opentext™

OpenText™ Information Archive

SAP and SharePoint Connectors Guide

This guide describes how to install, configure, and use the SAP connector and SharePoint connector for OpenText Information Archive. These connectors are Extract, Transform, Load (ETL) tools for ingesting data. This guide is intended for qualified consultants, system administrators, and knowledge workers who use OpenText Information Archive and its connectors for SAP and SharePoint.

EARCORE250400-AGC-EN-01

OpenText™ Information Archive
SAP and SharePoint Connectors Guide
EARCORE250400-AGC-EN-01
Rev.: 2025-Sept-08

This documentation has been created for OpenText™ Information Archive 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

Preface

i Typographic conventions

Convention	Description
>	Represents a pop-up or pull-down menu.
<Text enclosed within angle brackets>	Represents a variable name for which you must provide a value, or a defined term.
Information in this font	Represents code samples, user input, and computer output.
[] square brackets	Used in method command syntax specifications, square brackets indicate an optional argument.
{} curly brackets	Used in method command syntax specifications, curly brackets indicate an optional argument that can be repeated more than once.

Chapter 1

OpenText Information Archive for SAP connector

The OpenText Information Archive for SAP® connector provides a technology bridge between an SAP (R/3 ERP, or ECC) system and a repository using the SAP HTTP Archivelink® interface.

The OpenText Information Archive for SAP connector enables you to:

- Archive SAP data, reports, and documents into an OpenText Information Archive repository using the SAP ArchiveLink certified interfaces.
- Provide access to data, reports, and documents that are stored in an OpenText Information Archive repository from within the SAP GUI.
- Provide access to documents using the OpenText Information Archive interfaces, which can include the IA Web App and REST services.

1.1 SAP and the OpenText Information Archive for SAP connector terms and definitions

Table 1-1: Terms and definitions

Term	Definition
OpenText Information Archive for SAP Connector	An OpenText Information Archive connector that enables archiving from SAP into the repository and access to the stored information.
HTTP Archiving Services	The server component that uses an HTTP connection to SAP and enables you to access the OpenText Information Archive repository where you can archive reports, store data, and work with incoming and outgoing documents. HTTP Archiving Services is a Java servlet that communicates with SAP ArchiveLink. It gives you the ability to retrieve and view reports and archived documents at any time.
IA Web App	OpenText Information Archive user interface that you can use for searching ingested objects and documents. Also an administrative tool that allows you to: • Create and manage holdings • Manage any repository administration activities
ArchiveLink	A cross-functional interface that is part of the SAP Basis System. ArchiveLink handles storing and retrieving documents, and manages data to and from a repository that is external to SAP.

Term	Definition
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. Graphical menu or screen tool that connects a client to the SAP server.
SAP Document	An electronic transactional record of header data and line items in SAP.

1.2 Supported SAP document formats

OpenText Information Archive for SAP 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)
- All SAP document classes/formats supported OpenText Archive Center. For more information, see *Table 2-2: File types known in libdsh* in section 2.2.6 "Document/component formats (renditions, file types)" in *OpenText Archive Center - Programming Guide for the OpenText Archive Center API (AR240400-PSA)*.

1.3 Installation

General prerequisites

The OpenText Information Archive for SAP connector integrates the OpenText Information Archive platform with the SAP R/3 or ECC system.

Notes

- OpenText recommends that you engage an experienced Professional Services consultant or an OpenText certified partner to perform the installation and configuration.
- OpenText Support cannot assist you with your installation unless your systems were installed by an experienced Professional Services consultant or an OpenText certified partner.

Caution

Before you install the OpenText Information Archive for SAP connector, make certain that OpenText Information Archive is fully installed, running correctly, and that all of its services are accessible. These conditions must be met before you can install the OpenText Information Archive for SAP connector.

For more information about installation and configuration, please refer to the OpenText Information Archive product documentation.

Software prerequisites

The OpenText Information Archive for SAP connector is compatible with specific versions of OpenText Information Archive, and may not be compatible with some third-party products that may or may not be used with a compatible version of OpenText Information Archive.

Support and compatibility

SAP connector supports Java Development Kit (JDK) 17 and Apache Tomcat version 10.1.11.

For more information, see *OpenText Information Archive Release Notes* on support.opentext.com (https://support.opentext.com/).

1.3.1 SAP R/3 or ECC Server (UNIX or Windows)

Manual configuration of SAP ArchiveLink is a prerequisite for the installation of the OpenText Information Archive for SAP connector. For more information, see section 5.1.2 “Connectors” in *OpenText Information Archive - Configuration Guide (EARCORE-CGD)*.

Caution

OpenText Information Archive for SAP connector cannot be responsible for any SAP-related issues encountered during installation. SAP related issues may be caused by empty or wrongly configured tables, required support packages that are missing, hot fixes, or OSS Notes. OpenText requires that you read and apply all ArchiveLink-related OSS Notes and patches prior to the OpenText Information Archive for SAP connector installation. We do not provide support for SAP configuration issues, questions, or issues.

1.3.2 Preparing the application server host

Ensuring sufficient temporary disk space on the host

Application servers vary as to how much temporary disk space they require when an application is installed. Allow at least 512MB of free disk space on the application server host to install the OpenText Information Archive for SAP connector.

In addition to the above requirement:

- On Windows hosts, ensure that the free space is on the drive to which the TEMP environment variable points.
- On UNIX hosts, ensure that the free space is in `$TEMP`.
- On AIX with WebSphere, ensure that the free space is in the temporary directory used by WebSphere, which is `/tmp`.

Ensuring the Java heap size is sufficient

Setting the Java heap size is important to avoid out-of-memory errors. We recommend setting the Java heap size for all application servers to a minimum of 1GB or larger.

1.3.3 Obtaining current software version

To ensure that you install the current version of this product, visit support.opentext.com (https://support.opentext.com) and review the product versions available for your platform.

Windows

For Windows, download `sapconnector-x.x.zip`, which contains the following file and folder:

- `iasap.war`: This file contains the OpenText Information Archive for SAP connector.

1.3.4 Significant system objects

This section contains information about significant system objects that are installed during the OpenText Information Archive for SAP connector war file deployment.

`log4j.properties`

All logging details are available in the `log4j.properties` file, which is located in the following directory:

```
%TOMCAT_INSTALL_ROOT%\webapps\
<INFOARCHIVE_VIRTUAL_DIR>\WEB-INF
```

By default, the log output goes to the `alserver.log` file, which is located in the following directory:

```
%TOMCAT_INSTALL_ROOT%\webapps\
%INFOARCHIVE_ROOT%\WEB-INF/logs
```

You can change the `log4j.properties` file during the servlet's runtime and your updates will be automatically detected.

`al.properties`

The system parameters for the OpenText Information Archive for SAP servlet are defined in the `al.properties` file, which is located in the following directory:

```
%TOMCAT_INSTALL_ROOT%\webapps\<IASAP>\WEB-INF
```

For OpenText Information Archive, configure the following properties. By default, the SAP connector uses OpenText Information Archive as its repository:

```
infoarchive.hostname=<hostname or ip_address>
ia.server.uri=http://<hostname or ip_address>:8765/services
infoarchive.language.code=en_US
ia.applicaton.name=SAPConnector
```

```
ia.server.authentication.gateway=http://<hostname or ip_address>:8080/
ia.server.client_id=infoarchive.sap
ia.server.client_secret=<encrypted clientSecret of infoarchive.sap>
ia.server.authentication.user=<username having access to IAWA>
ia.server.authentication.password=<encrypted login password of above user>

infoarchive.services.user=<Admin user> or <user having access to run IA
webservices>
```

Note: `clientId by name - infoarchive.sap` is dedicated for iasap and configuration details are available in `<IA_ROOT>/config/iawebapp/application-CLIENTS.yml`. Update `refreshTokenValiditySeconds`, as per the requirement.

For more information, see "Setting values in the al.properties file" on page 14. Also refer to Encryption and decryption of Infoarchive connection parameters for more details on how to encrypt `ia.server.client_secret` and `ia.server.authentication.password`.

1.3.5 Deploying the iasap.war file

After you have completed all of the above installation procedures, perform the following steps to prepare the `iasap.war` file for installation:

1. Extract the `sapconnector-x.x.zip` file to a directory location where you want to install the connector. This distribution file contains the `iasap.war` file.
2. Create a temporary directory anywhere on your system. This directory can be used to modify `log4j.properties` file.
3. Copy the `iasap.war` file to the `iasap` directory.
4. Extract the contents of the `jar` file to the `iasap` directory by using the following jar command:

   ```
   jar -xvf iasap.war
   ```

5. Based on the operating system that you are using, edit the `log4j.properties` file so it has the correct path for the log file.
6. Use the `jar -cvf` command to re-jar the `iasap.war` file.
7. Deploy the `iasap.war` file to your application server using the most appropriate application server deployment procedure shown here.

Deploying `iasap.war` Using Apache Tomcat

To deploy the `iasap.war` file using Apache Tomcat:

1. Stop the Apache Tomcat server.
2. Ensure that Apache Tomcat is configured to deploy the `iasap.war` file in unpacked mode. This is Apache Tomcat's default setting.
3. Copy the re-jarred `iasap.war` file to the `<Tomcat_Dir>/webapps` directory.

4. Restart the Apache Tomcat server.

Deploying `iasap.war` on JBoss application server

To deploy the iasap.war on JBoss:

1. Extract the re-jarred `iasap.war` file to a directory called `iasap` with the following command:

```
jar -xvf iasap.war
```

2. Rename the `iasap` directory to `iasap.war`.
3. Copy the `iasap.war` directory to the `<JBOSS_HOME>\server\default\deploy` directory.
4. Restart the JBoss application server.

1.3.6 Setting values in the al.properties file

The `al.properties` file is located in the following directory:

`%TOMCAT_INSTALL_ROOT%\webapps\<IASAP>\WEB-INF`

For more information on working with the `al.properties` file, see al.properties.

