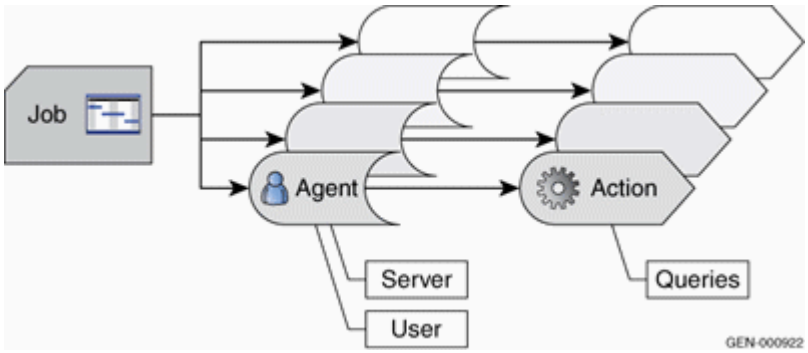


Figure 5-1: Agent services

5.2 Configuring queries

Actions depend on queries to identify objects that need to be linked or replicated. The queries can be made on SAP or OpenText Documentum CM systems.

5.3 SAP queries

In order to test SAP queries, at least one SAP user and one SAP server have to be configured.

This specifies a query that identifies all SAP objects that must be linked to a dynamic Documentum query or to a repository folder. The query is either a query through the Product Lifecycle Management Interface, a BAPI, or SAP table query.

5.3.1 Creating, viewing, and editing an SAP query

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **SAP** subnode and select the **Query** subnode.

   The **Query** screen appears.
3. Select **File** > **New** > **SAP Query** from the menu at the top of the **Query** screen.

   The **New SAP Query** screen appears.
4. Type the query name in the **Query Name:** field.
5. Select an SAP query type from the **SAP Query Type:** list box. Your choices are described in the following table:

Table 5-1: Query types

Old interface Query types (for SAP R/3 version 4.6c)	New interface Query types (for SAP R/3 version 4.7 and 4.6c)
Document info record	No Object Product Lifecycle Management (Formerly Document info record)
Equipment by short text	Equipment by short text Product Lifecycle Management
Functional location by text	Functional location by text Product Lifecycle Management
Material by description	Material by description Product Lifecycle Management **Note:** The Material by description Product Lifecycle Management query type has three query conditions: 1. `MATERIALSHORTDESCSEL_low` 2. `MATERIALSHORTDESCSEL_Sign` 3. `MATERIALSHORTDESCSEL_Option` All three query conditions are required if the query is to return a result.
Archive data	Archive data
Cost center	Cost center
Financial document	Financial document
Personnel links	Personnel links
Personnel master	Personnel master
Purchasing document	Purchasing document
Customer	Customer Table Product Lifecycle Management
Print list	Print list
Vendor	Vendor Table Product Lifecycle Management
WBS Element	WBS Element Product Lifecycle Management
Asset Master	Asset Master GetList Product Lifecycle Management

6. Build the query condition.

   For each query condition you want to define:

a. Select a parameter from the **Query Condition composer:** list box and type a value for the parameter in the text box.

b. Click the down arrow to add the parameter and value to the **Query Condition:** field.

Note: Highlight an entry in the **Query Condition** field and click '*X*' to delete an entry.

c. Continue to select parameters and type values to build the query condition.

Note: The conditions are `AND` linked.

7. Click **OK** to save SAP Query configuration.

   The **Query** screen reappears with the newly created SAP query.

8. Highlight the newly created SAP query and select **File** > **Test** from the menu at the top of the**Query** screen.

9. Select a Server and a User on which to test the query, and click **Test**.

Note: You must save any amendments before you implement any changes made. If the query execution on the SAP System takes too long, WebAdmin can receive a time out.

The window shows the test results and is blank until the query results are returned.

5.3.2 Documentum queries

This specifies a query that selects the complete set of objects to be linked. The query can be any valid DQL query that selects at least the `r_object_id` and the `object_name` as well as one or several attributes that contain the SAP object information.

5.3.2.1 Creating, viewing, and editing a Documentum query

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.

2. Click to expand the **Documentum** subnode and select the **Query** subnode.

   The **Documentum Query** screen appears.

3. Select **File** > **New** > **Documentum Query** from the menu at the top of the **Documentum Query** screen.

   The **New Documentum Query** screen appears.

4. Type a query name in the **Query Name:** field.

5. Type a DQL statement for the query in the **DQL Statement:** field.

You can use the $ARG expression when defining the DQL statement. For example:

```
select r_object_id,object_name from dm_document
where object_name ='$ARG1'...
```

"Creating, viewing, and editing SAP to OpenText Documentum CM links" on page 41 provides details of the *$ARG* expression.

6. Click **Execute** at the far right of the **DQL Statement:** field.

   The query executes.

7. Click **OK** to save Documentum Query configuration.

   The Documetum Query screen reappears with the newly created Documentum query.

8. Right-click the newly created Documentum query, right–click, and select **Properties**.

   The **Documentum Query Properties** screen appears.

9. Click **Execute** at the far right of the **DQL Statement:** field.

   The query executes.

5.3.2.2 Restricting SAP query results by Documentum query results

Restrict the results of an SAP query by the results of an Documentum query by defining an optional parameter to the SAP query type, `sap_query_plm_type_table`:

1. In the `custom.xml` file for the SAP query type, `sap_query_plm_type_table`, add a query parameter *$DQL*:

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort" NOTE="put your own methods inside this request">
        <OBJECT_CREATE ON_EXIST="version">
            <API_CONFIG TYPE="sap_query_plm_type_table" CLASS="sap">
                <ATTRIBUTE NAME="object_name" IS_KEY="true">
                    EKPO_Table  PLM
                </ATTRIBUTE>
                ...
                <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                    <VALUE>Client=MANDT</VALUE>
                    <VALUE>Document_Number=EBELN</VALUE>
                    <VALUE>Item_Number=EBELP</VALUE>
                    <VALUE>$DQL= </VALUE>
                </ATTRIBUTE>
                ...
            </API_CONFIG>
        </OBJECT_CREATE>
</REQUEST>
```

2. Extract the contents of `dmei_custom_installer.zip` into a temporary folder.

3. Locate the `dfc.properties` file in the temporary folder and edit the file with the directory information and connection broker information (`dfc.docbroker.host`).

4. Locate the `installer.properties` file in the temporary folder and edit the file with the following:

Value	Definition
user.language	The default is en_US (English).
docbase.name	Name of the repository that you want to configure.
docbase.user.name	Type repository install owner name.
domain	Name of the domain in which the repository is located.
custom.xml.path	The custom XML file path.

5. Locate the log4j2.properties file in the temporary folder and edit to change the directory location of the log files.

   Note: By default, log files are generated in the current (temporary) folder.

6. Run the following command from the command line to run the custom XML file installer:

```
java -jar dmei_custom_installer.jar
```

   Note: The file dmei_custom_installer.jar is in the temporary folder where you extracted the contents of the dmei_custom_installer.zip file.

7. Type the password for the repository specified in the installer.properties file.

   File processing begins and the following message appears when processing completes:

```
Processing Successfully Completed!
```

   You have now added the *$DQL* parameter to the SAP query type, sap_query_plm_type_table.

Note: If the *Processing Failed!* message appears, look in the log files to determine the problem with the custom.xml file installation.

A DQL query can be provided as value for the $DQL parameter created previously.

To assign a DQL query as value for the $DQL parameter for the SAP query type, sap_query_plm_type_table:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **SAP** subnode and select the **Query** subnode.

   The **SAP Query** screen appears.
3. Select **File** > **New** > **SAP Query** from the menu at the top of the *Query* screen.

   The **New SAP Query** screen appears.

4. Type a query name in the **Query Name:** field.
5. From the **SAP Query Type** list box, select **EKPO_Table PLM**.
6. From the **Query Condition Composer** list box, select **$DQL**.
7. Type a DQL query as value for this parameter.

   Example 5-1:

   ```
   $DQL=select item_id from sap_abc_document where sap_document_number IS NULL
   ```

8. Click the down arrow to add the parameter and value to the **Query Condition:** field.
9. In **Condition Composer**, compose another condition as follows:

   ```
   Item_number=$item_id
   ```

10. Click the down arrow to add the parameter and value to the **Query Condition:** field.
11. Click **OK** to save the query.

SAP queries can now be restricted by selecting query conditions based on values obtained from the $DQL query result set.

5.3.2.3 Testing queries with $ARG# statements

By configuring a Documentum Query with Arguments ($ARG#) in the DQL statement, the query can be used later for Link Actions.

The $ARG# variable has to be numbered, for example, $ARG1, $ARG2. The variable can be used as a placeholder that will be resolved during runtime.

Example 5-2:

```
select r_object_id, object_name from dm_document where object_name ='$ARG1'
```

This query will select all documents of type dm_document, where the object name equals a given substitute for the argument with the number 1. An explanation on how these queries can be used for a link action is in "Creating, viewing, and editing SAP to OpenText Documentum CM links" on page 41.

When testing, the query will be parsed for occurrences of $ARG# and the user will be prompted to enter a substitution for every argument found.

When all arguments are replaced according to the string, the final query that is about to be tested will be shown.

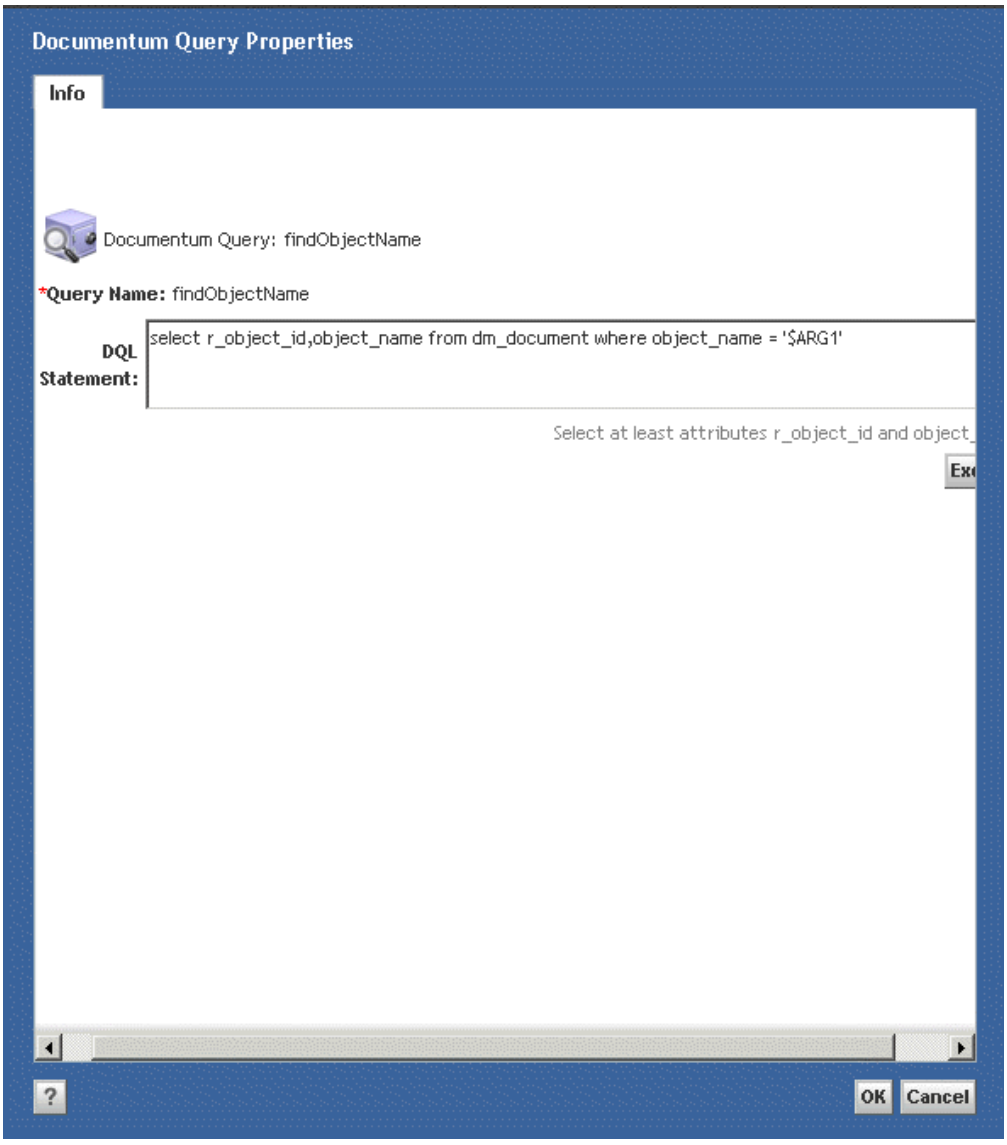


Figure 5-2: Query test

5.3.2.4 Support for $TODAY in FromDate parameter for sap_query_type_rfc query type

Documentum Content Services for SAP Solutions supports $TODAY in FromDate parameter for sap_query_type_rfc query type.

In order to download selected SAP object keys from transaction oa01 (RFC ARCHIV_GET_CONNECTIONS), SAP makes the selection based on the current date. This allows SAP to select a dataset limited by the amount of archived files produced in a day, in addition to a given Business Object Type or Archive Type.

$TODAY parameter can be specified in FromDate parameter in Condition Composer.

To select all BKPF-linked documents archived in the last three days, use a query of type "Financial document." This uses sap_query_type_rfc to call ARCHIV_GET_CONNECTIONS:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **SAP** subnode and select the **Query** subnode.

   The **SAP Query** screen appears.
3. Select **File** > **New** > **SAP Query** from the menu at the top of the Query screen.

   The **New SAP Query** screen appears.
4. Type a query name in the **Query Name:** field.
5. From the **SAP Query Type** list box, select **Financial document**.

   Financial document is of query type sap_query_type_rfc.
6. From the **Query Condition Composer** list box, select **FromDate**.
7. Type the following value for this parameter:

   ```
   $TODAY-3
   ```
8. Click the down arrow to add the parameter and value to the **Query Condition:** field.

   The query would be as follows:

   ```
   FromDate=$TODAY-3
   ```

   Example 5-3:

   If this query was executed on October 17, 2005, then the symbolic value is expanded to 20051017–3 = 20051014. This query instructs ARCHIV_GET_CONNECTIONS to select data of the last 3 days only.
9. Click **OK** to save the query.

5.4 Linking objects

Agent services make use of the SAP Document Management System (DMS) interface to perform linking of objects from OpenText Documentum CM into SAP. The Document Management System interface was originally built to integrate CAD applications into an SAP system. Subsequently, SAP expanded the Document Management System interface to include integrations with Product Lifecycle Management (PLM) systems. The Product Lifecycle Management Interface is the next generation of the Document Management System interface and greatly enhances its functionality.

Note: CAD based applications are not supported starting Documentum Content Services for SAP Solutions 6.5 release.

The linking of SAP object types Business Partner and Opportunities is not supported on SAP CRM as of 6.5 SP2. Only Replicate SAP is supported.

The “PLM Interface” is comprised of a set of API functions and allows you to access the SAP server from external applications such as the Agent services or Product Lifecycle Management Systems, such as OpenText Documentum CM . With the Product Lifecycle Management Interface, you can access SAP server version 4.7 and 4.6c. In addition to accessing the content on your SAP 4.7 server, the Product Lifecycle Management Interface enables you to get editable copies of it, and check it into and out of your OpenText Documentum CM repository.

In order to link from objects in SAP to objects held in OpenText Documentum CM , the SAP Document Management System creates objects in SAP called Document Info Records (DIRs).

A DIR is created in SAP for every document released from OpenText Documentum CM . The DIR contains several attributes such as description, document ID, document version, and a reference to a specific OpenText Documentum CM object in the OpenText Documentum CM repository.

The SAP client application (SAPGUI) can launch an external application for specific content or carrier types (in SAP terminology). These external applications include Content Services View installed on workstations running SAPGUI.

When a document previously released from OpenText Documentum CM into SAP is viewed, Content Services View is launched and the information stored in the DIR is passed to this application. With this information, the Documentum CM Server is queried and the requested document is retrieved and displayed with a viewer application on the SAP workstation.

Document linking actions allow you to specify attributes for the DIR using the “Rule Composer.” The rule composer allows you to specify the following attributes for a DIR:

- DocumentNumber: SAP DIR number. A document number specified by the Agent could be, for example, “DocumentNumber="%s",i_chronicle_id”. This must be a unique number.

- Description: Description attribute of DIR. A value must be defined. Example: “Description="%s",object_name.” If not specified, the object name is used by default.
- DocumentType: SAP document type (for example, “DocumentType="DOC”"). The default is “DRW”.

Note: Since the Product Lifecycle Management type objects are usually drawings, a default value of “DRW” is used. You can always override this setting in the Query Conditions field of the SAP Query Composer, as described in “Creating, viewing, and editing an SAP query” on page 32.

Example 5-4:

You can assign the value of DocumentType as DES for a DES document type.

For all document types, you can set the value of DocumentType as one of the following:

– DocumentType=*

– DocumentType=

In this case, no value has been assigned to the parameter. The value for the parameter has been set to blank.

5.4.1 Creating, viewing, and editing SAP to OpenText Documentum CM links

In the following example you want to link all materials within SAP that have a description starting with “pump*” to a document in the repository with the same object_name as the material name.

You will need to create:

- An SAP Query “Select all pumps PLM” which selects materials with the following condition:

```
Description = pump*
```

This query (Query Type: Material by description Product Lifecycle Management) returns the following attributes:

```
Material, Description, Material_External, Material_Guid, Material_Version
```

- Documentum Query “findObjectName” which selects objects of type dm_document where the object_name equals “$ARG1”:

```
Select r_object_id, object_name from dm_document where object_name = '$ARG1'
```

- Link SAP Action “Link query to pumps PLM” uses both queries (“Select all pumps PLM” and “findObjectName”) and has the following map rules:

$ARG1	= Material
DocumentType	= "DRW"
DocumentDescription	= "Documents for %s", Material

The Link SAP action runs “Select all pumps PLM” (SAP Query), returning a number of SAP material objects with their description matching “pump*”.

For each object returned, the content of attribute “Material” is passed as a substitution for “$ARG1” to “findObjectName” (DCTM Query). Assuming the attribute “Material” of the first object is “Pump One”, the action continues as follows:

- “findObjectName” selects a number of documents [1..n] with an object_name that equals “Pump One”.
- The DIR is created for the current SAP object with links to the documents in the repository named “Pump One”.
- The additional DIR attributes are assigned according to the map rules. The DocumentType is “DRW” and the DocumentDescription is “Documents for Pump One”.
- This loop will be repeated for each object returned by “Select all pumps PLM” (SAP Query), thereby establishing the goal.

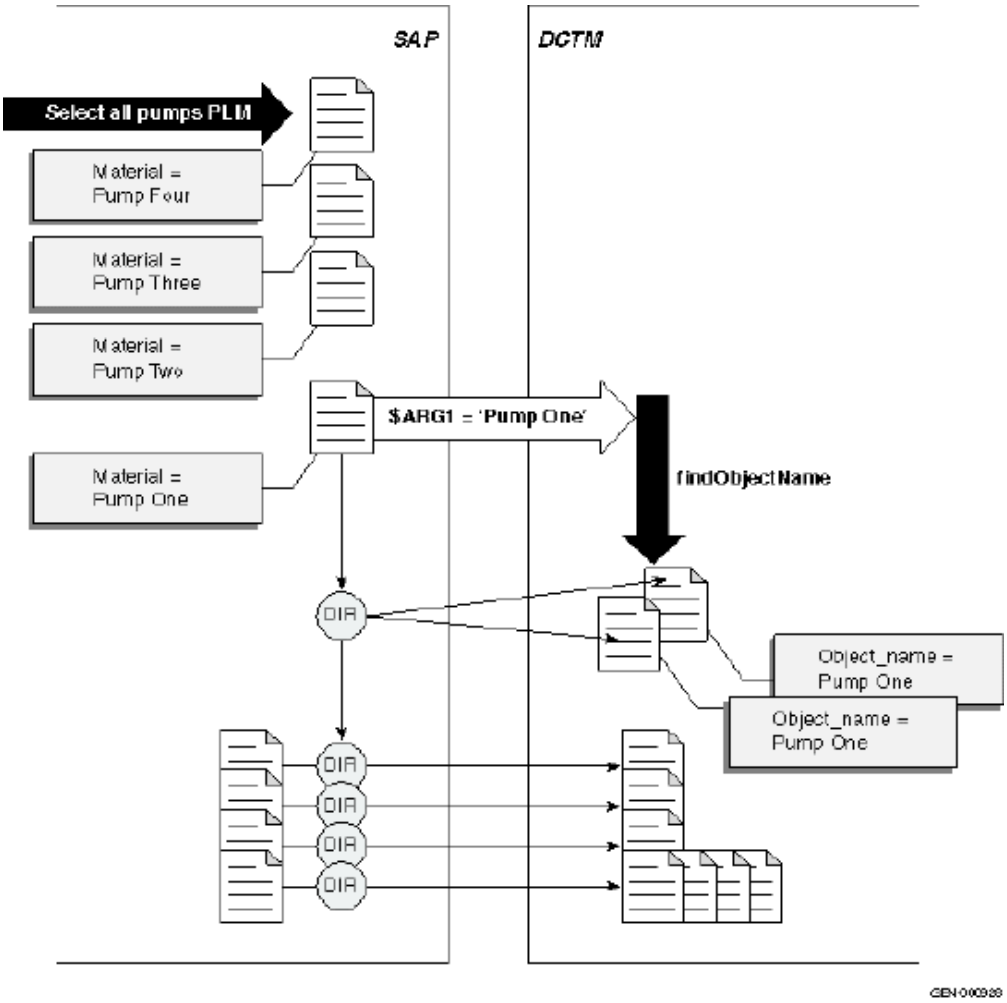


Figure 5-3: SAP to Documentum linking

To configure a Link SAP to OpenText Documentum CM action, you will need a Documentum Query and a previously configured SAP Query.

To create, view, or edit SAP to OpenText Documentum CM links:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Actions** subnode and select the **Link SAP** subnode.

   The **Link SAP** screen appears.
3. Select **File** > **New** > **Link SAP** from the menu at the top of the Link SAP screen.

   The **New Link SAP Action** screen appears.
4. Type an action in the **Action:** field.

5. Select the Documentum query from the **Documentum Query:** list box.
6. Select the SAP query from the **SAP Query:** list box.
7. Link Workflow is set to No Workflow.
8. Check **Verify object links**, if required.
9. Use the Map Rule Composer for each rule you want to define:
   a. Select the variable from the **Variables** list box.
   b. Type the format of the variable in the **Format** field.
   c. Select the parameter required from the list box and click the up arrow to add the parameter to the **Parameters** field.

      The up arrow also alters the format string by adding %s at the end.

      The attribute map allows you to specify the following:

      DIR attributes: When a document is released to SAP, an SAP DIR is created for it. Values can be set for DocumentType, Description, and DocumentNumber. For example, the rule Document Description = "Document for %s", Material will build a DIR description containing the Material attribute from the SAP object returned by the SAP query chosen.

      $ARG#s: In order for the SAP object to be linked to a OpenText Documentum CM object, Content Services must be able to find the related object in the repository. Here you specify the substitution for an $ARG# in the DQL statement of the Documentum query. In this way a lookup into OpenText Documentum CM is defined which identifies the objects which the SAP object should be linked to.
   d. Click the down arrow to add the rule to the **Defined Map Rules:** field.
   e. Click **OK** to save the SAP to OpenText Documentum CM link configuration.

Note: For Link SAP, CRM is not supported and the default SAP System Type is set to SAP R/3.

5.4.2 Creating, viewing, and editing OpenText Documentum CM to SAP links

This action links specific OpenText Documentum CM objects to SAP objects. When configured and executed using the Auto Manage function, this action works as follows:

- Reads the Agent configuration object to retrieve the SAP connection parameters.
- Connects to SAP.
- Executes the Documentum Query.
- Each returned object is then processed as follows:

  - The Attribute Map is used to find the specific related SAP object.
  - If the SAP object is found, then Documentum Content Services for SAP Solutions checks to see if this object is already linked. If it is not linked, then a DIR is created in SAP. The attribute map is used to set the DIR attributes.

In the following example you want to link all subfolders of the folder `/SAP/Material` to a Material in SAP where the material attribute matches the folder name.

You will need to create/configure:

- A Documentum Query “SelectMaterialFolders” selects objects of type dm_folder where the folder's location has to be `/SAP/Material`:

```
select r_object_id,object_name from dm_folder
where folder('/SAP/Material')
```

- “Material by description PLM” is the SAP Object Type that you want to link to your material folders. This SAP Object Type has the following attributes:

```
Material, Description, Material_External, Material_Guid,
Material_Version
```

  The attributes shown with the prefix “key” in the map rules, indicates that they can be used as search conditions in the generated SAP Query.

- Link OpenText Documentum CM Action “Link material folders PLM” has the following map rules:

key.Material	= "%s",object_name
DocumentType	= "DRW"
Description	= "Document folder for %s", object_name
DocumentNumber	=”%s”,r_object_id

The action will link the following version and format of the folders:

Format:	Best Format
Version:	CURRENT

The Link OpenText Documentum CM action runs “SelectMaterialFolders” (Documentum Query), returning a number of folders in the folder `/SAP/Material`.

For each folder returned, the content of attribute “object_name” is passed to an SAP query for SAP object type Material by description Product Lifecycle Management as condition for the attribute Material. Assuming the attribute “object_name” of the first folder is “100-200,” the action continues as follows:

- The created SAP Query for “Material by description PLM” selects one object [1] with the attribute Material matching “100-200”.
- The attributes shown with the prefix “key” in the map rules, indicates that they can be used as search conditions in the generated SAP Query.
- Then the DIR is created for the current folder and the selected SAP objects, using the defined version and format.

- The additional DIR attributes are assigned according to the map rules. The DocumentType is “DRW” and the Description is “Document folder for 100-200”.
- This loop will be repeated for each folder returned by “SelectMaterialFolders” (Documentum Query), thereby establishing the objective.

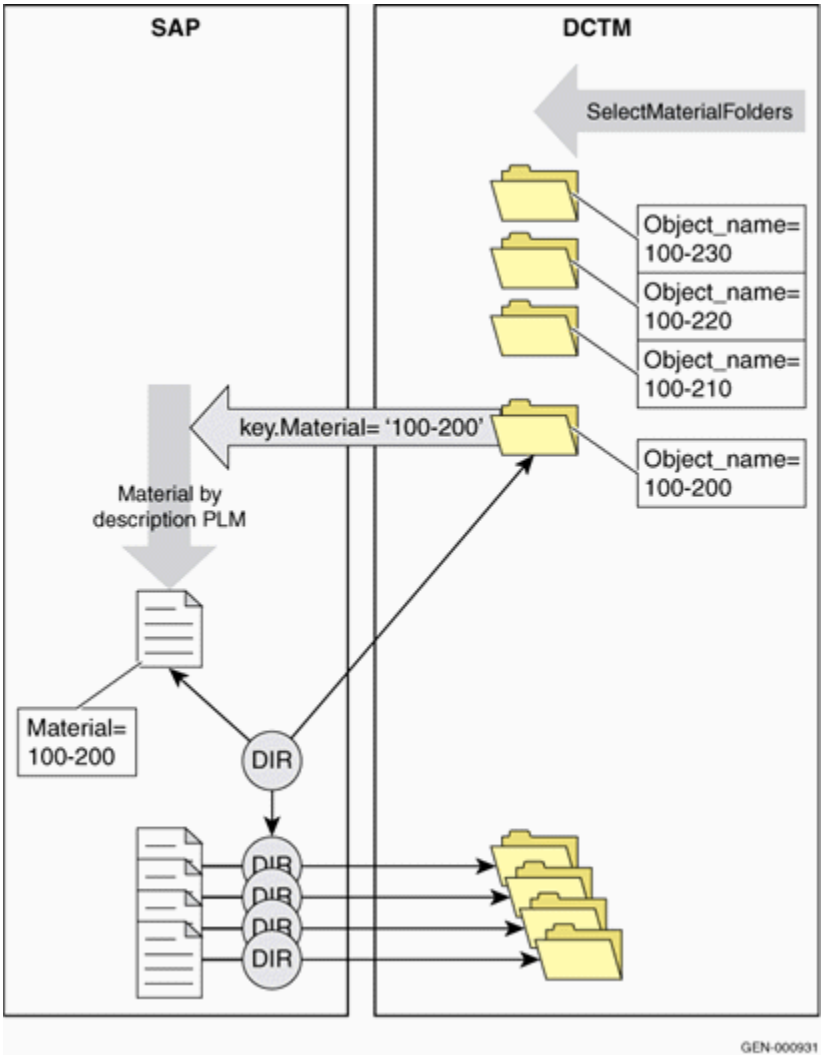


Figure 5-4: Linking result

To configure a Link OpenText Documentum CM to SAP action, you will need a previously configured Documentum Query.

To create, view, or edit OpenText Documentum CM to SAP links:

1. Connect to WebAdmin. “Logging in to WebAdmin through Documentum Administrator” on page 13 provides information.

2. Click to expand the **Actions** subnode and select the **Link Documentum** subnode.

   The **Link Documentum** screen appears.

3. Select **File** > **New** > **Link Documentum** from the menu at the top of the Link Documentum screen.

   The **New Link Documentum Action** screen appears.

4. Type an action name **Action:** field.

5. Select the SAP system from the **SAP System Type:** list box.

6. Select the SAP object from the **SAP Object Type:** list box.

7. Select the Documentum Query from the **Documentum Query:** list box.

8. Link Workflow is set to No Workflow.

9. Select **Verify object links**, if required.

10. Select **Conversion Routines**, if required.

    This executes the **Conversion Routines** associated with the SAP fields to determine the actual input values of SAP to create links between OpenText Documentum CM and SAP.

11. Define the binding rules:

    a. Select the format from the *Format:* list box.

       Type which document format/rendition should be released to SAP. Best Format and Primary Content Format are configured using Documentum Content Services for SAP Solutions WebAdmin as described in Configuring the Manage and View Components.

    b. Select the version of the document that should be released and the version required from the list box to the right of the **Version:** field, and then click the arrow to add that version to the **Version:** field.

       Select a specific version label, such as “DRAFT,” or use a keyword such as dms_selected_version or dms_all_versions. If all versions are released to SAP, then the Documentum Content Services for SAP Solutions Viewer will display a list of all possible document versions.

12. Use the Rule Composer to define each Attribute Map:

    a. Select a variable from the **Variables** list box.
    b. Type the format of the variable in the **Format** field.
    c. Select the parameter from the list box below the **Parameters** field, and click the up arrow to add the parameter to the **Parameters** field.

       The up arrow also alters the format string by adding %s at the end.
    d. Click the down arrow to add the rule to the **Defined Map Rules** field.

       The attribute map allows you to specify the following:

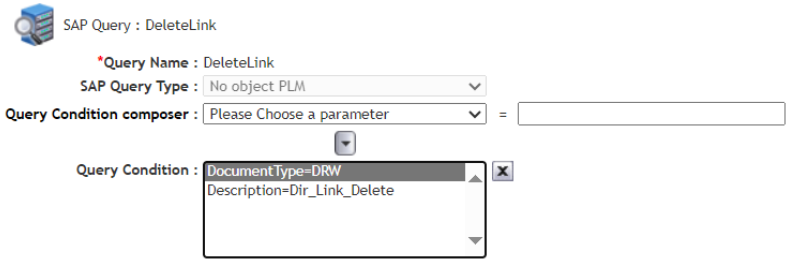
- *DIR attributes*: When a document is released to SAP, an SAP DIR is created for it. Values can be set for DocumentType, Description, and DocumentNumber. For example, the rule Description = "Related SOP for %s," object_name will build a DIR description containing the object_name attribute from the OpenText Documentum CM document.
- *Lookup Key Values*: In order for the OpenText Documentum CM object to be linked to an SAP object, Content Services must be able to find the related object in SAP. Here you specify a lookup into SAP which identifies the single object which the document should be linked to. For example, key.Material="%s", object_name will instruct Content Services to link the OpenText Documentum CM object to an SAP material which has the material name equal to the OpenText Documentum CM object_name.

13. Click **OK** to save the OpenText Documentum CM to SAP link configuration.

    If there is any change in the properties attribute, you can update the respective object links of the DIR with the new properties of the OpenText Documentum CM object.

    When an object link is not needed, you can delete the linked DIR. To delete the linked DIR, run a new CSSAP job. This verifies the existing DIRs, and if there are no linked OpenText Documentum CM objects in OpenText Documentum CM, the DIR is deleted.

    For example, in the following images, the `SAP Query : DeleteLink` (image 1) will be triggered with the action `New Unlink Document Info Record action` (image 2) to delete the DIR link.



Note: For Link OpenText Documentum CM action configuration, SAP system types supported are R/3 and CRM. The CRM SAP object types are Image Assign SAP Workflow and SAP Barcode 4.5.

5.4.3 Automated early archiving using the Agent component

The Agent component can now be used to automate the movement of incoming TIFF images to SAP work items. This is done by configuring a Link Documentum action using parameters similar to the following example:

```
SAP Object Type = Image assign sap workflow
Dctm Query = SelectInvoiceFolders
```

Create a Documentum Query that returns the object name and object ID of the documents to be sent to SAP, for example, “select r_object_id,object_name from sap_invoice where folder('/SAP/Invoices'):”

```
Key.DocumentType=ZFIINVOICE   (enter your custom SAP document type for incoming TIFF
images)
Key.Objecttype=BKPF   (enter your SAP object type)
Key.ArchiveId=Q2
Key.Drl=Drl
```

5.4.3.1 Arbitrary parameters when starting an SAP workflow

Documentum Content Services for SAP Solutions expands the customizing options described previously in “Automated Early Archiving Using the Agent component” similar to the 'Image assign sap workflow' action. When creating a custom BAPI, based on the ARCHIV_PROCESS_RFCINPUT function, define optional parameters to be passed to the DOCUMENT_DATA table parameter.

Add an attribute name to appear in the WebAdmin attribute map to a name on the lines of DOCUMENT_DATA-<*paramname*>; for example, DOCUMENT_DATA-DESCRIPTION, to the “sap_container” attribute. The mapped value is added in the DOCUMENT_DATA table with NAME=<*paramname*>.

For example, the addition of the attribute name, DESCRIPTION, to the following line:

```
<VALUE>Description=DOCUMENT_DATA-DESCRIPTION</VALUE>
```

and the following mapping:

```
"Description" -> "Object ID: %s, r_object_id"
```

adds a line such as the following:

```
NAME=DESCRIPTION, WERT=Object ID: 09f97a8d8000e658
```

to the DOCUMENT_DATA table parameter.

5.5 Checking the integrity of linked documents

After the Agent has automatically linked the SAP and OpenText Documentum CM object, it is technically possible to manually edit or delete DIRs in SAP. It is therefore possible to change/delete links from SAP without receiving a notification of this within OpenText Documentum CM. To help ensure the integrity of the links between SAP and OpenText Documentum CM, it is possible to write a rule to perform an integrity check between the two systems. *Content Services for SAP Solutions does not attempt to fix problems: it just reports them.*

The aim of this action is to generate a report that details any mismatches between OpenText Documentum CM and SAP. The action builds two lists and looks to see whether there is a OpenText Documentum CM object, such as a document, folder, or query, related to each retrieved DIR. In the end, all DIRs that have no OpenText Documentum CM object linked to them are listed in a report. If there is no DIR for a OpenText Documentum CM object, the relation to this object is listed in a report. This is shown in the following illustration:

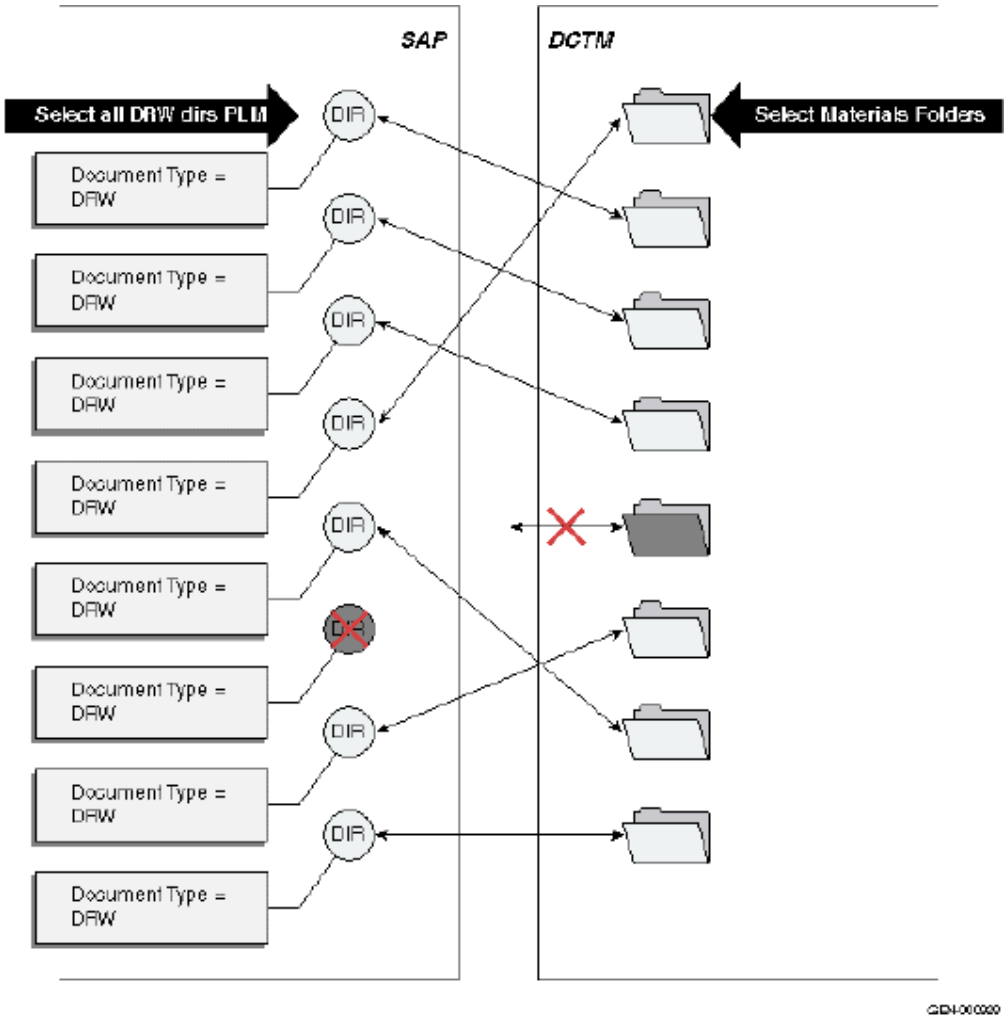


Figure 5-5: Integrity checking

Note: To configure a Check DIR action, you will need a Documentum query and an SAP query, both previously configured.

To check integrity of objects in both systems:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Actions** subnode and select the **Check Document Info Records** subnode.

   The **Check Document Info Records** screen appears.
3. Select **File** > **New** > **Check Document Info Records** from the menu at the top of the Check Document Info Records screen.

The **New Check Document Info Records Action** screen appears.

4. Type an action name in the **Action:** field.
5. Select the Documentum query from the **Documentum Query:** list box.
6. Select the SAP query from the **SAP Query:** list box.
7. Click **OK** to save the DIR check.

5.6 Replication of information between Documentum and SAP

Documentum Content Services for SAP Solutions provides a facility for maintaining the integrity of documents held in SAP with those stored in OpenText Documentum CM.

Replication is the duplication of data held in one system into another system.

The replication process, once started updates all objects not matching the set conditions and not yet updated.

5.6.1 Replicating SAP objects

Replication creates images of SAP objects in OpenText Documentum CM. For example, you may want to replicate invoice information into archived images in OpenText Documentum CM.

You will need to create and configure:

- An SAP Query “Select all pumps PLM” selects materials with the following condition:

```
Description = pump*
```

This query (Query Type: Material by description Product Lifecycle Management) returns the following attributes:

```
Material, Description, Material_External, Material_Guid, Material_Version
```

- OpenText Documentum CM Object Type “dm_folder” is the OpenText Documentum CM Object Type that you want to represent the SAP pumps in the repository.
- Replicate SAP Action “Replicate material folder PLM” has the following rules:

The `Object Key` rules are used to check whether there is a dm_folder object with the path `/SAP/Material` and the `object_name` matching the material attribute of the current pump object in SAP.

`object_name`	`= "%s", Material`
`FOLDER`	`= "/SAP/Material"`

The `Update Condition` rules are used to check whether the attributes of the folder have to be updated (only validated if “update object” is checked). Here you only update the folder if the title of the folder is empty.

The `Map` rules are used to define the mapping of the SAP attributes to the OpenText Documentum CM attributes.

object_name	= "%s", Material
title	= "%s", Description
FOLDER	= "/SAP/Material"

The Replicate SAP action runs “Select all pumps PLM” (SAP Query), returning a number of SAP material objects with their description matching “pump*”.

For each object returned, the Replication action checks whether the dm_folder object already exists. Assuming the attribute Material of the first pump is “Pump One” the query looks such as this:

Select r_object_id from dm_folder where object_name =‘Pump One’and folder (‘/SAP/Material’)

If the folder does not exist (no record returned), the action checks whether it should create one (“create object” is checked). Otherwise it checks the update condition (“update object” is checked). In this case, the action looks for whether the title of the returned dm_folder is empty.

If the folder has to be created or updated, the action maps the SAP attributes to the specified OpenText Documentum CM attributes.

object_name	= "Pump One"
title	= "Electr. Pump"
FOLDER	= "/SAP/Material"

Note: `FOLDER` is a special attribute since it does not really exist. The action creates the dm_folder object in the specified repository folder if a creation is necessary. This loop will be repeated for each object returned by “Select all pumps PLM” (SAP Query), thereby establishing the goal.

In the following example you want to replicate all pumps within SAP as folders in the `/SAP/Material` folder (in the OpenText Documentum CM repository) in order to store additional documents to each pump in that folder. Replicating here means, that the folder should have the same attributes, as the actual SAP object has, in order to be able to search for a specific pump in the repository as well. When the folders are generated you would be able to link them back to the SAP pump objects.

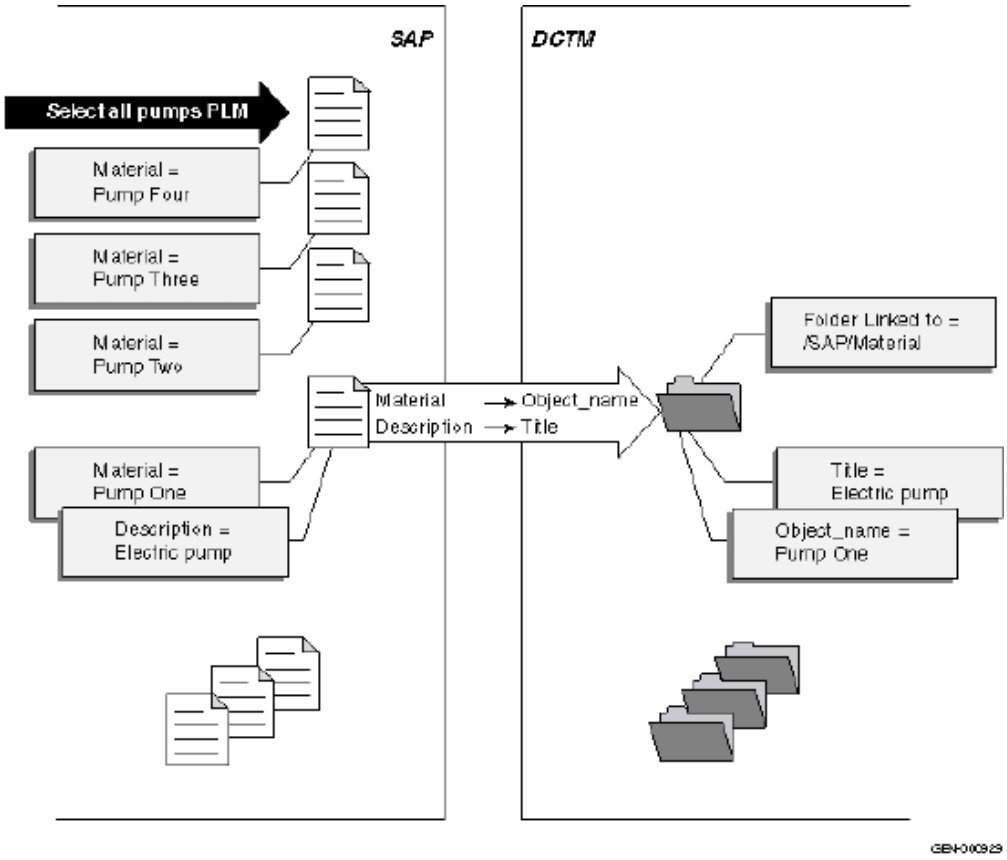


Figure 5-6: OpenText Documentum CM and SAP replication

To configure a Replicate SAP action, you will need a previously configured SAP query.

To replicate SAP objects in OpenText Documentum CM:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Actions** subnode and select the **Replicate SAP** subnode.

   The **Replicate SAP** screen appears.
3. Select **File** > **New** > **Replicate SAP** from the menu at the top of the Replicate SAP screen.

   The **New Replicate SAP Action** screen appears.
4. Type an action name in the **Action:** field.
5. Select the Object type from the **Object Type:** list box.
6. Select the SAP query from the **SAP Query:** list box.

7. **Link Workflow** is set to No Workflow.
8. Select **Update Object**, if required.
9. Select **Create Object**, if required.
10. Select **Extended ECM**, only if you are running the replicate SAP job to connect to xECM for folder creation.
11. For eachObject Key, Update Condition, and Map Rule you want to define:
    a. Select the variable from the *Variable* list box.
    b. Type the format of the variable in the *Format* field.

       The entry is altered with %s automatically added at the end.
    c. Select the parameter required from the list box below the **Parameters** field, and click the up arrow to add the parameter to the **Parameters** field.
    d. Click **Add** to add the rule to the **Object Key**, **Update Condition**, or **Map Rules** fields.
    e. Use the variables prefixed with **"xECM."**. Perform this step only if you selected **Extended ECM**.
12. Click **OK** to save the action.

Note: For Replicate SAP action configuration, SAP system types supported are R/3 and CRM. The CRM SAP object types are Business Partner and Opportunities.

5.6.2 Replicating OpenText Documentum CM objects

In the following example, you want to update the Status of Document Info Record (DIR) attribute in SAP to reflect a change in the status of the OpenText Documentum CM object. Replication in this example means updating SAP objects, not creating new ones. This example can be found as a configuration object in WebAdmin, called Update DIR status. We will execute a Documentum query and update the related DIRs to reflect a change in document status.

You will need to create/configure:

- A Documentum Query "SelectMaterialFolders" selects objects of type dm_folder where the folder's location has to be `/SAP/Material`:

```
1 select r_object_id,object_name from dm_folder
2 where folder('/SAP/Material')
```

- An SAP Object Type: "Document info record PLM" is the SAP Object Type that you want to update in SAP.
- Replicate OpenText Documentum CM Action "Update DIR status" has the following rules:

  The *Object Key* rules are used to check whether there is an SAP object for the current dm_folder with the document number attribute matching the folder's object_ID.

DocumentNumber	= “%s”, r_object_id
DocumentType	= “DRW”
DocumentPart	= “000”
DocumentVersion	= “00”

The *Update Condition* rules are used to check whether the attributes of the SAP object have to be updated (only validated if “update object” is checked). This means we update the DIR only if its status is “WR.”

STATUSEXTERN	= “WR”

The *Map Rules* are used to define the mapping of the OpenText Documentum CM attributes to the SAP attributes.

STATUSEXTERN	= “IW”

The Replicate OpenText Documentum CM action runs “SelectMaterialFolders” (Documentum Query), returning the subfolders of `/SAP/Material`.

For each folder returned, the Replication action checks whether the corresponding Document Info Record Product Lifecycle Management object exists in SAP. If it does, the STATUSEXTERN attribute for the DIR will be checked to see if it is “WR.” If it is, the status will be updated to “IW.”

To replicate OpenText Documentum CM objects in SAP:

1. Connect to WebAdmin. “Logging in to WebAdmin through Documentum Administrator” on page 13 provides information.
2. Click to expand the **Actions** subnode and select the **Replicate Documentum** subnode.

   The **Replicate Documentum** screen appears.
3. Select **File** > **New** > **Replicate Documentum** from the menu at the top of the **Replicate Documentum** screen.

   The **New Replicate Documentum Action** screen appears.
4. Type an action name **Action:** field.
5. Select the SAP object from the **SAP Object:** list box.
6. Select the Documentum query from the **Documentum Query:** list box.
7. **Link Workflow** is set to No Workflow.
8. Select **Update object**, if required.
9. Select **Create object**, if required.
10. For the Object Key, Update Condition, and each Map Rule you want to define:

    a. Select the variable from the **Variables** list box.

b. Type the format of the variable in the **Format** field.

The entry is altered with %s automatically added at the end.

c. Select the parameter required from the list box below the **Parameters** field, and click the up arrow to add the parameter to the **Parameters** field.

d. Click **Add** to add the rule to the **Object Key**, **Update Condition**, or **Map Rules** fields.

11. Click **OK** to save the action.

Note: For Replicating OpenText Documentum CM objects, CRM is not supported and the default SAP System Type is set to SAP R/3.

5.6.3 Working with the FILTER attribute

The FILTER attribute is a symbolic target that specifies an external command line to run when creating links.

The FILTER attribute conforms to the following syntax:

```
<<Path>> <<Arg1>> <<Arg2>> <<ArgN>> <<r_object_ID>> <<Repository>> <<User Name>>
<<Password>>
```

The parameters used in the syntax are described in the following table:

Parameter	Description
Path	Fully qualified path to an executable.
Arg1, Arg2,...ArgN	Arbitrary parameters as defined by your filter program.
r_object_id	r_object_id of the current object.
Repository	Name of the repository where you would like to run the executable.
Username	Username used to connect to the repository.
Password	Password that corresponds to the username described in this table.

You can use these parameters to pass additional values to the external filter; use the printf()–format string for this purpose.

If the execution is successful, the external filter returns 0 as the exit code; if unsuccessful, it returns a non–zero value as the exit code.

5.7 Unlink Document Info Record action

Unlink Document Info Record (DIR) action allows you to remove the Document Info Record in SAP and OpenText Documentum CM for linked SAP Product Lifecycle Management documents. This action searches a OpenText Documentum CM object, such as, document, folder, or query related to each retrieved Document Info Record. All the Document Info Records that have OpenText Documentum CM object linked to them are removed in SAP. If there is no Document Info Record for a OpenText Documentum CM object then no action is taken in SAP or OpenText Documentum CM.

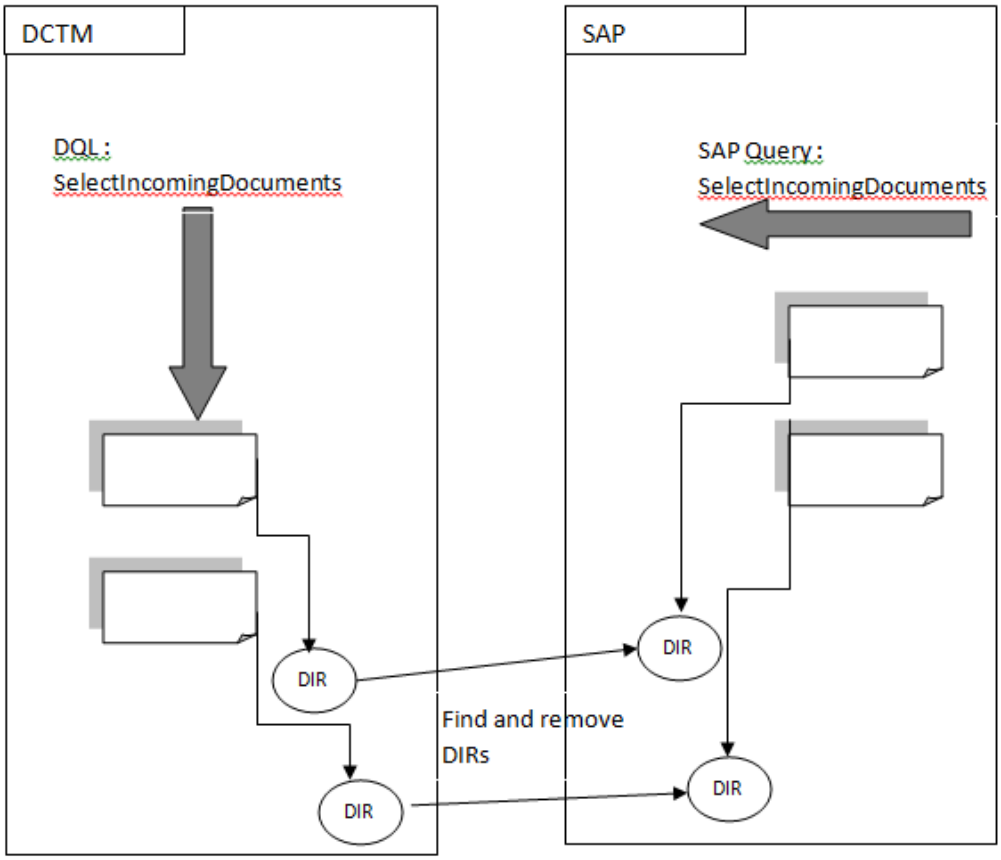


Figure 5-7: Unlink Document Info Record (DIR) action

To Unlink the Document Info Record of objects:

Note: To configure an Unlink Document Info Record action, create and configure a Documentum query and an SAP query.

1. Connect to WebAdmin. For details, see "Logging in to WebAdmin through Documentum Administrator" on page 13.

2. Click to expand the **Actions** subnode and select the **Unlink Document Info Record action** subnode.

   The **Unlink Document Info Record action** screen appears.

3. From the menu, select **File** > **New** > **Unlink Document Info Record action**.

   The **New Unlink Document Info Record Action** screen appears.

4. In the **Action** field, type an action name.

5. From **Documentum Query**, select the Documentum query.

6. From **SAP Query**, select the SAP query.

7. To define the rules, use **Map Rule Composer**.

   a. From **Variables**, select the variable.

   b. In the Format field, enter the format of the variable.

   c. Select the required parameter from the list and click the up arrow to add the parameter to the **Parameters** field.

      The attribute map allows you to specify the following:

Attribute	Description
Document Info Record (DIR) attributes	When a document is released to SAP, an SAP DIR is created for it. You can set the values for **DocumentType**, **Description**, and **DocumentNumber** For example, the rule *Document Description = "Document for %s"*, Material builds a DIR description containing the Material attribute from the SAP object returned by the SAP query chosen.
$ARG#s	To link a SAP object to a OpenText Documentum CM object, Content Services finds the related object in the repository. Specify the substitution for an $ARG# in the DQL statement of the Documentum query. This defines a lookup into OpenText Documentum CM which identifies the object to link to the SAP object.

   d. Click the down arrow to add the rule to **Defined Map Rules**.

8. Click **OK** to save the Unlink Document Info Record configuration.

Note: For Unlink Document Info Record action, CRM is not supported and the default SAP System Type is set to SAP R/3. This Unlink Document Info Record action works for all related Product Lifecycle Management SAP queries.

5.8 Customization of Document Management System attributes in OpenText Documentum CM using the custom installer

Content Services for SAP Solutions allows you to customize Document Management System attributes in OpenText Documentum CM using the custom installer.

5.8.1 Installing the dmei_custom_installer

To install and execute the dmei_custom_installer:

1. Extract the contents of the dmei_custom_installer.zip or .tar file to a local directory.
2. Edit the `dfc.properties` file in the previous directory to point to the connection broker.
3. Create or copy the `custom.xml` file with the required customizations to the local directory.
4. Edit the `installer.properties` file for the following parameters:

   Table 5-2: Parameters in installer.properties

Parameter name	Value
docbase.name	<repository name>
docbase.user.name	<repository admin username>
domain	<domain>
custom.xml.path	<absolute path to the custom.xml file>

   An example of installer.properties file:

   ```
   user.language=
   docbase.name=myrepo
   docbase.user.name=Administrator
   domain=
   custom.xml.path=C:\\CSSAP\\custom_installer\\custom.xml
   ```

   Note: Change the switch of custom.xml.path from \\ to \ depending on the Operating System .

5. The custom.installer log is based on the filename specified in the log4j2.properties present in the same directory.
6. Execute the custom installer using the following command:

   ```
   java -jar dmei_custom_installer.jar
   ```

5.8.2 Custom.xml

The `custom.xml` file is used to customize the Documentum Content Services for SAP Solutions based on Document Management System attributes in SAP for any new objects or SAP object types that are not present in OpenText Documentum CM already.

Note: Only Replicate SAP action is supported for SAP CRM.

The following are some of the sample custom.xml files:

5.8.2.1 Purchase requisitions custom.xml

You can write the following custom.xml file for Purchase requisitions based on your requirement. The table name in SAP for Purchase requisition is EBAN. The SAP query we intend to filter is based on the Client, PR Number and the item number. The result of the SAP query that we intend to see is Client, PR Number, Item Number and Short text. The key attributes for a PR is the PR Number and the Item number.

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort" NOTE="put your own methods inside this request">
        <OBJECT_CREATE ON_EXIST="version">
        <API_CONFIG TYPE="sap_query_plm_type_table" CLASS="sap">
                <ATTRIBUTE NAME="object_name" IS_KEY="true">Purchase Requisitions_PLM</
ATTRIBUTE>
                <ATTRIBUTE NAME="table_name">EBAN</ATTRIBUTE>
                <ATTRIBUTE NAME="function_module">RFC_READ_TABLE</ATTRIBUTE>
                <ATTRIBUTE NAME="sap_object_type">EBAN</ATTRIBUTE>
                <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                        <VALUE>Client=MANDT</VALUE>
                        <VALUE>PR_Number=BANFN</VALUE>
                        <VALUE>PR_Item_Number=BNFPO</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="parameter_defaults" IS_REPEATING="true"></ATTRIBUTE>
                <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                        <VALUE>Client=MANDT</VALUE>
                        <VALUE>PR_Number=BANFN</VALUE>
                        <VALUE>Short text=TXZ01</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                        <VALUE>PR_Number=BANFN</VALUE>
                        <VALUE>PR_Item_Number=BNFPO</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                        <VALUE>Link</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="descriptive_field">TXZ01</ATTRIBUTE>
        </API_CONFIG>
        </OBJECT_CREATE>
</REQUEST>
```

5.8.2.2 Purchase order custom.xml

Similarly, the following is a custom.xml file for Purchase orders

```
<?xml version="1.0"?>
        <OBJECT_CREATE ON_EXIST="version">
        <API_CONFIG TYPE="sap_query_plm_type_table" CLASS="sap">
                <ATTRIBUTE NAME="object_name" IS_KEY="true">Purchase Orders_PLM</
ATTRIBUTE>
                <ATTRIBUTE NAME="table_name">EKPO</ATTRIBUTE>
                <ATTRIBUTE NAME="function_module">RFC_READ_TABLE</ATTRIBUTE>
                <ATTRIBUTE NAME="sap_object_type">EKPO</ATTRIBUTE>
                <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                        <VALUE>Client=MANDT</VALUE>
                        <VALUE>PO_Number=EBELN</VALUE>
                        <VALUE>Item_Number=EBELP</VALUE>
                        <VALUE>$DQL=</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="parameter_defaults" IS_REPEATING="true"></ATTRIBUTE>
                <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                        <VALUE>Client=MANDT</VALUE>
                        <VALUE>PO_Number=EBELN</VALUE>
                        <VALUE>Item_Number=EBELP</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                        <VALUE>PO_Number=EBELN</VALUE>
                        <VALUE>Item_Number=EBELP</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                        <VALUE>Link</VALUE>
                </ATTRIBUTE>
                <ATTRIBUTE NAME="descriptive_field">EBELP</ATTRIBUTE>
        </API_CONFIG>
        </OBJECT_CREATE>
</REQUEST>
```

5.8.2.3 REVLEVEL custom.xml

sample custom.xml file with REVLEVEL

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort">
        <OBJECT_CREATE ON_EXIST="version">
                <API_CONFIG TYPE="sap_query_type_plm" CLASS="sap">
                        <ATTRIBUTE NAME="object_name" IS_KEY="true">Document Info Record
PLM</ATTRIBUTE>
                        <ATTRIBUTE NAME="function_module_create">BAPI_DOCUMENT_CREATE</
ATTRIBUTE>
                        <ATTRIBUTE NAME="function_module_update">BAPI_DOCUMENT_CHANGE2</
ATTRIBUTE>
                        <ATTRIBUTE NAME="sap_object_type">Document Info Record PLM</
ATTRIBUTE>
                        <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">

<VALUE>DocumentType=DOCUMENTDATA.DOCUMENTTYPE,3</VALUE>

<VALUE>Description=DOCUMENTDATA.DESCRIPTION,40</VALUE>

<VALUE>DocumentNumber=DOCUMENTDATA.DOCUMENTNUMBER,25</VALUE>

<VALUE>DocumentVersion=DOCUMENTDATA.DOCUMENTVERSION,2</VALUE>

<VALUE>DocumentPart=DOCUMENTDATA.DOCUMENTPART,3</VALUE>

<VALUE>DataCarrier1=DOCUMENTDATA.DATACARRIER1,10</VALUE>

<VALUE>WSApplication1=DOCUMENTDATA.WSAPPLICATION1,3</VALUE>
```

```
<VALUE>DocFile1=DOCUMENTDATA.DOCFILE1,255</VALUE>

<VALUE>STATUSEXTERN=DOCUMENTDATA.STATUSEXTERN,2</VALUE>

<VALUE>USERDEFINED1=DOCUMENTDATA.USERDEFINED1,14</VALUE>

<VALUE>USERDEFINED2=DOCUMENTDATA.USERDEFINED2,14</VALUE>

<VALUE>USERDEFINED3=DOCUMENTDATA.USERDEFINED3,14</VALUE>

<VALUE>USERDEFINED4=DOCUMENTDATA.USERDEFINED4,14</VALUE>

<VALUE>Laboratory=DOCUMENTDATA.LABORATORY,3</VALUE>
                                        <VALUE>HostName=HOSTNAME,20</VALUE>
                                        <VALUE>Revlevel=DOCUMENTDATA.REVLEVEL,2</
VALUE>
                                        <VALUE>ECNumber=DOCUMENTDATA.ECNUMBER,
12</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="parameter_defaults" IS_REPEATING="true">
                                        <VALUE>DocumentType=DRW</VALUE>
                                        <VALUE>DocumentVersion=00</VALUE>
                                        <VALUE>DocumentPart=000</VALUE>
                                        <VALUE>DataCarrier1=DOCUMENTUM</VALUE>
                                        <VALUE>WSApplication1=DCM</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                                        <VALUE>DocumentNumb=DOCNUMBER,25</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                                        <VALUE>DocumentType=DOCUMENTTYPE,0,3</
VALUE>
                                        <VALUE>DocumentNumber=DOCUMENTNUMBER,
3,25</VALUE>
                                        <VALUE>DocumentVersion=DOCUMENTVERSION,
28,2</VALUE>
                                        <VALUE>DocumentPart=DOCUMENTPART,30,3</
VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                                        <VALUE>Create</VALUE>
                                        <VALUE>Update</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="result_table">DOCUMENTSTRUCTURE,64</ATTRIBUTE>
                </API_CONFIG>
        </OBJECT_CREATE>
</REQUEST>
```

5.8.2.4 CRM based SAP Object and Query types custom.xml

Custom.xml for CRM based SAP Object and Query types. The following custom.xml is for Opportunities specific to CRM system.

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort" NOTE="put your own methods inside this request">
        <OBJECT_CREATE ON_EXIST="version">
        <API_CONFIG TYPE="sap_query_plm_type_table" CLASS="sap">
                        <ATTRIBUTE NAME="object_name" IS_KEY="true">Opportunities</
ATTRIBUTE>
                        <ATTRIBUTE NAME="table_name">CRMD_OPPORT_H</ATTRIBUTE>
                        <ATTRIBUTE NAME="function_module">RFC_READ_TABLE</ATTRIBUTE>
                        <ATTRIBUTE NAME="sap_object_type">Opportunities</ATTRIBUTE>
                        <ATTRIBUTE NAME="sap_system_type" IS_REPEATING="true">
                                        <VALUE>CRM</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                                        <VALUE>Client=CLIENT</VALUE>
                                        <VALUE>GuiID=GUID</VALUE>
                                        <VALUE>Oppurtunity_desc=DESCRIPTION_OPP</
```

```
VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="parameter_defaults" IS_REPEATING="true"></
ATTRIBUTE>
                        <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                                        <VALUE>Client=CLIENT</VALUE>
                                        <VALUE>GuiID=GUID</VALUE>
                                        <VALUE>Oppurtunity_desc=DESCRIPTION_OPP</
VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                                        <VALUE>GuiID=GUID</VALUE>
                        </ATTRIBUTE>
                        <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                                        <VALUE>Link</VALUE>
                        </ATTRIBUTE>
                </API_CONFIG>
                </OBJECT_CREATE>
</REQUEST>
```

If you want the Opportunities SAP Object type to be displayed for both R/3 and CRM you could add:

```
<ATTRIBUTE NAME="sap_system_type" IS_REPEATING="true">
<VALUE>R/3</VALUE>
<VALUE>CRM</VALUE>
</ATTRIBUTE>
```

Note: If sap_system_type is not specified in the custom.xml, the SAP Object type is defaulted to R/3.