To set values in the al.properties file:

1. Stop the Tomcat server.
2. The following parameters must be set when working with OpenText Information Archive 23.2 and later:

- `ia.server.uri=http://<hostname or ip_address>:8765/services"`
- `ia.applicaton.name=SAPConnector`
- `ia.server.authentication.gateway=http://<hostname or ip_address>:8080/`
- `ia.server.client_id=infoarchive.sap`
- `ia.server.client_secret=<encrypted clientSecret of infoarchive.sap>`
- `ia.server.authentication.user=<username having access to IA Web App>`
- `ia.server.authentication.password=<encrypted password of above user>`

Note: SAP connector does encryption and decryption through the IA Decrypt Library. The `<IA_ROOT>/config/sapConnector/al.properties` file contains values encrypted through the IA Decrypt Library. This file is generated during OpenText Information Archive setup.

3. The following parameters must be set in the `al.properties` file to enable decryption of encrypted `ia.server.client_secret` and `ia.server.authentication.password`.

```
passwordEncryption.enabled=true
passwordEncryption.encryptionAlgorithm=
passwordEncryption.encryptionMode=
passwordEncryption.gemaltoClientCertificate=
passwordEncryption.gemaltoPropertiesFile=
passwordEncryption.gemaltoUserName=
passwordEncryption.gemaltoGroup=
passwordEncryption.keyID=
passwordEncryption.keySize=
passwordEncryption.keyStorePath=
passwordEncryption.keyStoreType=
passwordEncryption.paddingScheme=
passwordEncryption.securityProvider=
```

   Values for above keys can be found in the `<IA_ROOT>/config/sapConnector/al.properties` file. Refer to Encryption and decryption of Infoarchive connection parameters for more information.

   Note: `passwordEncryption.enabled=false` would accept plain values for `clientSecret` and `password` with no encryption. It is recommended, however, to use encrypted values for `ia.server.client_secret` and `ia.server.authentication.password` in a production environment.

4. Clear the cache and restart the Tomcat server.

   Note: Updating the `archiving.enableIngestionCache` parameter to true in the `al.properties` file can help improve the performance of update type requests received from SAP. By default, this parameter is set to true.

Encrypting and decrypting of OpenText Information Archive connection parameters

Some of the sensitive information, such as `clientSecret` and `password`, need to be encrypted before adding it to the `al.properties` file.

`ia.server.client_secret` and `ia.server.authentication.password` can be encrypted in following ways:

- Password encryption can be enabled as a part of an OpenText Information Archive installation done through the install script. Post installation encrypted values and other encryption related parameters can be found in the `<IA_ROOT>/config/sapConnector/al.properties` file.
- Password encryption can be done manually using password encryption utility available in OpenText Information Archive. Encrypted values and other parameters can be added to the `al.properties` file accordingly. For more information, see section 7.2.2 “Password encryption utility” in *OpenText Information Archive - Encryption Guide (EARCORE-AGE)*.

Editing and viewing the al.properties file using JConsole

This section describes the procedures that are used for viewing and editing the `al.properties` file in the JConsole editor without the need to restart the application server.

Configuring Apache Tomcat

To enable the JMX agent for Apache Tomcat for a Windows environment, add the following line in the `<TOMCAT_HOME_DIR>\bin\catalina.bat` file:

```
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=9999
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

To enable the JMX agent for Apache Tomcat for a Solaris environment, add the following line in the `<TOMCAT_HOME_DIR>\bin\catalina.sh` file:

```
set JAVA_OPTS=$JAVA_OPTS  -Dcom.sun.management.jmxremote.port=9999
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
```

Editing `al.properties` using JConsole

To edit the al.properties file using JConsole:

1. Open a command prompt at:

   ```
   <JAVA_HOME>/bin
   ```

2. Type `JConsole` to display the JConsole editor.
3. In the remote tab, enter `<Host Name>` and the JMX port number.

   Use `9999` for the JMX port number.
4. Click **Connect**.
5. In the `com.documentum.ei.al.jmx.mbeans` section, navigate to the **Mbeans** tab and click **AlProperties**.
6. Edit the desired properties and press **ENTER**.

1.3.7 Configuring the temporary store

The `al.properties` file is located in `%TOMCAT_INSTALL_ROOT%\webapps\<IASAP>\WEB-INF`. You must configure the `al.properties` file so that it points to a valid directory with complete access rights. The OpenText Information Archive for SAP connector writes data to a file in this temporary store before reading or writing to an HTTP Request.

For example, customize the relevant section of the `al.properties` file as follows:

```
archiving.tempDir=C:\\archivelink
```

Note: When you are running the OpenText Information Archive for SAP connector on UNIX, you must customize the file differently. In UNIX, you could customize the relevant section as follows:

```
archiving.tempDir=/usr/temp/AL_SERVER_DATA
```

1.4 Testing the installation

Use the information in this section to test the OpenText Information Archive for SAP connector installation.

1.4.1 Connecting to OpenText Information Archive IA Web App

To connect to the IA Web App:

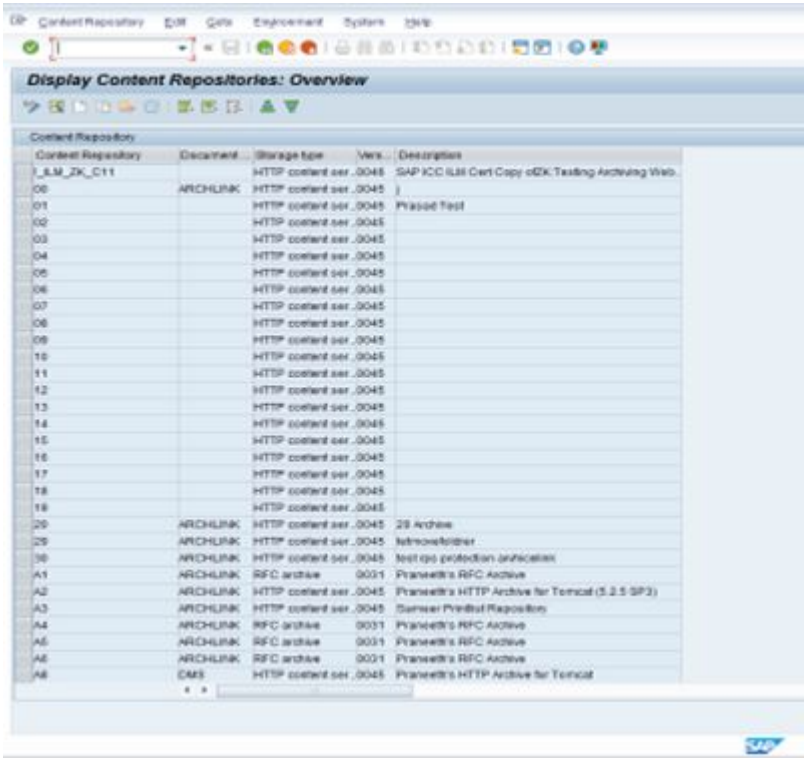
1. Open a web browser on your client machine.
2. Connect to the following URL, where `host` is the location of your IA Web App installation, and `portnumber` is the port number provided during application server installation:

   `http://<hostname or ip_address>:8080/`
3. Type your login name and password on the IA Web App login screen.
4. Click **Login**.
5. After you have successfully logged in, open the SAPConnector application and verify the holding installations.

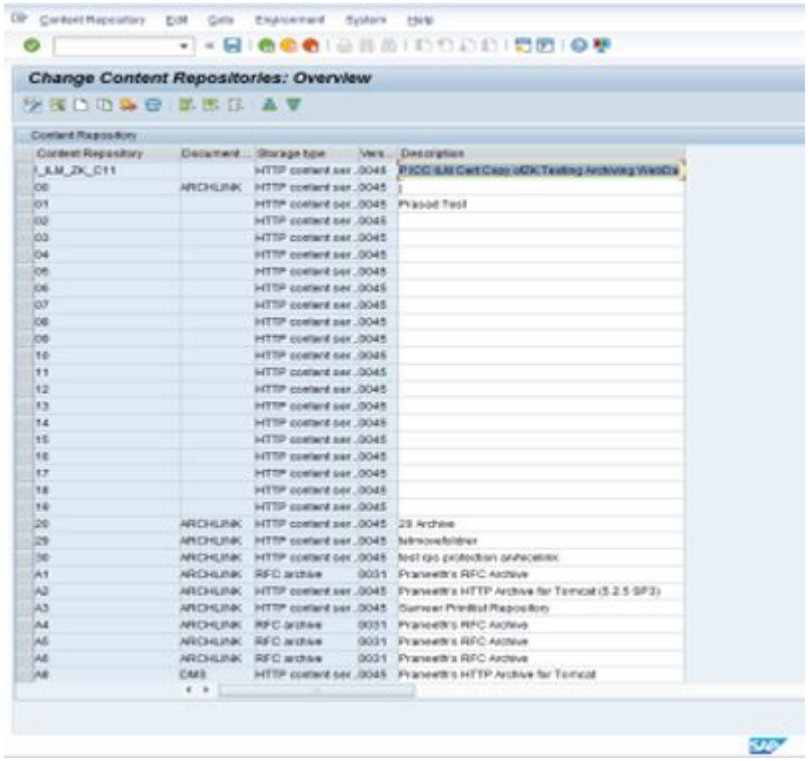
1.4.2 oac0 – Defining a logical ArchiveID in SAP

To define a new ArchiveID in SAP:

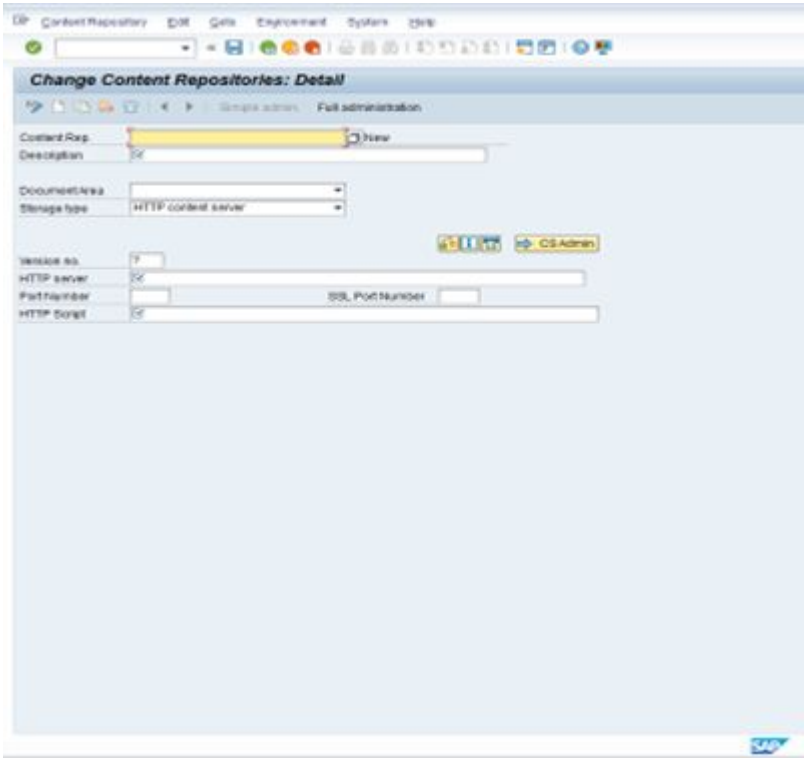
1. Open the SAP GUI.
2. In the archive transaction code field, enter `oac0` and click on the check mark icon to run it.



3. In the **Display Content Repositories: Overview** screen, click the **Display/ Change** icon to set the screen to detail view.

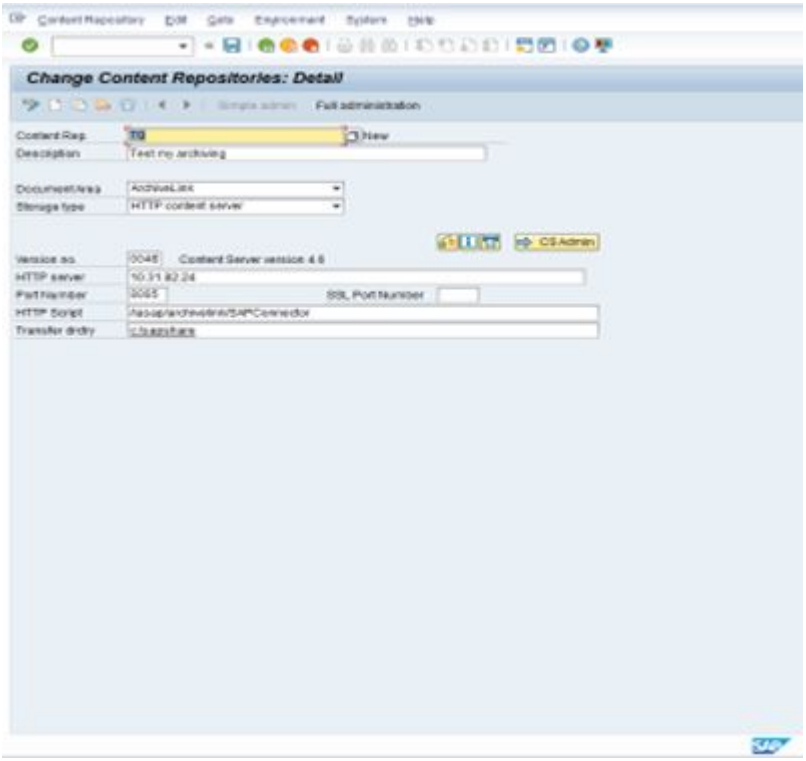


4. In the **Display Content Repositories: Detail** screen, click the **Display/Change** icon to enable the **Create new** icon.



5. Optional You can use an existing ArchiveID and the **Copy as** icon (next to the **Create new** icon) to create a working copy of the existing ArchiveID that you can edit and use as your new ArchiveID.

6. Enter all of the values for your new ArchiveID. You must enter values for all of the fields on this screen, because they are all required.

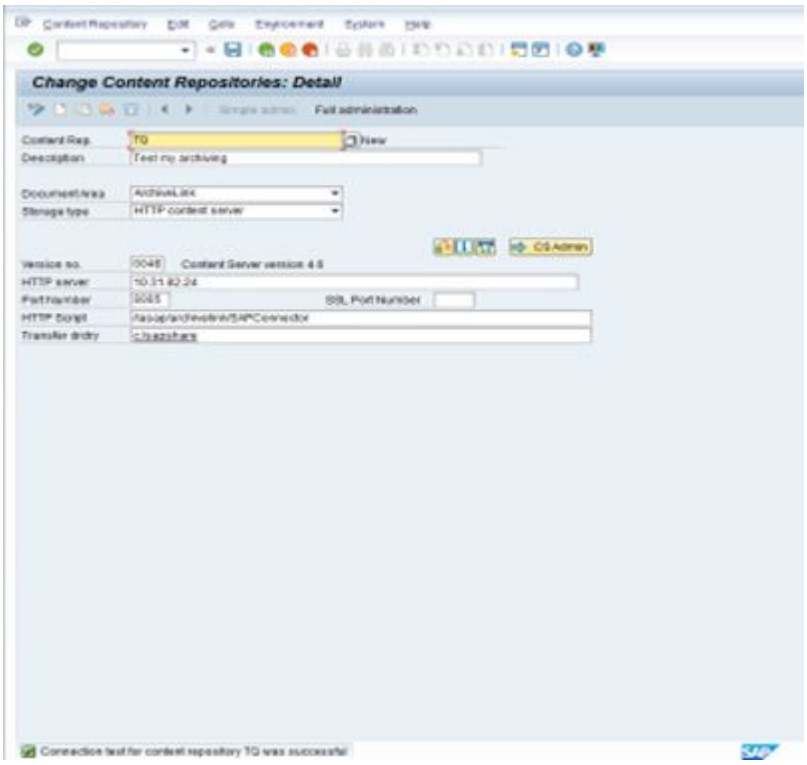


- **Content Rep.**: Enter a name for your new ArchiveID.
- **Description**: Enter a description for your ArchiveID.
- **Document Area**:
  - *When you are archiving files*: Use `ArchiveLink` as the value for the **Document Area** field.
  - *When you are archiving data*: You can leave this field empty or use `DataArchiving` as the value for the **Document Area** field.
- **Storage type**: Select `HTTP content server` as the storage type.
- **Version no.**: Enter the SAP ArchiveLink version (or SAP Content Server version) that is supported. Versions 4.5, 4.6, and 4. 7 are all supported.
- **HTTP server**: Enter the IP address of your system.
- **Port Number**: Enter the port number used by the application server that is hosting the OpenText Information Archive for SAP connector.
- **HTTP Script**: Enter the path used to invoke the OpenText Information Archive for SAP connector.

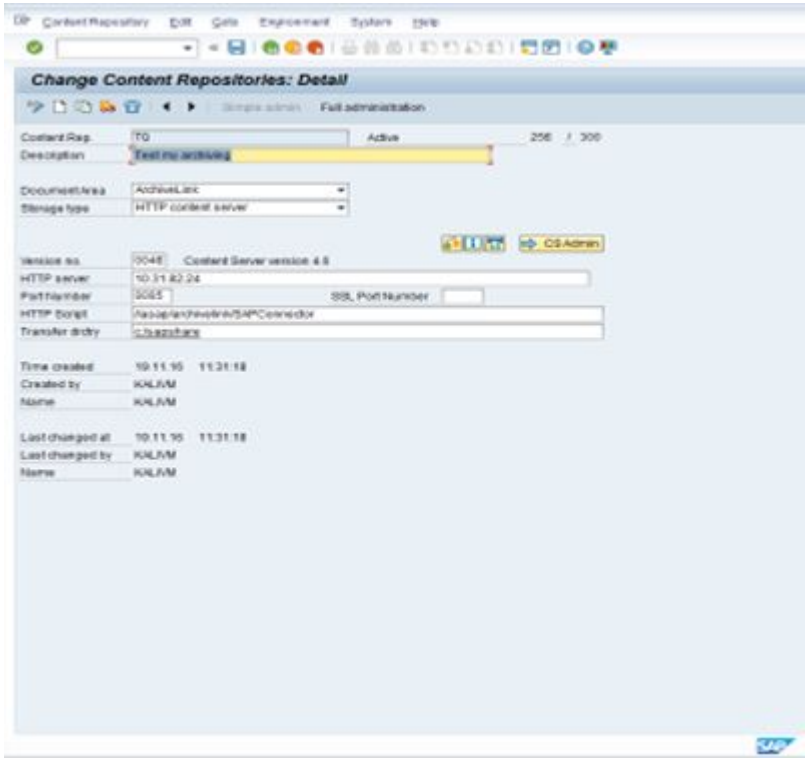
```
<IASAP>/archivelink/<SAPConnector_holding_name>
```

- **Transfer drctry**: Enter the path to the directory where SAP can write the Print List.

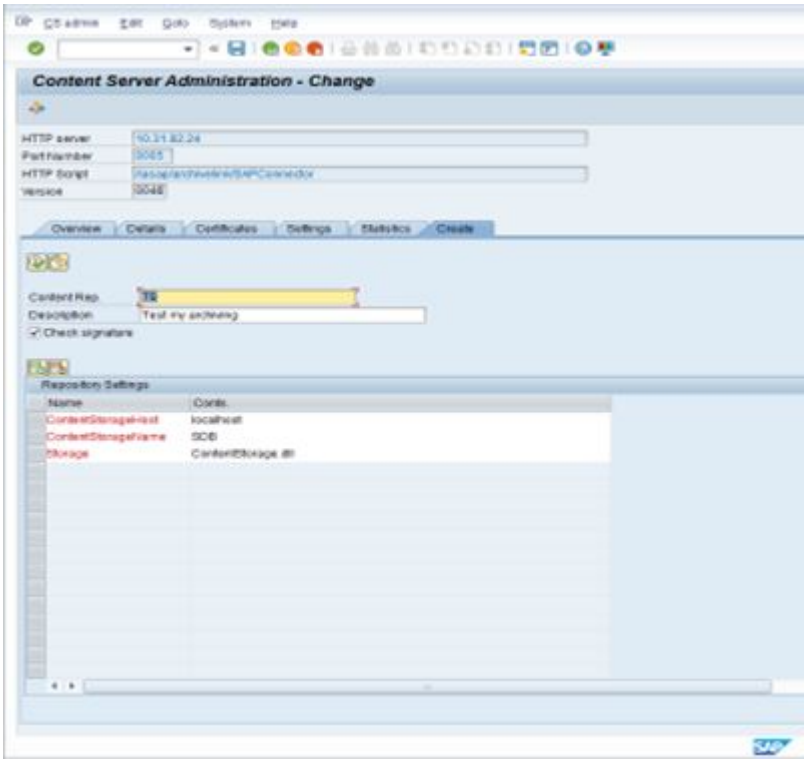
7. Click the **Test connection** icon to verify your connection to your content repository. The test result is shown at the bottom of the screen.



8. Click the **CS Admin** button to go to the **Content Server Administration - Change** screen.

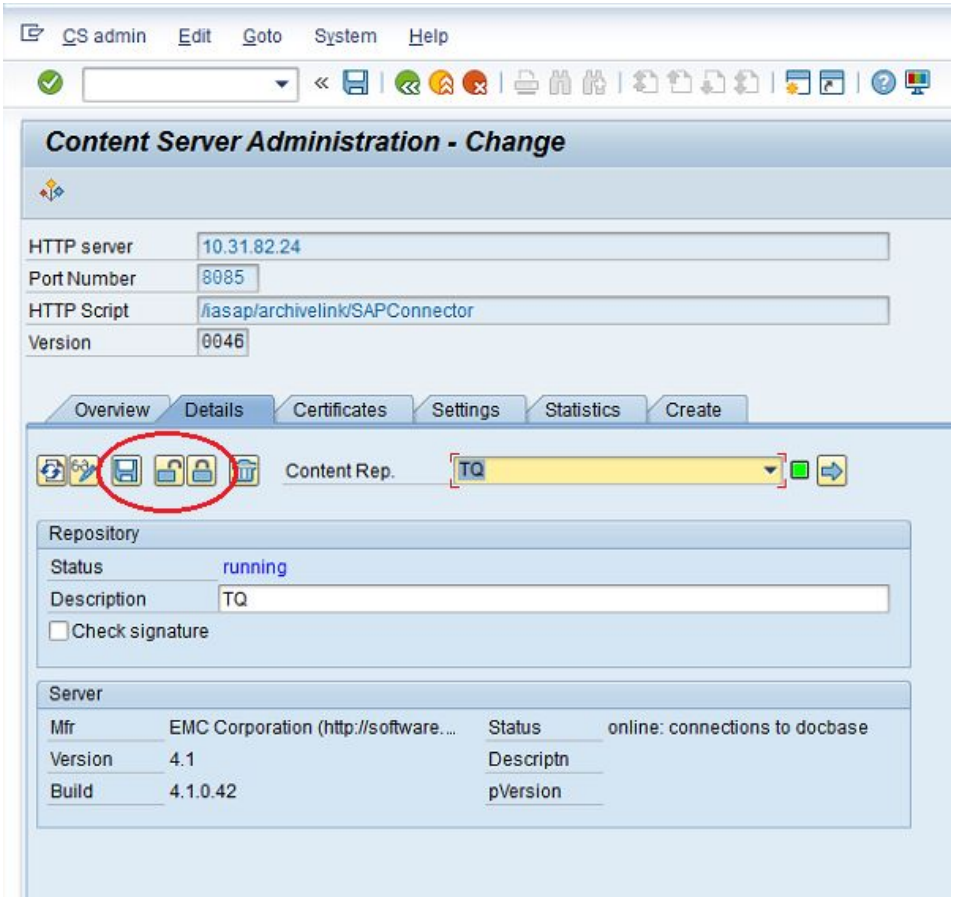


9. In the **Content Server Administration - Change** screen, click the **Create** tab, then click the **Create Repository** icon.



10. Click the **Unlock** icon before making any changes. The **Unlock** icon changes the repository status from **defined** to **running**, which allows you make and save your changes.

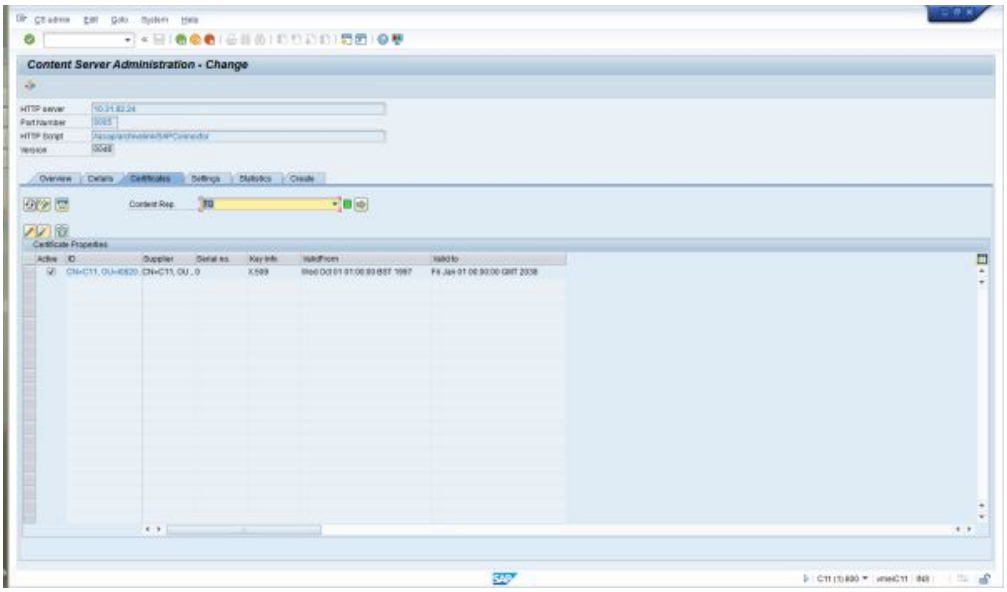
11. Your new ArchiveID has been created and is running in the repository. Click the **Save** icon.



12. You must send the certificate to OpenText Information Archive before you can use your new ArchiveID. The certificate allows the system to identify the SAP instance that is sending it data. Once the export of the certificate is successful, you can start using your new ArchiveID to archive documents that are sent from SAP.

    Click the **Certificates** tab to activate it and click the **Send Certificate** icon to send the certificate to OpenText Information Archive.

Now you can view your new ArchiveID details in the **Certificate Properties** table.



1.4.3 Creating a document type in SAP

To test the archival and retrieval cycle, a Print List can be configured by following these instructions:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   ```
   oac3
   ```

   The **Display View "Links for Content Repositories": Overview** screen appears.
3. Select **Table View > Display > Change**.
4. Select any item with the ObjectType **SOOD** from the listbox.
5. Click the **Copy As** icon and run.
6. Define a new document type using the `oac2` transaction code.

   For more information, see oac2 - Defining Document types.
7. In the **Storage system** field, enter the name of the new ArchiveID that you created earlier in "oac0 – Defining a logical ArchiveID in SAP" on page 17, which is where the documents are to be archived.
8. Click the **Save** icon to save your changes.

1.4.4 Archiving a PrintList

To archive a PrintList:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   ```
   f.21
   ```

3. Click the **Execute** icon.
4. In the **List of Customer Open Items** screen, select **List > Print**.

   The **Print Screen List** appears.
5. Click **Properties** to display the **Spool Request Attributes** screen.
6. In the **Parameters name** column, click **Output Options**.
7. From the **Storage Mode** list box, select **Archive only**.

   Note: We recommend that you select **Show Selected Print Parameters on Initial Screen** whenever you edit a parameter in this screen. This ensures that the customized settings are available in the initial **Print Screen List** screen.
8. Set the **Object Type** to **SOOD**.
9. In the **Document Type** field, enter the document type you created earlier in "Creating a document type in SAP" on page 49, and run it.
10. Enter additional information in the **Information** field.

    The information field can contain any information that you want to enter there.
11. Optional You can enter a description of the Print List in the **Text** field.
12. In the **Parameter name** column, select **General attributes > Time of printing**.
13. From the **Time of print** list box, select **Print out immediately**.
14. Click **Continue** in the **Spool Request Attributes** dialog box.
15. Click **Continue** in the **Print Screen List** dialog box.
16. To view the status of the document in the storage queue, run the following transaction in the transaction code field:

    ```
    oam1
    ```

    The **ArchiveLink Monitor** screen appears.

    The Print List is now queued, waiting for the scheduler to pick it up and transfer it to the archive.

    For more information, see oaat - Scheduling jobs.

17. To accelerate the transfer of the Print List from the queue to the archive in the repository, click **Storage Queue**.

    The **Queue: Content server (CARA)** screen appears. You should now see an entry for the newly archived Print List in this screen.

18. To view the archiving parameter details, double-click the Print List entry.

    The **Archiving request** window opens.

    Verify that the details are correct.

19. To archive the Print List in an archive located in a repository, click **Execute**.

    You should see a confirmation message that the queue has been processed.

    If no errors occur, you can continue testing by displaying the archived Print List.

1.4.5 Viewing a Print List

To view the Print List, do the following:

1. Open the SAP GUI.
2. Run the following code in the transaction code field: `oadd`.
3. In the **ArchiveLink: Hit List for Stored Print Lists** screen for a particular item, select **Print List > Display From Content Server**.

   SAP runs the **GET** command to retrieve the document from OpenText Information Archive. The retrieved document has the ArchiveID of the Print List in OpenText Information Archive.

1.5 Troubleshooting installation issues

This section discusses how to resolve some commonly encountered issues in the OpenText Information Archive for SAP connector.

1.5.1 High levels of logging

You may encounter high levels of logging if the logging parameter for the OpenText Information Archive for SAP connector has been set as DEBUG or WARN. By default, this parameter is set as WARN.

When the logging parameter is set to DEBUG, the OpenText Information Archive for SAP connector logs each and every request. In a production environment, this setting may cause an extremely high level of logging to occur, which may cause the following issues:

- Since the OpenText Information Archive for SAP connector writes each and every Request to a log, this level of logging may use a large amount of storage space.
- High levels of logging degrade the performance of the host.

When the logging level is set to DEBUG, the OpenText Information Archive for SAP connector writes certain parameters of each Request to memory. With each Request, the amount of memory used by the OpenText Information Archive for SAP connector increases incrementally. This leads to a performance penalty.

To change the logging level:

1. Open the `log4j.properties` file.

   In Windows, this file is usually located in the following location:

   ```
   %TOMCAT_INSTALL_ROOT%\webapps\<IASAP>\WEB-INF
   ```

2. Locate the following parameters:

   ```
   logger.eilog.name=com.documentum.ei
   logger.eilog.level=WARN
   ```

   Note: Depending on the setting of the logging levels in the environment, change the `logger.eilog.level` to WARN, INFO, or DEBUG. By default, the logging level is set to WARN.

3. We recommend that you update `logger.eilog.level` appropriately based on the desired logging level.

   Example 1-1: Setting the logging level as INFO

   When you want to set the logging level to INFO, edit the logging parameters as follows:

   ```
   logger.eilog.name=com.documentum.ei
   logger.eilog.level=INFO
   ```

   With this setting, the logging level is changed to `INFO`.

4. Save the file.

Note: Setting the logging level to `DEBUG` is necessary only when you need to troubleshoot issues with the OpenText Information Archive for SAP connector. In a production environment, the `DEBUG` logging parameter for the OpenText Information Archive for SAP connector must be commented out, which turns it off.

1.5.2 Incorrect Tomcat installation

After installing the OpenText Information Archive for SAP connector, when you connect to the IA Web App for the first time, you may see this JSP error:

```
Unable to find a javac compiler; com.sun.tools.javac.Main is not on the classpath.
```

This error message is displayed because the JRE option was selected, instead of the J2SE option, in the Tomcat installation wizard.

Follow these steps to resolve this issue:

1. Stop the Tomcat server.
2. Download and install a full J2SE/JDK on a local machine.
3. Copy the `tools.jar` file from the `lib` directory, which is located in the following directory:
   ```
   <J2SE-Installation-Directory>/lib
   ```
4. Paste the `tools.jar` file to the following directory:
   ```
   %TOMCAT_INSTALL_ROOT%/common/lib
   ```
5. Restart the Tomcat server.

1.5.3 To verify whether a document has been archived correctly

Sign in to the IA Web App and perform a general search for all documents. Documents that have been archived correctly will be present in the search results.

1.6 Uninstalling the OpenText Information Archive for SAP connector

Uninstalling the OpenText Information Archive for SAP connector consists of the high-level tasks and procedures listed in this section.

1.6.1 Preparing application servers for IA SAP uninstallation

To prepare application servers for the OpenText Information Archive for SAP connector uninstallation:

- Close all browser sessions that are running WebAdmin.
- Ensure that the application server is in the correct state:
  - Apache Tomcat must be stopped.
  - Oracle Application Server must be running.
  - Sun Java System Application Server must be running.

1.6.2 Uninstalling IA SAP using Apache Tomcat

To uninstall using Apache Tomcat:

1. Stop the Apache Tomcat server.
2. Delete the `iasap.war` file from the `<Tomcat>/webapps` directory.
3. Restart the Apache Tomcat server.

1.7 Configuration and administration

1.7.1 ArchiveLink interface for OpenText Information Archive for SAP

You can store and retrieve documents, reports, and data through SAP ArchiveLink. Prior to running the OpenText Information Archive for SAP connector, the SAP R/3 or ERP systems must be configured. This chapter describes how to use the SAP GUI to configure SAP R/3 ArchiveLink, and how to use other related transactions.

Note: The configuration steps often reference transaction codes. These codes allow you to navigate directly to the correct configuration screen. Most of the system configuration is performed in the SAP Implementation Guide for R/3 customizing (IMG). You can navigate to this screen using the following transaction code:

```
spro
```

When configuring the OpenText Information Archive for SAP connector, you configure an HTTP-based Archive Server.

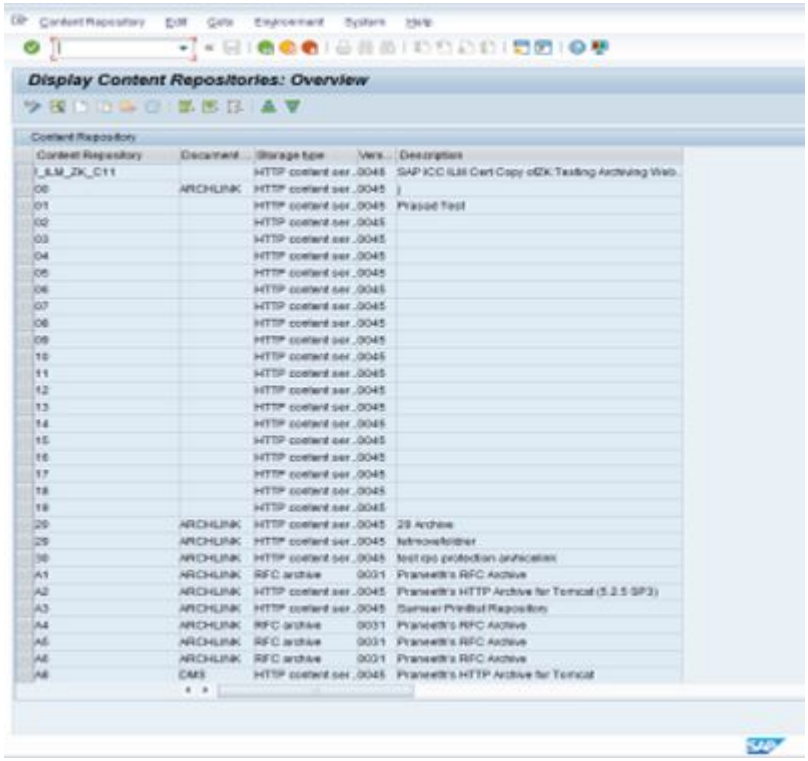
Note: OpenText Information Archive recommends that all new installations use the HTTP-based Archive Server. The primary reason is that SAP is meant to be used with the HTTP-based archive protocol.

1.7.1.1 oac0 – Defining a logical ArchiveID in SAP

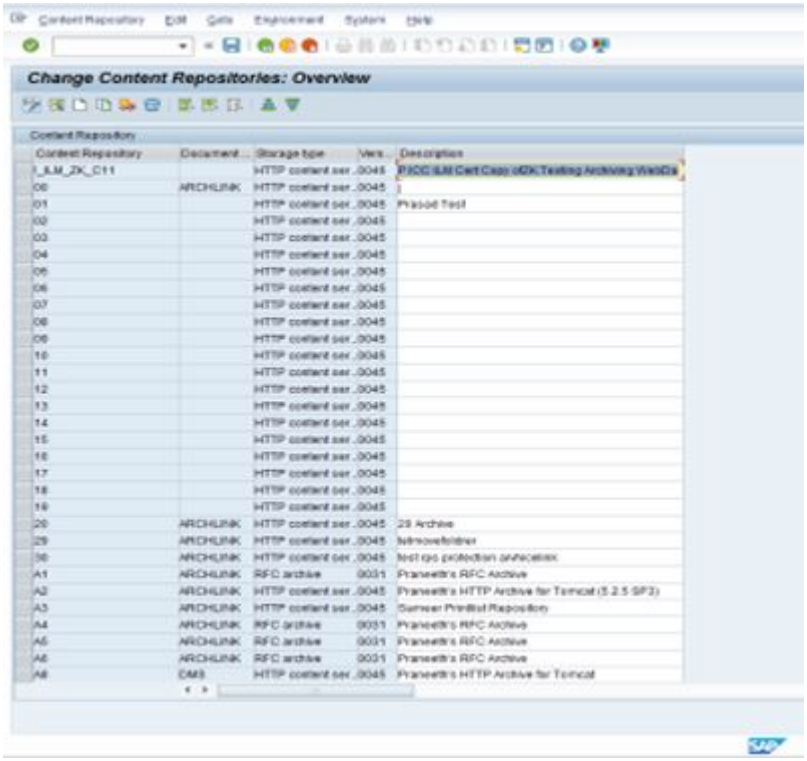
The name of the new ArchiveId and the ArchiveId created using the IA Web App must be the same.

To define a new ArchiveId in SAP:

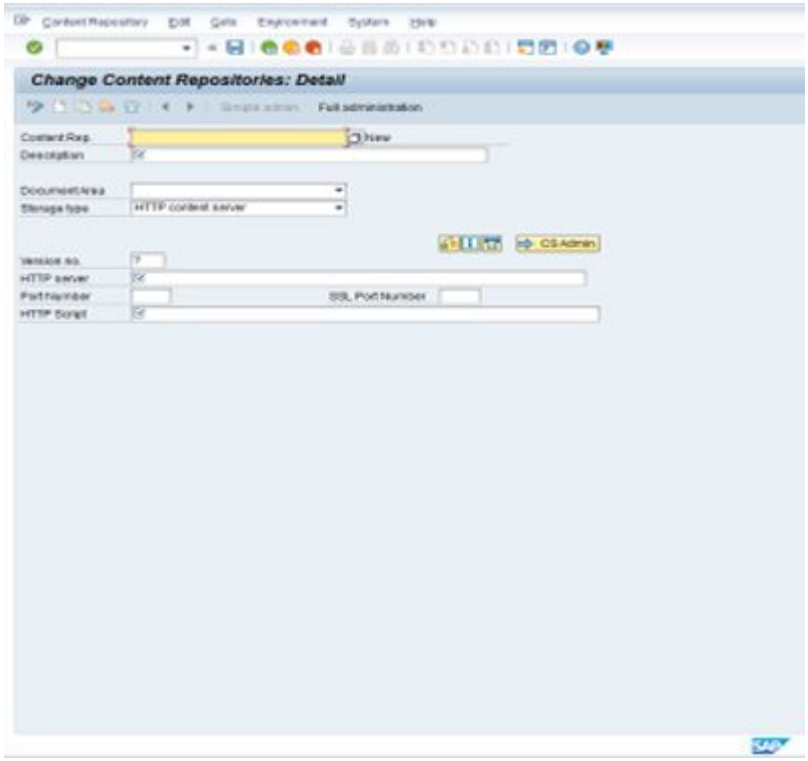
1. Open the SAP GUI.
2. In the archive transaction code field, enter `oac0` and click on the check mark icon to run it.



3. In the **Display Content Repositories: Overview** screen, click on the **Display/ Change** icon to set the screen to **Detail** view.

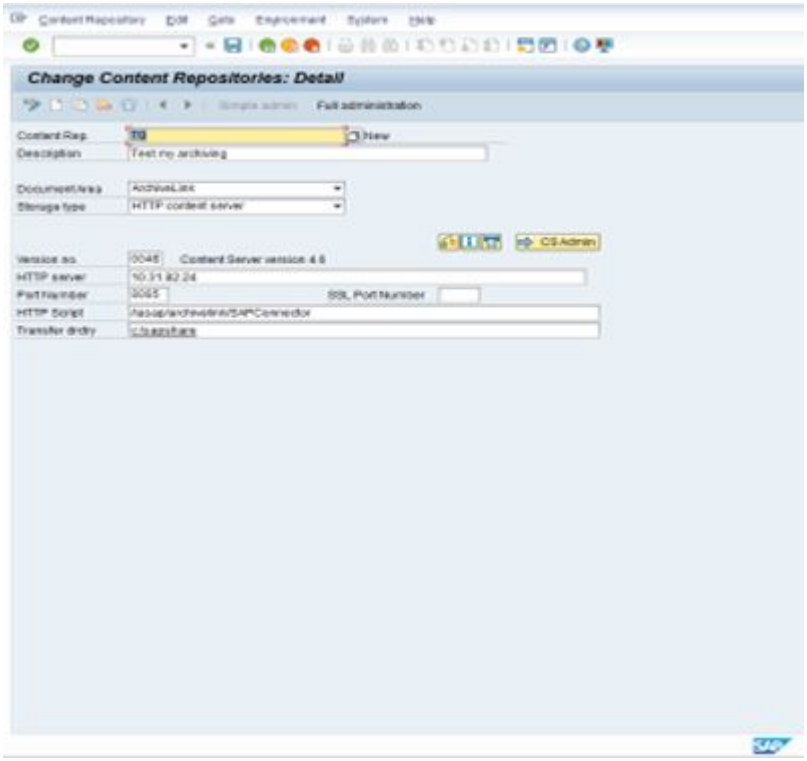


4. In the **Display Content Repositories: Detail** screen, click **Display/Change** icon to enable the **Create new** icon.



5. Optional Use an existing ArchiveId and the **Copy as** icon (which is next to the **Create new** icon) to create a working copy of the existing ArchiveId that you can edit and use as your new ArchiveId.

6. Enter all of the values for your new ArchiveId. You must enter values for all of the fields on this screen, because they are all required.



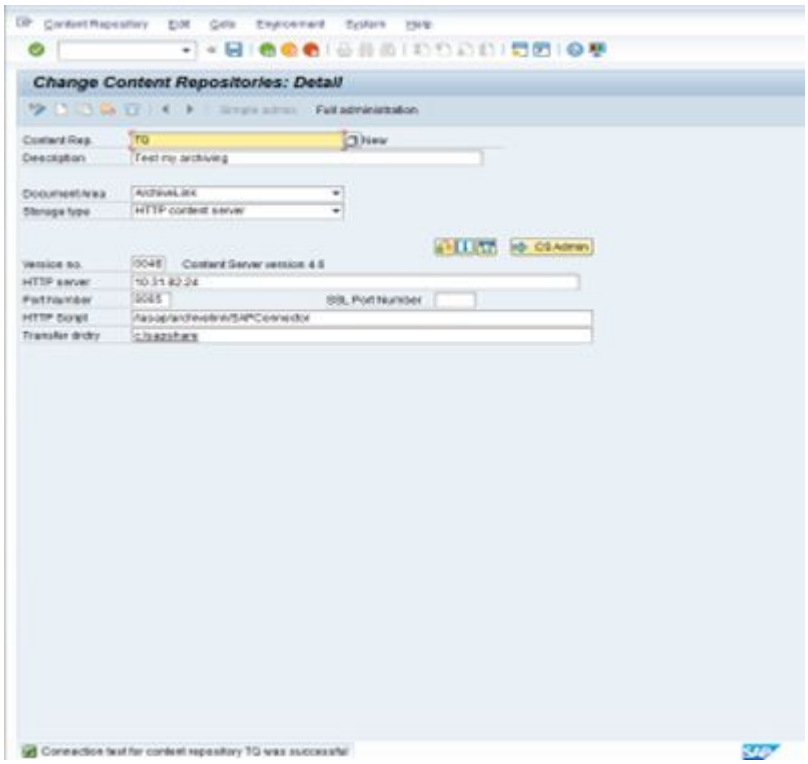
- **Content Rep.**: Enter a name for your new ArchiveId.
- **Description**: Enter a description for your ArchiveId.
- **Document Area**:
  - *When you are archiving files*: Use `ArchiveLink` as the value for the **Document Area** field.
  - *When you are archiving data*: You can leave this field empty or use `DataArchiving` as the value for the **Document Area** field.
- **Storage type**: Select `HTTP content server` as the storage type.
- **Version no.**: Enter a supported SAP ArchiveLink or SAP Content Server version.
- **HTTP server**: Enter the IP address of the app server where the connector is installed.
- **Port Number**: Enter the port number used by the application server that is hosting the OpenText Information Archive for SAP connector.
- **HTTP Script**: Enter the path used to invoke the OpenText Information Archive for SAP connector.
  - `<IASAP>/archivelink/<SAPConnector_holding_name>` to apply the default retention policy.

  - `<IASAP>/archivelink/<SAPConnector_holding_name>/rt_<retention_class_name>` to apply the specific retention class that is defined during the creation of the holding in OpenText Information Archive.

    Note: The retention class must start with `rt_`. IASAP recognizes `rt_*` as a retention class.

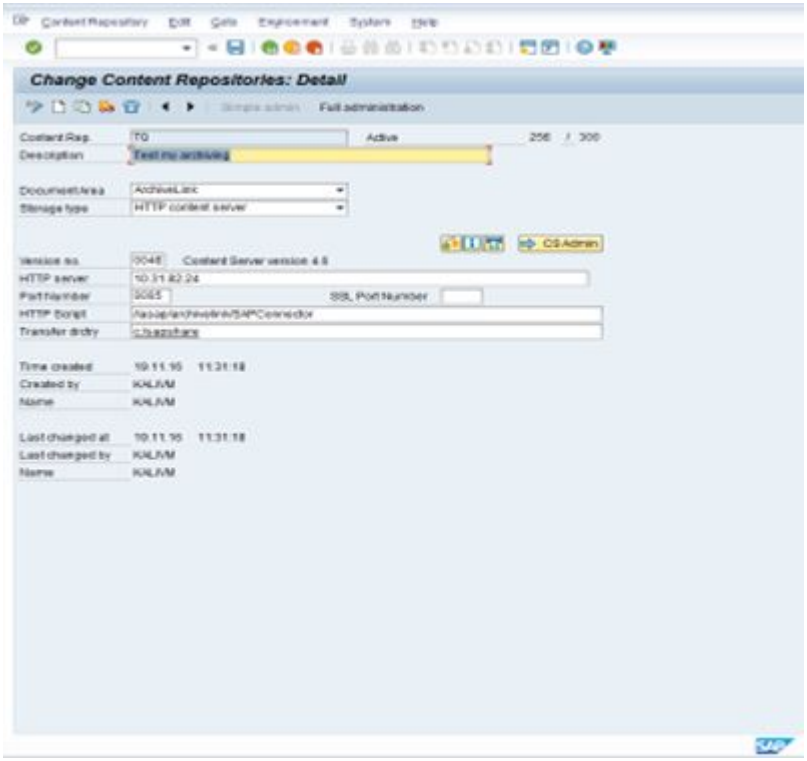
- **Transfer drctry**: Enter the path to the directory where SAP can write the Print List.

7. Click the **Test connection** icon to verify your connection to your content repository. The test result is shown at the bottom of the screen.

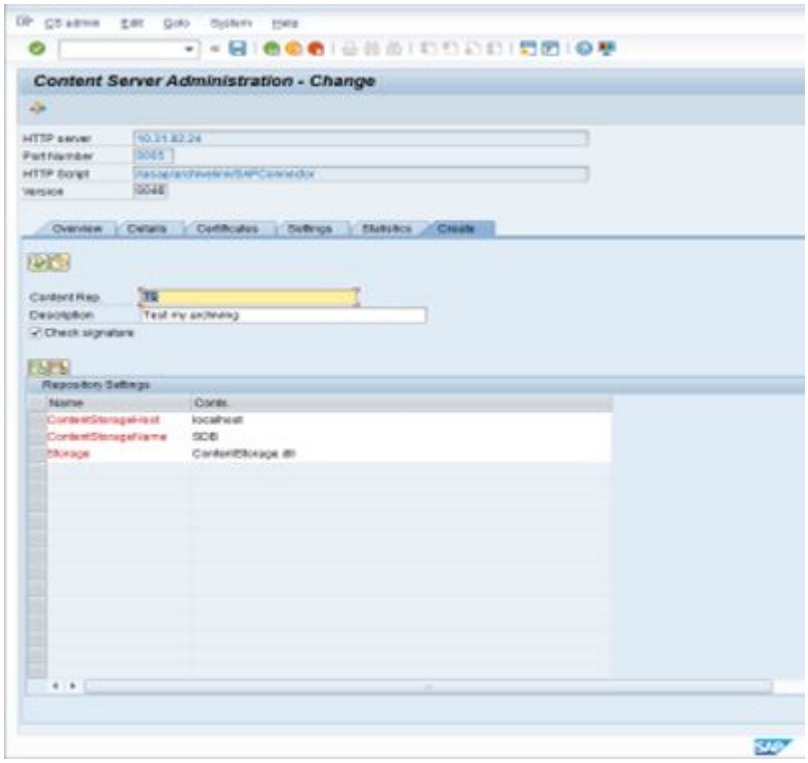


8. Click the **Save** icon on the toolbar to save the ArchiveId details.

9. Click the **CS Admin** button to go to the **Content Server Administration - Change** screen.



10. In the **Content Server Administration - Change** screen, select the **Create** tab, then click the **Create Repository** icon.

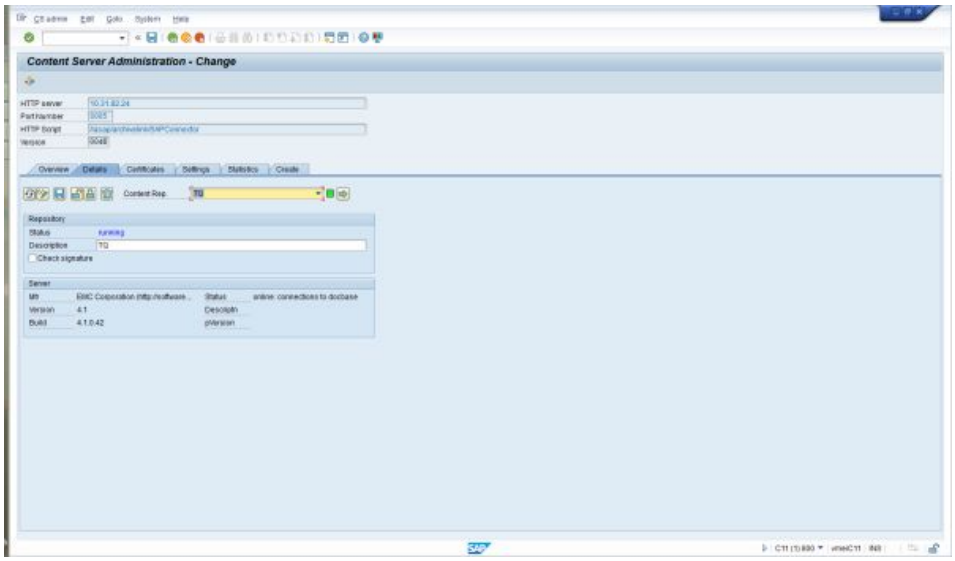


11. Click the **Unlock** icon before making any changes. The **Unlock** icon changes the repository status from **defined** to **running**, which allows you make and save changes.

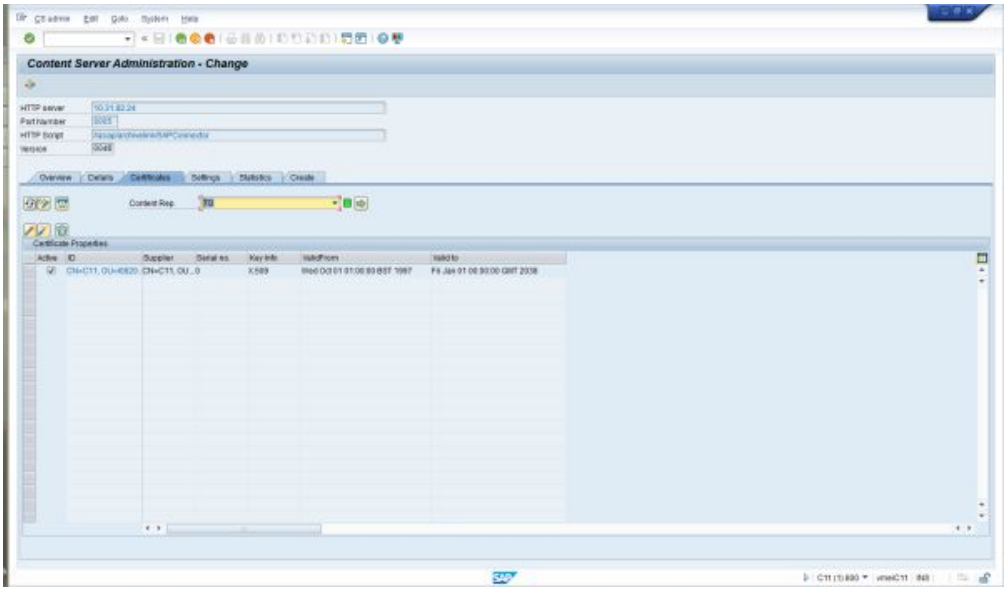
12. Your new ArchiveId has been created and is running in the repository. Click the **Save** icon.

13. You must send the certificate to OpenText Information Archive before you can use your new ArchiveId. The certificate allows OpenText Information Archive to identify the SAP instance that is sending it data. Once the export of the certificate is successful, you can start using your new ArchiveId to archive documents that are sent from SAP.

    Click on the **Certificates** tab to activate it and click the **Send Certificate** icon to send the certificate to OpenText Information Archive.



Now you can view your new ArchiveId details in the **Certificate Properties** table.



1.7.1.2 oag1 – Configuring basic settings

You must define the basic settings for ArchiveLink.

To configure the basic settings:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:
   `oag1`
3. Ensure that **Deactivate Print List Management in DMS** is selected.
4. Save your changes.

1.7.1.3 oanr – Configuring number ranges

You must configure the number ranges for ArchiveLink.

To configure the number ranges:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:
   `\oanr`.
3. To edit the configuration, click **Intervals** (marked with a pencil).
   The **Display Number Range Intervals** window opens.
4. Configure **Number Range 01** with default values.
5. Save your changes.

1.7.1.4 oaqi – Creating queues

When Print Lists are archived, the SAP print spooler puts the document into the asynchronous queue. The scheduler then picks up the document and sends it to the archive. The queue serves as a buffer for everything sent to an SAP archive. Other queues are used for outgoing documents and barcodes. These queues can be seen on the ArchiveLink Monitor screen (transaction code: `oam1`). If this screen shows the word MISSING instead of zeros, you must create queues and specify an administrator. Define an SAP user with the proper ArchiveLink administrator profile as the administrator for these queues. Defining an administrator will automatically create the queues. Check that the queues have been created by reviewing transaction Amy.

To create queues:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:
   `\oaqi`

The **SAP ArchiveLink: Create All Queues (CFBC, CARA, CGDA)** screen appears.

3. Fill in all the options with `x`.
4. Type a valid SAP login name in the **Queue Administrator** field.
5. Select **Program > Execute**.
6. Click **Cancel** to close this window.

1.7.1.5 oaat – Scheduling jobs

The SAP schedule job periodically checks the output queue and sends all the items in the queues to the archive. You must configure the schedule job to successfully archive documents to an SAP archive.

To schedule a job:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field: `oaat`
3. Create a new job.
4. Type `ILQBATCH` in the **ABAP Program** field.
5. To verify that there are no errors, click **Check**.
6. Save your changes.