5.8.3 Configuring classification attributes for sap_query_type_plm query types

Content Services for SAP Solutions supports custom Document Management System classification attributes. Custom Document Management System classification attributes can be set for the CLASSIFICATIONVALUES and CLASSALLOCATIONS table parameters of BAPI_DOCUMENT_CHANGE/CREATE.

Note: The custom.xml is supported only on SAP R/3 systems.

To configure classification values for sap_query_type_plm query types:

1. Browse to the directory where you extracted the contents of the Documentum Content Services for SAP Solutions installer archive, and open the custom.xml file for editing.
2. In the custom.xml file, use the following convention to configure the sap_query_type_plm query type to the corresponding custom Document Management System classification attributes:

   ```
   <VALUE><<VariableName>>=CHARACTERISTICVALUES.<<CLASSTYPE>>.<<CLASSNAME>>.<<CHARACTER
   ISTICNAME>>.<<DELETIONFLAG>><|VALUE>
   ```

   For example, you can set classification values and class allocations as follows:

   ```
   <VALUE>Instruction=CHARACTERISTICVALUES.017.SPEC_APPEARANCE.
   ```

```
CHARNAME_INSTRUCTION.0</VALUE>
```

A sample sap_query_type_plm definition would be as follows:

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort">
<OBJECT_CREATE ON_EXIST="version">
                <API_CONFIG TYPE="sap_query_type_plm" CLASS="sap">
                            <ATTRIBUTE NAME="object_name" IS_KEY="true">Document
Info Record

                          PLM</ATTRIBUTE>
                            <ATTRIBUTE
NAME="function_module_create">BAPI_DOCUMENT_CREATE

                            </ATTRIBUTE>
                            <ATTRIBUTE
NAME="function_module_update">BAPI_DOCUMENT_CHANGE

                            </ATTRIBUTE>
                            <ATTRIBUTE NAME="sap_object_type">Document Info Record
PLM</ATTRIBUTE>
                            <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                                    <VALUE>DocumentType=DOCUMENTDATA.DOCUMENTTYPE,
3</VALUE>
                                    <VALUE>Description=DOCUMENTDATA.DESCRIPTION,40</
VALUE>

<VALUE>DocumentNumber=DOCUMENTDATA.DOCUMENTNUMBER,25</VALUE>

<VALUE>DocumentVersion=DOCUMENTDATA.DOCUMENTVERSION,2</VALUE>
                                    <VALUE>DocumentPart=DOCUMENTDATA.DOCUMENTPART,
3</VALUE>

<VALUE>DataCarrier1=DOCUMENTDATA.DATACARRIER1,10</VALUE>

<VALUE>WSApplication1=DOCUMENTDATA.WSAPPLICATION1,3</VALUE>
                                    <VALUE>DocFile1=DOCUMENTDATA.ORIGINAL,255</
VALUE>
                                    <VALUE>STATUSEXTERN=DOCUMENTDATA.STATUSEXTERN,
2</VALUE>

<VALUE>USERDEFINED1=DOCUMENTDATA.USERDEFINED1,14</VALUE>

<VALUE>USERDEFINED2=DOCUMENTDATA.USERDEFINED2,14</VALUE>

<VALUE>USERDEFINED3=DOCUMENTDATA.USERDEFINED3,14</VALUE>

<VALUE>USERDEFINED4=DOCUMENTDATA.USERDEFINED4,14</VALUE>
                                    <VALUE>Laboratory=DOCUMENTDATA.LABORATORY,3</
VALUE>
                                    <VALUE>HostName=HOSTNAME,20</VALUE>
                                    <VALUE>Color=CHARACTERISTICVALUES.
017.SPEC_APPEARANCE.

                                                CHARNAME_COLOR.0</VALUE>
                                    <VALUE>Instruction=CHARACTERISTICVALUES.
017.SPEC_APPEARANCE.
                               CHARNAME_INSTRUCTION.0</VALUE>
                              </ATTRIBUTE>
                            <ATTRIBUTE NAME="parameter_defaults"
IS_REPEATING="true">
                                    <VALUE>DocumentType=DRW</VALUE>
                                    <VALUE>DocumentVersion=00</VALUE>
                                    <VALUE>DocumentPart=000</VALUE>
                                    <VALUE>DataCarrier1=DOCUMENTUM</VALUE>
                                    <VALUE>WSApplication1=DCM</VALUE>
                            </ATTRIBUTE>
                            <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                                    <VALUE>DocumentNumb=DOCNUMBER,25</VALUE>
                              </ATTRIBUTE>
```

```
                                <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                                        <VALUE>DocumentType=DOCUMENTTYPE,0,3</VALUE>
                                        <VALUE>DocumentNumber=DOCUMENTNUMBER,3,25</
VALUE>
                                        <VALUE>DocumentPart=DOCUMENTPART,30,3</VALUE>
                                </ATTRIBUTE>
                                <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                                        <VALUE>Create</VALUE>
                                        <VALUE>Update</VALUE>
                                </ATTRIBUTE>
                                <ATTRIBUTE NAME="result_table">DOCUMENTSTRUCTURE,64</
ATTRIBUTE>
                        </API_CONFIG>
</OBJECT_CREATE>
</REQUEST>
```

3. Extract the contents of dmei_custom_installer.zip into a temporary folder.
4. Locate the dfc.properties file in the temporary folder and edit the file with the OpenText Documentum Content Management (CM)Foundation Java API directory and connection broker information (dfc.docbroker.host).
5. Locate the installer.properties file in the temporary folder and edit the file with the following:

Value	Definition
user.language	The default is en_US (English).
docbase.name	Name of the repository that you want to configure.
docbase.user.name	Type repository install owner name.
domain	Name of the domain in which the repository is located.
custom.xml.path	The custom XML file path.

6. Locate the log4j2.properties file in the temporary folder and edit to change the directory location of the log files.

   Note: By default, log files are generated in the current (temporary) folder.

7. Run the following command from the command line to run the custom XML file installer:

```
java -jar dmei_custom_installer.jar
```

   Note: The file dmei_custom_installer.jar is in the temporary folder where you extracted the contents of the dmei_custom_installer.zip file.

8. Type the password for the repository specified in the installer.properties file.

   File processing begins and the following message appears when processing completes:

```
Processing Successfully Completed!
```

   You have configured classification values for sap_query_type_plm query types; Content Services for SAP Solutions can now replicate custom Document

Management System attributes of a document from OpenText Documentum CM to SAP.

The classification values that you configured here are accessible from the Rule Composer section of the **Link Documentum** and **Replicate Documentum** tabs in WebAdmin.

Note: If the Processing Failed! message appears, look in the log files to determine the problem with the custom.xml file installation.

5.8.4 Replicating custom Document Management System attributes to SAP custom tables

Content Services for SAP Solutions supports replicating Document Management System classification attributes from OpenText Documentum CM to SAP custom tables.

To replicate custom Document Management System attributes to SAP custom tables:

1. Start SAP GUI and connect to an SAP R/3 system.
2. In the command field, execute the /se80 transaction code.
3. Use the options available in the **Object Navigator** page to define a custom SAP table.

   Definition of a sample custom SAP table is as follows:

Field	Element	Type	Length	Description
MANDT	MANDT	CLNT	3	Client
DOKAR	DOKAR	CHAR	3	Document Type
DOKNR	DOKNR	CHAR	25	Document Number
DOKTL	DOKTL_D	CHAR	3	Document Part
DOKVR	DOKVR	CHAR	2	Document Version
DOKDSR	DOKDSR	CHAR	25	Document Description

4. Browse to the directory where you extracted the contents of 's installer archive, and open the custom.xml file for editing.
5. In the custom.xml file, specify entries that correspond to the definition of the custom table you created in step 2.

   The emphasized portion of the following sample snippet (from a custom.xml file) indicates how the entries in the custom.xml file correspond with the definition of the sample custom SAP table shown in step 2:

```
<?xml version="1.0"?>
<REQUEST ON_ERROR="abort">
<OBJECT_CREATE ON_EXIST="version">
               <API_CONFIG TYPE="sap_query_type_plm" CLASS="sap">
                           <ATTRIBUTE NAME="object_name" IS_KEY="true">Document
Info Record

                           PLM</ATTRIBUTE>
                           <ATTRIBUTE
NAME="function_module_create">BAPI_DOCUMENT_CREATE

                           </ATTRIBUTE>
                           <ATTRIBUTE
NAME="function_module_update">BAPI_DOCUMENT_CHANGE

                           </ATTRIBUTE>
                           <ATTRIBUTE NAME="sap_object_type">Document Info Record
PLM</ATTRIBUTE>
                           <ATTRIBUTE NAME="query_parameters" IS_REPEATING="true">
                                   <VALUE>DocumentType=DOCUMENTDATA.DOCUMENTTYPE,
3</VALUE>
                                   <VALUE>Description=DOCUMENTDATA.DESCRIPTION,40</
VALUE>

<VALUE>DocumentNumber=DOCUMENTDATA.DOCUMENTNUMBER,25</VALUE>

<VALUE>DocumentVersion=DOCUMENTDATA.DOCUMENTVERSION,2</VALUE>
                                   <VALUE>DocumentPart=DOCUMENTDATA.DOCUMENTPART,
3</VALUE>

<VALUE>DataCarrier1=DOCUMENTDATA.DATACARRIER1,10</VALUE>

<VALUE>WSApplication1=DOCUMENTDATA.WSAPPLICATION1,3</VALUE>
                                   <VALUE>DocFile1=DOCUMENTDATA.ORIGINAL,255</
VALUE>
                                   <VALUE>STATUSEXTERN=DOCUMENTDATA.STATUSEXTERN,
2</VALUE>

<VALUE>USERDEFINED1=DOCUMENTDATA.USERDEFINED1,14</VALUE>

<VALUE>USERDEFINED2=DOCUMENTDATA.USERDEFINED2,14</VALUE>

<VALUE>USERDEFINED3=DOCUMENTDATA.USERDEFINED3,14</VALUE>

<VALUE>USERDEFINED4=DOCUMENTDATA.USERDEFINED4,14</VALUE>
                                   <VALUE>Laboratory=DOCUMENTDATA.LABORATORY,3</
VALUE>
                                   <VALUE>HostName=HOSTNAME,20</VALUE>
                                   <VALUE>Color=CHARACTERISTICVALUES.
017.SPEC_APPEARANCE.

                                              CHARNAME_COLOR.0</VALUE>
                                   <VALUE>Instruction=CHARACTERISTICVALUES.
017.SPEC_APPEARANCE.

                                              CHARNAME_INSTRUCTION.0</VALUE>
                              <VALUE>Z_CustomBapiClient=ZCUSTOM_BAPI.MANDT,
                                        3</VALUE>
                               <VALUE>Z_CustomBapiDocuType=ZCUSTOM_BAPI.DOKAR,3</VALUE>
                               <VALUE>Z_CustomBapiDocuNumber=ZCUSTOM_BAPI.DOKNR,25</
VALUE>
                               <VALUE>Z_CustomBapiDocuPart=ZCUSTOM_BAPI.DOKTL,3</VALUE>
                               <VALUE>Z_CustomBapiDocuVersion=ZCUSTOM_BAPI.DOKVR,2</
VALUE>

<VALUE>Z_CustomBapiSAMPLE32=ZCUSTOM_BAPI.ZDM_DMS_SAMPLE32,
                                        32</VALUE>

<VALUE>Z_CustomBapiSAMPLE255=ZCUSTOM_BAPI.ZDM_DMS_SAMPLE25,
                                        255</VALUE>
```

```
                              </ATTRIBUTE>
                            <ATTRIBUTE NAME="parameter_defaults"
IS_REPEATING="true">
                                      <VALUE>DocumentType=DRW</VALUE>
                                      <VALUE>DocumentVersion=OO</VALUE>
                                      <VALUE>DocumentPart=OOO</VALUE>
                                      <VALUE>DataCarrier1=DOCUMENTUM</VALUE>
                                      <VALUE>WSApplication1=DCM</VALUE>
                              </ATTRIBUTE>
                            <ATTRIBUTE NAME="result_parameters" IS_REPEATING="true">
                                      <VALUE>DocumentNumb=DOCNUMBER,25</VALUE>
                              </ATTRIBUTE>
                              <ATTRIBUTE NAME="key_attributes" IS_REPEATING="true">
                                      <VALUE>DocumentType=DOCUMENTTYPE,0,3</VALUE>
                                      <VALUE>DocumentNumber=DOCUMENTNUMBER,3,25</
VALUE>
                                      <VALUE>DocumentPart=DOCUMENTPART,30,3</VALUE>
                            </ATTRIBUTE>
                            <ATTRIBUTE NAME="methods" IS_REPEATING="true">
                                      <VALUE>Create</VALUE>
                                      <VALUE>Update</VALUE>
                              </ATTRIBUTE>
                              <ATTRIBUTE NAME="result_table">DOCUMENTSTRUCTURE,64</
ATTRIBUTE>
                  </API_CONFIG>
</OBJECT_CREATE>
</REQUEST>
```

6. Extract the contents of dmei_custom_installer.zip into a temporary folder.
7. Locate the dfc.properties file in the temporary folder and edit the file with the Foundation Java API directory and connection broker information (dfc.docbroker.host).
8. Locate the installer.properties file in the temporary folder and edit the file with the following:

Value	Definition
user.language	The default is en_US (English).
docbase.name	Name of the repository that you want to configure.
docbase.user.name	Type repository install owner name.
domain	Name of the domain in which the repository is located.
custom.xml.path	The custom XML file path.

9. Locate the log4j2.properties file in the temporary folder and edit to change the directory location of the log files.

Note: By default, log files are generated in the current (temporary) folder.

10. Run the following command from the command line to run the custom XML file installer:

```
java -jar dmei_custom_installer.jar
```

Note: The `dmei_custom_installer.jar` file is in the temporary folder where you extracted the contents of the `dmei_custom_installer.zip` file.

11. Type the password for the repository specified in the installer.properties file.

    File processing begins and the message Processing Successfully Completed! appears when processing completes.

    Note: If the following message appears, look in the log files to determine the problem with the custom.xml file installation:

    ```
    Processing Failed!
    ```

    Documentum Content Services for SAP Solutions is now configured to replicate custom Document Management System attributes of a document from OpenText Documentum CM to SAP custom tables.

    The classification values that you configured here are accessible from the **Rule Composer** section of the **Link Documentum** and **Replicate Documentum** tabs in WebAdmin.

    Notes

    - Special processing rules are defined for the following attributes:
      - DOCUMENT_TYPE
      - DOCUMENT_NUMBER
      - DOCUMENT_PART
      - DOCUMENT_VERSION

      The values for these attributes can be set to the corresponding values for the SAP DIR object that is created or updated. Invoke the special processing rules by setting the following symbolic values:

Attribute	Symbolic value
Document Type	@DOCTYPE
Document Number	@DOCNUMBER
Document Part	@DOCPART
Document Version	@DOCVERSION

      Depending on your requirements, set these symbolic values in the Rule Composer section of the **Link Documentum** and **Replicate Documentum** tabs in WebAdmin.

    - To use the zcustom BAPI, map the zcustom BAPI and zcustom table name using the rule composer in WebAdmin. The zcustom table name is the name of the parameter for the zcustom BAPI and not the name of the table itself.

5.9 Using Auto Manage to execute Documentum Content Services for SAP Solutions actions

Jobs can be made to run automatically at regular intervals by creating an Agent to run a defined action:

- Jobs run Agents.
- Agents execute actions.
- Actions perform linking, replication, or integrity checking.

Job progress and status can be monitored.

The following illustration shows the relationship between these parts.

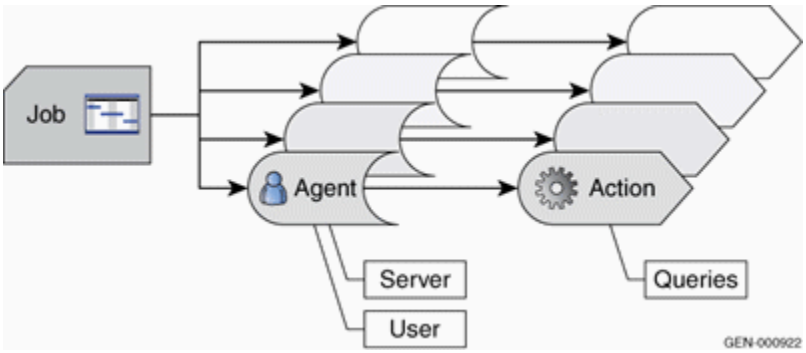


Figure 5-8: Agent services

5.9.1 Creating, viewing, and editing an Agent

Agents run actions and define on which server and what user an action will run. If an action was meant to run on several servers, each server has to have an Agent configured. You should be careful choosing an SAP user since the Agent will run the action with the access rights this user has on the SAP Server defined in the Agent.

To configure an Agent, you will need a previously configured Action as well as a previously defined SAP Server and SAP user.

To create, view, or edit an Agent:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Auto Manage** subnode and select the **Agents** subnode.

   The **Agents** screen appears.
3. Select **File** > **New** > **Agent** from the menu at the top of the Agent screen.

   The **New Agent** screen appears.

4. Enter a name for the Agent in the **New Agent Name:** field.
5. Select the SAP system type from the **SAP System Type:** list box.
6. Select the action required by the Agent from the **Action:** list box.
7. Select the SAP server where the Agent is running from the **SAP Server:** list box.
8. Select the SAP user with the rights to run the Agent from the **SAP User:** list box.
9. Click **OK** to save the Agent configuration.

5.9.2 Creating, viewing, and editing HVPS

The following sections detail how to register and unregister an HVP Worker.

5.9.2.1 Registering an HVP Worker

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Auto Manage** subnode and select the **HVPS** subnode.

   The **HVPS** screen appears.
3. Select **File** > **New** > **Register HVP Worker** from the menu at the top of the HVPS screen.

   The **New Register HVPS Worker** screen appears.
4. Enter a name for the Worker in the **Name:** field.
5. Enter the web address for the Worker in the **Worker URL:** field as:

   ```
   http://<host>:<port>/HvpWorker/hvpCommand/
   ```

   or

   ```
   http://<host>:<port>/HvpWorker/hvpCommand
   ```
6. Select **Is Available:** if required.
7. Click **OK** to save the Worker registration information.

5.9.2.2 Unregistering an HVP Worker

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Auto Manage** subnode and select the **HVPS** subnode.

   The **HVPS** screen appears.
3. Select the HVP Worker you want to unregister from the **HVPS** screen.

   The **HVP Worker confirm delete** screen appears.
4. Click **OK** to unregister the Worker.

The HVP Worker is removed from the **HVPS** screen.

Note: The HVP Worker can also be rendered unavailable by deselecting **Is Available** on **HVPS properties** screen.

5.9.3 Creating, viewing, and editing Documentum Content Services for SAP Solutions jobs

A job can be scheduled to start on a defined date (Activation Date) and to expire on a defined date (Expiration Date). The format of the date is "day, month, year" and the format of the time is "hours: minutes: seconds."

The frequency of invoking the job can be defined with an interval. The interval consists of a number and a measurement. The following measurements are available:

- Minute(s)
- Hour(s)
- Day(s)
- Week(s)
- Month(s)
- Years(s)
- Day of the week
- Day of the month
- Day of the year

There can be a number of runs defined, deactivating the job after having completed the defined number of runs. If 0 is defined, the job runs with no limitations.

To configure a Job, you will need a previously configured Agent.

To create, view, or edit jobs:

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.
2. Click to expand the **Auto Manage** subnode and select the **Jobs** subnode.

   The **Jobs** screen appears.
3. Select **File** > **New** > **SAP Job** from the menu at the top of the **Jobs** screen.

   The **New Job** screen appears with four tabs across the top.
4. In the *Info* tab:
   a. Enter a job name in the **Name:** field.
   b. Enter a job type in the **Job Type:** field.

c. Select a trace level for the job in the **Trace Level:** list box. The trace level detail depends on the log level set in the hvp.properties of the HVP Worker. Set hvp.log.level=DEBUG in hvp.properties for detailed log.

d. Select **Active** or **Inactive**, if required.

e. Select **Deactivate upon failure, Run after Update, or Save if invalid,** if required.

f. Click **Next** or **Schedule** to continue the job configuration.

5. In the **Schedule** tab:

a. Select a job start date and time with the **Start Date And Time:** calendar pop-up and time list boxes.

b. Select a metric for job to repeat in the **Repeat:** list box.

c. Enter how often the job will repeat in the **Frequency:** list box.

d. Select a job end date and time with the **End Date And Time:** calendar pop-up and time list boxes, or enter a specific number of time for the job to run.

e. Click **Next** to continue the job configuration.

6. In the **SAP Job** tab:

a. Select job agents from the **Agents:** list box.

b. Click **Add** to add the agent to the **Agents to Run** field.

Use the up and down arrows to adjust the order that agents run.

7. Click **Finish** to save the Job configuration.

The newly created job appears in the **Jobs** screen.

5.9.4 Performing job maintenance

1. Connect to WebAdmin. "Logging in to WebAdmin through Documentum Administrator" on page 13 provides information.

2. Click to expand the **Auto Manage** subnode and select the **Jobs** subnode.

The **Jobs** screen appears showing the job **Object Name**, job **Last Run**, current **State**, and **Status** of the job.

3. Right-click on the desired job and select:

Properties – To make adjustments to the job settings.

Run – To immediately run the job.

Refresh – To refresh the Jobs screen.

View Trace File – To view a report based on the trace level set for the job run.

Delete – To delete the job and all its settings.

The *OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD)* and *OpenText Documentum Content Management - Server*

System Object Reference Guide (EDCCS250400-ORD) have more information on job status attributes [dm_job].

5.10 Monitoring Documentum Content Services for SAP Solutions logging

You can monitor Documentum Content Services for SAP Solutions logging through the job logs and HVP Worker logs.

5.10.1 Job logs

To view the SAP job logs executed from the WebAdmin:

- Right-click on the desired job and select View Trace File to view the job trace files.

  The job trace files are stored at /System/HVPS/JOBS in the repository. The trace files are not versioned. All will be seen as 1.0 CURRENT.

- For HVP Controller logs, select the controller from the component's log list box to view the HVP controller log. The controller log specifies the version information for Enterprise Integration Core DAR and HVPS DAR and details the availability of the HVP Workers.

  The HVP Controller logs are stored at `/System/HVPS/JOBS` in the repository. The logs are not versioned. All will be seen as 1.0 CURRENT.

- For Job logs, select the agent name from the component's log list box.

5.10.2 HVP Worker logging

The HVP Worker log is present where the HVP Worker is running. The path for `hvpWorker.log` file can be set in the `log4j2.properties` file. Change the `log4j2.logger.com.documentum.ei=INFO` to `DEBUG` if you need detailed logs for troubleshooting.

The trace level for the HVP Worker can be set in the `hvp.properties` file. The `hvp.properties` file contains information about the job execution logs:

```
hvp.log.base=C:\Documentum\logs
hvp.log.format=%d{DATE} [%-5p] [%t] %c - %m%n
hvp.log.level=INFO
hvp.log.retain=retainAlways
hvp.job.copyOnRepSapAction=false
hvp.job.encryptSapUser=false
```

Property	Description
`hvp.log.base`	Provides information about the log path. The log file that is generated is created in the "job name" folder. For example, execution of a job named "myjob" would create the log files in the `C:\Documentum\logs\myjob`.

Property	Description
hvp.log.format	Provides information about the logger statements format. For example, %d{DATE} [%-5p] [%t] %c - %m%n can result in a logger statement such as "INFO 14 Apr 2009 11:05:20,078 Request executed".
hvp.log.level	Indicates the log level of the logs generated. Set hvp.log.level=DEBUG for detailed logs.
hvp.log.retain	The values are: • true: Job logs are not deleted as long as the application server and JMS are running. • false: Job logs are deleted as soon as they are uploaded to Documentum CM Server. • retainAlways: Job logs are never deleted.
hvp.job.copyOnRepSapAction	One of the functionalities of the Replicate SAP action is to move a document from one folder to another. Instead of moving a document, you can copy the document from one folder to another by setting the value of the *hvp.job.copyOnRepSapAction* parameter in hvp.properties to true. The default value is false.
hvp.job.encryptSapUser	Indicated the complexity of the password stored in the sap_user_config objects. If the value is set to true, the complexity of the password is high and you must reset the existing users' passwords. The default value is false.

Chapter 6

Configuring the View components

This chapter describes how to set the configuration of the View component.

6.1 Configuring the View component

1. Connect to WebAdmin. See "Logging in to WebAdmin through Documentum Administrator" on page 13.
2. Click to expand the **Clients** subnode and select the **View** subnode.

   The **View** screen appears.
3. Right-click **Doclink Viewer Configuration** in the **View** screen and select **Properties**.

   The **View Properties** window appears.

   The field names are described in the following table:

Table 6-1: Parameters

Field name	Description
Available Formats\Best Formats	Defines a list of OpenText Documentum CM content types to be used. The first value defined is the most preferred format, and the last value defined the least preferred format. If this attribute is empty or if the object is not configured, then the View component uses the default content type. For example, if Best Formats is defined as PDF, HTML, the View component first checks for PDF content. If PDF content is not available, View then checks for HTML. If neither content type is available, View uses the default format (for example, WinWord). If this attribute is not configured, then View displays the main document content.

Field name	Description
Filter Formats	Defines the formats to be generated with a Documentum CM Server filter. The formats defined here must be a subset of the formats defined in Best Formats. Any format configured here forces the filter mechanism to be executed to generate the required rendition. Has no effect if not configured. You must have the corresponding filter installed on the server in order to use this feature. For example, you may have a Word-to-HTML filter installed on the Documentum CM Server. The preferred format configured in Best Formats is HTML. When viewing a WinWord document linked with the Best Format, the View component does not find a HTML rendition, and displays the Word document. Because HTML is configured as a filter format, View now launches the filter on the Documentum CM Server and displays the document in HTML.
Standard Attributes\Attributes to Display	The attributes defined here will be used as column header in View's outline view or will be displayed upon request in WebView.
Force Login	If this attribute is selected, the user must enter a password each time a document is launched. This is useful in an environment where several people share the same workstation. This attribute is turned off by default.

4. For each of the formats and attributes that you want to define:

   a. Select the item from the list box.

   b. Click the right arrow to add the rule to the relevant list.

   Note: You can rearrange the order of these items with the up/down arrow or delete them with the delete button.

5. Click **OK** to save the View configuration.

Note: For procedures to install the new client installer (Dmview), see the *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD).*

Appendix A. Troubleshooting

This section provides troubleshooting solutions for some of the known issues.

A.1 Content Services for SAP node not displayed in Documentum Administrator

Documentum Content Services for SAP Solutions node is not displayed in Documentum Administrator.

Suggested Resolution

Verify the following:

- Check if the WebAdmin component has been installed as part of Documentum Administrator deployment.
- Check if the file `DA/custom/app.xml` file has been edited as explained in the *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)*.

A.2 Error on clicking the Content Services for SAP node in Documentum Administrator-WebAdmin

When you click on the Content Services for SAP Solutions node in Documentum Administrator-WebAdmin, the following error message occurs:

```
[DFC_BOF_CANNOT_FIND_OBJECT] Business object com.documentum.ei.core.IDmeiCoreFactory
```

Suggested Resolution

Install Enterprise_Integrations_Core.dar file on the repository. The HVPS.dar should also be installed for Documentum Content Services for SAP Solutions.

A.3 No option to register HVPWorker in WebAdmin

No option to register HVPWorker in WebAdmin. When navigating to the *Content Services for SAP > Auto Manage > HVPS node*, there is no option to register HVP Worker in the Menu.

Suggested Resolution

Install HVPS.dar on the repository.

A.4 Execution of SAP Query errors out

Execution of SAP Query errors out from Documentum Administrator-WebAdmin.

Suggested Resolution

Verify the following:

- The SAP Server and SAP User are configured in Documentum Administrator with the correct SAP Instance number and the client number.
- The SAP Query is executed with the properly configured SAP Server and User.
- Copy the SAP JCo v3.1.x to *da* > *WEB-INF/lib* > directory. The SAP JCo v3.1.x contains sapjco3.jar and two native libraries specific to the Operating System . Make sure you download the Operating System and processor specific SAP JCo v3.1.x. Specify the java.library.path to the JCo library as explained in the Postinstallation configuration section of the *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)*.
- If HVPWorker and Documentum Administrator-WebAdmin are installed on the same application server, then make sure that the SAP JCo v3.1.x libraries are copied to the WEB-INF/lib directories of both Documentum Administrator and HVPWorker and java.library.path for the SAP JCo v3.1.x libraries are specified for both the directories.

A.5 Archivelink repository registration issues

Unable to register repositories related to . The registered repositories do not get displayed.

Suggested Resolution

Verify the following:

- If the repository being accessed is not a GR (Global repository), then make sure the Enterprise_Integrations_Core.dar file is installed on the GR also.
- Install (assap.war) on a supported application server.
- Specify the Global registry details of the repository in the dfc.properties file. Make sure that the Global registry details specified is the same in both the dfc.properties files of Documentum Administrator.

A.6 Documentum Content Services for SAP Solutions Job Execution errors

A.6.1 Jobs upgraded from 5.3 SPx or 6.0 SPx are not getting executed with Documentum Content Services for SAP Solutions 6.5 SPx

Jobs upgraded from 5.3 SPx or 6.0 SPx are not getting executed with Documentum Content Services for SAP Solutions 6.5 SPx. Getting the error on the Documentum Administrator WebAdmin as: the job object indicated the job was in progress, but the job was not actually running. It is likely that the dm_agent_exec utility was stopped while the job was in progress.

Suggested Resolution

The following tasks need to be completed:

- After the upgrade of Documentum Content Services for SAP Solutions, delete the Documentum Administrator - method server cache ($DOCUMENTUM/cache) and restart the repository. Also, restart the method server. Re-execute the job created in 5.3 SPx or 6.0 SPx again.
- If you get the same error on execution of the job, create a new job associating the same agent.
- If upgrading from Documentum Content Services for SAP Solutions 6.0 SPx to 6.5 SPx, then make sure the BOF dependency modules are deleted as explained in the Upgrading Documentum Content Services for SAP Solutions section of the *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)*.

A.6.2 Job do not get executed and no logs are generated

Documentum Content Services for SAP Solutions jobs do not get executed and there are no logs generated; Jobs fail with an Operating System error.

Suggested Resolution

Make sure the Java Method server on the Documentum CM Server is running. The HVP Controller runs in the realm of the Java Method server. Restart the Java method server and re-execute the job

A.6.3 Job Time out exception

Though the HvpWorker is processing the job request, refreshing the job at Documentum Content Services for SAP Solutions WebAdmin shows the job status as *TIMEOUT*, for jobs which are being processed for longer duration.

Suggested Resolution

Try navigating to *Job Management* > *Methods* > *com.documentum.ei.doclink.agentJob.DmeiSapAgent* and increase the timeout values of the method. For example, TimeOut Minimum= 600, TimeOut Default= 5000 and TimeOut Maximum= 10000. Restart the Documentum Content Services for SAP Solutions for changes to take effect.

A.6.4 No HvpWorker available

"No HVP Worker available" in the job trace files.

Suggested Resolution

Ensure that:

- The HvpWorker.war is installed on an application server. The name of the application should not be changed from HvpWorker.
- The dfc.properties of the HvpWorker has the information to connect to the repository. Also, ensure that the SAP JCo v3.1.x libraries are present in the WEB-INF/lib directory of the HvpWorker with the java.library.path specified.
- There is network connectivity between the system on which the Documentum CM Server is running and the HvpWorker system.
- The HvpWorker is registered in WebAdmin.
- The HVP Worker URL is of the format:

```
http://<IP or host>:<port>/HvpWorker/hvpCommand/
```

  Note: The URL is case-sensitive. In 6.5 and 6.5 SP1 Documentum Content Services for SAP Solutions, it is important that the URL ends in a "/". In 6.5 SP2, this is not mandatory.

- The HvpWorker in the Documentum Administrator-WebAdmin is available. The available check box against the HVP Worker should be checked.

  Note: The registered HvpWorker will get rendered unavailable soon after a job is executed, if the previous configurations are not done.

A.6.5 Execution of concurrent jobs – “No HvpWorker available”

“No HVP Worker available” for one of the jobs when multiple jobs are executed at the same time.

The HvpWorker gets rendered unavailable during the time the job is getting executed. The Worker is set back to available soon after the job gets executed. Hence, during the execution of concurrent jobs, the Worker will execute the job that is queued up first. During the time the first job is executed, you would see “No HVP Worker available” for the other jobs. The HvpController that delegates the jobs to the registered HvpWorkers, will poll for the availability of any HvpWorker for 20 times of 5 seconds duration each. If no valid HvpWorkers are made available during such time, the controller comes out with a job failure.

Suggested Resolution

Complete the following:

- Load balance HvpWorkers: Have a setup with multiple HvpWorkers running on different application servers, different systems. The HvpController will have a choice of delegating each of the concurrent jobs to the available HvpWorkers there by removing the dependency on one HvpWorker.
- Reschedule the jobs to run at different intervals.

A.7 SAP JCo errors seen in the HvpWorker logs even though the JCo libraries are present

SAP JCo errors are seen in the HvpWorker log files even though the JCo libraries are present.

Suggested Resolution

Make sure that:

- The SAP JCo version installed is 3.1 and relevant to the Operating system and the processor type used.
- The SAP JCo has a sapjco3.jar and two native libraries. The java.library.path should be specified to the directory where the native libraries are present. The *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)* details on how this could be done for each of the Operating systems.
- If you are running both Documentum Administrator-WebAdmin and HvpWorker on the same application server, then refer to “Execution of Job/ SAP Query having both Documentum Administrator-WebAdmin and HvpWorker installed on the same application server” on page 84.

A.8 Execution of Job/ SAP Query having both Documentum Administrator-WebAdmin and HvpWorker installed on the same application server

Execution of Job or SAP Query having both Documentum Administrator-WebAdmin and HvpWorker installed on the same application server.

Suggested Resolution

Verify the following:

- The SAP JCo v3.1.x library is needed for both execution of a SAP Query from WebAdmin and execution of jobs on the HvpWorker. The SAP JCo v3.1.x libraries should be copied to each of the WEB-INF/lib directories of Documentum Administrator and HvpWorker.
- Specify the java.library.path for both the JCo library directories of Documentum Administrator and HvpWorker. Details on how java.library.path can be specified in the *OpenText Documentum Content Services for SAP Solutions - Installation Guide (EDCCOSAPCS250400-IGD)*.

A.9 Exceptions on HVPWorker console: java.io.FileNotFoundException: hvp.properties

Exceptions on HVPWorker console: java.io.FileNotFoundException: hvp.properties.

Suggested Resolution

This is specific to Documentum Content Services for SAP Solutions 6.5 SP2 and later. Create the "DOCUMENTUM" Operating System environment variable with the path to a directory of choice. Make sure the hvp.properties file packaged with HvpWorker.zip/tar is copied to the `$DOCUMENTUM/share` directory; Edit the hvp.log.base to point to the location where you want the job logs to be generated; Change the hvp.log.level to DEBUG if you need more detailed logs; Restart the application server and the logs should be generated in the location referred to in the hvp.properties file.

Also, on the HvpController system (Documentum Server), copy the hvp.properties file packaged in the `hvps.zip/tar` file to the `$DOCUMENTUM/share` directory. Perform the same set of steps that is done for the HvpWorker. The Controller logs would be generated in the location referred to in the hvp.properties file.

A.10 Controller and Job logs

There is an issue with finding the Controller and Job logs.

Suggested Resolution

You can find the Controller and Job logs at:

- From WebAdmin, right-click on the Documentum Content Services for SAP Solutions job and view trace file. The controller and the agent log files are seen here.
- The detail of the job logs are based on the log level specified in the HvpWorker's hvp.properties file. The default value of the log level in the hvp.properties file is INFO. Setting it to DEBUG will give a detailed job log.
- In Documentum Content Services for SAP Solutions 6.5 and 6.5 SP1, the Job logs are present in the `C:\Documentum\logs` directory on the Documentum CM Server system. The job logs are also present in the path specified in the HvpWorker's hvp.properties file (on the HvpWorker system).
- Starting 6.5 SP2, the hvp.properties file will no longer be packaged in the HvpWorker. The hvp.properties file should be copied to the `$DOCUMENTUM/share` directory. Create a DOCUMENTUM environment variable and copy the hvp.properties file in both the HvpWorker and the Documentum CM Server system (HvpController system). The job logs will be generated in the path specified in the hvp.properties file.

A.11 View trace file of the Documentum Content Services for SAP Solutions Jobs is taking too long to load in the WebAdmin

View trace file of the Documentum Content Services for SAP Solutions Jobs is taking too long to load in the WebAdmin.

Suggested Resolution

The problem is due to many executions of the job, there are many versions of the log files generated which take time to be loaded on Documentum Administrator. A possible resolution could be manually (or have a scheduled job) delete the older logs from the `/System/HVPS/JOBS`.

Note: You might not be able to delete all the job logs because of a pending bug.

A.12 Clean up older job logs

The older logs need to be deleted.

Suggested Resolution

The logs could be manually deleted from the `/System/HVPS/JOBS` or have a scheduled job to delete this.

Note: You might not be able to delete all the job logs because of a pending bug.

A.13 Cannot initialize OpenText Documentum CM DMCL DLL while working with DM view

You must perform the following checks:

- Check if the `java.ini` file points to the correct `jvm.dll` file.
- Check if the CLASSPATH is pointing to the following files:
  - dfc.properties
  - Doclink
  - dctm.jar
  - dfc.jar
- Check if the environment variable PATH is pointing to `Doclink\shared` folder.
- Check if the JAVA_HOME is configured correctly.
- After you perform the previous checks, run the application as an Administrator.