7. Click **Back** to return to the **Define Background Job** window.
8. Click **Start Date**.
9. Set **Start Time** to **Immediate**.
10. Select **Periodic Job**.
11. Click **Period Values** and specify an interval such as 20 minutes.

    This value should be determined by the following factors:

    - Frequency of archival
    - Time taken to archive your largest document; for example, your largest Print List
12. Save your changes
13. Click **Exit**.

1.7.1.6 spad – Configuring optical archives as output devices

An output device is the driver software for the logical output device which archives documents using the ArchiveLink interface. To correctly archive a document to SAP, you must define the output device as an optical archive. Configuring an output device includes defining the type, the device, the spool server, and so on. There should be only ONE ARCH device; therefore, prior to defining a device with the ARCH short name, you should delete any existing ARCH devices such as ARCHIXOS.

To configure an optical archive as the output device:

1. Open the SAP GUI and connect to your R/3 server.
2. Run the following transaction in the transaction code field: `spad`.
3. Click the **Output Devices** button.
4. Click the pencil icon to edit settings.

   Note: You can only have one printer and that printer must be named ARCH. If you are already using a printer named ARCH you can either delete it or use the existing ARCH device.

5. In case ARCH already exists, select device ARCH and click **Delete**.
6. Click the **New Entries** button to create a new output device.

   The **Change Output Device** screen is displayed.

7. Fill in the following information:

   Table 1-2: Field Descriptions of Change Output Device Screen

Field name	Value
Output Device	`ARCHIVE`
Short Name	`ARCH`
Device Type	ARCHLINK
Spool Server	Select spool server from drop list. Usually there is only one spool server listed
Device Class Type	`A`, for archive program
Access Method Fill	`I`, for archive service
Location Fill	`Archive`
Message Fill	`ArchiveLink Device`

8. Click **Enter** and the dialog box changes.
9. Save your changes.

1.7.1.7 sm50 – Verifying spool processes

OpenText Information Archive uses the standard SAP print spooler to send reports to the ArchiveLink queue. To successfully archive a document, the print spooler must be running.

To verify if a spool process is running:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field: `sm50`.
3. Verify that a process named SPO is waiting or running.

   Contact your SAP system administrator if you do not have a spool process running. Starting a spool process usually involves restarting the SAP server.

1.7.1.8 oaa3 – Configuring the SAP inline Print List viewer protocol

SAP R/3 includes a Print List viewing feature that allows you to view saved Print Lists, in SAP GUI, in their native ALF format. To view saved Print Lists in SAP GUI, you must configure the protocol to support the viewer.

To configure the Print List viewer to display the Print Lists in the SAP SAP:

1. Open the SAP GUI.
2. In the transaction code field, run the following transaction: `oaa3`.

   The **SAP ArchiveLink: Communications interface administration** page appears.
3. Click **Create**.

   The **New Protocol** dialog box appears.

   For example, when you define the protocol name as `DCTMALF`, this refers to OpenText Information Archive with ALF Viewer.
4. Type `version 0045` or `0046` for HTTP-based archiving.
5. Save your changes.

   The **Overview of a Protocol** window appears, displaying your newly defined protocol.
6. Double-click **Display Stored Document**.

   The **Overview of a Protocol** window appears, showing a list of Document Classes.
7. Select the * row, which indicates all document classes.
8. Click the pencil icon.

   The **Overview of a Protocol** dialog box appears.
9. Type `OPEN` in the **Communication Type** field.

10. Press **Enter**.
11. Type `DCTM` in the **Application** field.
12. Click **Continue**.
13. In the **Overview of a Protocol**window showing the list of Document Classes, select the ALF document class.
14. Click the pencil icon.

    The **Overview of Protocol** dialog box appears.
15. Type `R/3` in the **Communication Type** field.
16. Press **Enter**.

    No application is entered here.
17. Save your changes.

    You are returned to the **Overview of a Protocol** window.

    If you repeat steps 4 and 11, you should see that the * document class and the ALF document class are indicated as **Maintained Explicitly**. All other document classes should be indicated as **Not Maintained Explicitly**.

1.7.2 Customizing SAP document classes

The following sections describe how to archive document classes from SAP. The customization of each SAP setup is different depending on the individual class of each archived document.

The document classes are as follows:

- ALF
- FAX
- OTF
- REO

Note: You must configure each of the following transaction codes for all document classes:

- `oac0`
- `oac2`
- `oac3`

1.7.2.1 oac0 - Configuring content repositories

SAP allows you to define a content repository (previously known as a logical archive). This content repository is mapped to OpenText Information Archive using an ArchiveId. The ArchiveId allows the system to link to a set of rules, such as content folders, OpenText Information Archive document types, and so on in the repository. It is recommended that you store each document type in a separate archive with a unique ArchiveId.

Each archive has an ArchiveId that defines what actions (types, folders, retention, and so on) are applied on an SAP document when it is archived.

For more information, see "oac0 – Defining a logical ArchiveID in SAP" on page 32.

1.7.2.2 oac2 - Defining document types

You must select which document types you will be archiving, and correlate those to SAP document classes, such as FAX, ALF, OTF, and REO, The list of document types provided by SAP, when using transaction OAC2, are the templates to be used for categorizing your SAP documents for archiving. You can customize the standard document types by using the SAP XYZ naming conventions.

- FAX documents are documents that will be scanned and linked to SAP.
- ALF are reports produced and linked from SAP.
- OTF are outgoing SAPSCRIPT documents produced and linked from SAP.
- REO documents are data archive files which are compressed by the SAP ADK and stored externally in that format.

To define document types for SAP:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   ```
   oac2
   ```

3. Select **New Entries** to define a new document type.

   You can also copy existing document types and edit their settings to create your new document type.

4. When you are modifying an existing document type, use **Change and Details**.
5. Enter information in the **Description** and **Doc. class** fields as required for your configuration.
6. When you have finished selecting document types, you must Save your changes.

1.7.2.3 oac3 - Defining links

Using the link table, you can categorize SAP documents to define what combination of SAP object type (such as SOOD) and SAP document type (such as GENPRILIS) are stored and linked to which archive.

The SAP document type is linked to the SAP document class, previously known as SAP document type (such as ALF, OTF, REO), which determines how the document is produced and linked. For example, the ALF class refers to reports, FAX to scanned documents, and OTF to outgoing SAPSCRIPT documents.

To define links:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   ```
   oac3
   ```

   The **Display View "Links for Content Repositories": Overview** screen appears.
3. Select **Table View > Display > Change**.
4. Fill in the following information in the **New Entries** table:

   Table 1-3: Field descriptions of New Entries table

Field name	Value
Object Type	Use legitimate object type such as SOOD for Print Lists, and BKPF for accounting documents
Document Type	The SAP doc type as defined in OAC2 above such as GENPRILIS
S (Status)	Type an "x" in this column
Arch. ID	Type the archive ID of a newly created archives, for example, AA
Link	Type the name of the link tables where links will be maintained. TOA01-TOA03
Ret Prd.	0

5. Save your changes.

1.7.2.4 Verifying the installation of holdings

You can verify if the holdings are installed correctly as follows:

1. Open the IA Web App.
2. Go to **Applications** and click **SAP applications**.
3. Click **Holdings** and you should find the following Holdings installed:

   - **SAPConnector**
   - **SAPBusinessMetaData**
   - **SAPArchiveId**
   - **SAPComponent**

   This is a primary verification to check if the holdings are installed correctly. Once you identify these folders, you have to perform archiving of a print list as listed in "Archiving a Print List" on page 50.

1.7.3 Testing the archiving process using a Print List

Once you have completed the configuration procedures described previously in this guide, you should test the archiving process by:

- Testing the connection between SAP and the OpenText Information Archive for SAP connector.
- Creating a document type in SAP.
- Archiving a Print List.
- Displaying the archived Print List in the SAP GUI.

1.7.3.1 Creating a document type in SAP

To fully test the archival and retrieval cycle, a simple Print List can be configured by following the instructions below:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   `oac3`

   The **Display View "Links for Content Repositories": Overview** screen appears.
3. Select **Table View > Display > Change**.
4. From the list, select any item with the ObjectType **SOOD**.
5. Click the **Copy As** icon and run.
6. Define a new document type using the `oac2` transaction code.

   For more information, see ???

7. In the **Storage system** field, enter the name of the logical ArchiveId where the documents will be archived, as described in "oac0 – Defining a logical ArchiveID in SAP" on page 32.
8. Save your changes.

1.7.3.2 Testing the connection between SAP and OpenText Information Archive

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   `\oac0`
3. In the **Display Content Repositories: Overview** page, click on any content repository whose **Storage type** is HTTP content server.
4. In the **Display Content Repositories: Detail** page, click the **Test connection** button.

   An information message appears at the bottom of the page, indicating that the connection test has been successful.

1.7.3.3 Archiving a Print List

To archive a Print List:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   f.21
3. Click **Execute**.
4. In the **List of Customer Open Items** page that appears, select **List > Print**.

   The **Print Screen List** dialog box appears.
5. Click **Properties**.

   The **Spool Request Attributes** dialog box appears.
6. In the **Parameters name** column, click **Output Options**.
7. From the **Storage Mode** list box, select **Archive only**.

   Note: Whenever you edit a parameter in this dialog box, we recommend that you select **Show Selected Print Parameters on Initial Screen**. This ensures that the customized settings are available in the initial **Print Screen List** dialog box.
8. Verify that the **Object Type** and **Document Type** fields contain the correct values, as configured in the **New Entries** table, as described in "oac3 - Defining links" on page 48. Pick the following values, for example: `SOOD GENPRILIS`.

If your Document Type is incorrectly configured, refer to the following procedures for configuring document types in SAP archives: "oac2 - Defining document types" on page 47 and "oac3 - Defining links" on page 48.

9. Type additional information in the **Information** field.

   This label may be anything you want; for example, your initials.

10. Optional Type a description of the Print List in the **Text** field.

11. In the **Parameter name** column, select **General attributes > Time of printing.**

12. From the **Time of print** list box, select **Print out immediately**.

13. Click **Continue** in the **Spool Request Attributes** dialog box.

14. Click **Continue** in the **Print Screen List** dialog box.

15. To verify that the Print List is in the archiving queue, run the following transaction in the transaction code field: `oam1`.

    The **ArchiveLink Monitor** screen appears.

    The Print List is now queued, waiting for the scheduler to pick it up and transfer it to the archive.

    "oaat – Scheduling jobs" on page 43 contains information about configuring the ArchiveLink scheduler.

16. To accelerate the transfer of the Print List from the queue to the archive in the repository, click **Storage Queue**.

    The **Queue: Content server (CARA)** screen appears. You should now see an entry for the newly-archived Print List in this screen.

17. To view the archiving parameter details, double-click the Print List entry.

    The **Archiving request** window opens.

    Verify that the details are correct

18. To archive the Print List in an archive located in a repository, click **Execute**.

    You should see a confirmation message that the queue has been processed.

    If no errors occur, you can continue testing by displaying the archived Print List.

1.7.3.4 Displaying an archived Print List in the SAP GUI

To display the newly-archived Print List in SAP GUI:

1. Open the SAP GUI.
2. Run the following transaction in the transaction code field:

   `\oadr`
3. To find a particular Print List, type appropriate search parameters in the appropriate fields.

   The **ArchiveLink: Hit List for Stored Print Lists** page appears.
4. Double-click an item in the list to select it.
5. Select **Print List > Display From Content Server**.

   SAP runs the **GET** command to retrieve the document from the OpenText Information Archive archive. The retrieved document is displayed in the SAP GUI and has the DocumentID of the Print List in OpenText Information Archive.

1.7.3.5 Adding metadata to archived documents in OpenText Information Archive

The ability to add SAP Business object details to all archived documents in OpenText Information Archive allows you to view and verify the documents of SAP objects such as Sales Orders, Finance Accounting, or Material management invoices. These are all SAP objects that have been archived using the OpenText Information Archive for SAP connector.

To view business object details from SAP, the OpenText Information Archive for SAP connector runs the following two BAPI functions:

- `ARCHIV_GET_CONNECTIONS`
- `SWO_GET_KEYFILEDS_FROM_ID`

Adding metadata to archived documents

There are two ways to add metadata to archived documents in OpenText Information Archive:

- Synchronous
- Asynchronous using the `QuartzJobScheduler`.

The `iasap.docs.update.metadata` property

The `iasap.docs.update.metadata` property acts as a flag that is used when updating a document's metadata. This property can be set to one of the following values:

- `No`

This value indicates that no metadata is updated by the connector.

- `Sync`

  This value causes the connector to update the metadata along with the original document during the ingestion into OpenText Information Archive of the document.

- `Async`

  This value causes the connector to update the object's metadata when the update job runs. The update job is scheduled by setting the `iasap.metadata.job.schedule` property and using the MBean server.

MBean server

`MBean` and `IAMetaDataMBean` are used to schedule `IAMetaDataJob` for reading metadata from SAP and adding it to documents in OpenText Information Archive. When the `iasap.docs.update.metadata` property has been set to `Async` then `MBean` is registered with the server to run updating jobs, and it triggers the `IAMetaDataJob` according to the schedule time set in the `al.properties` file.

You can use the `iasap.metadata.job.schedule` property to set the scheduled time for the job to run. `IAMetaDataJob` is a Quartzscheduler based job that allows you to use the quartz or cron tab notation to create expressions that set the run schedule time.

These notations allow up to seven parameters separated by a space character. The format is as follows where each star character below represents one of the seven parameters:

```
Seconds Minutes Hours Day-of-Month Month Day-of-Week Year (optional field)
* * * * * * *(Year optional)
```

Here are some examples that show you how to work with the quartz or cron notation to set a scheduled run time:

Example 1-2: Trigger expression sample 1

Here's an example that shows you how to create an expression that triggers every 5 minutes:

```
"0 0/5 * * * ?"
```

Example 1-3: Trigger expression sample 2

Here's an example that shows you how to create an expression that triggers every 5 minutes, at 10 seconds after the minute. such as 10:00:10 AM, or 10:05:10 PM, etc.:

```
"10 0/5 * * * ?"
```

Example 1-4: Trigger expression sample 3

Here's an example that shows you how to create an expression that triggers at 10:30, 11:30, 12:30, and 13:30 every Wednesday and Friday:

```
"0 30 10-13 ? * WED,FRI"
```

Example 1-5: Trigger expression sample 4

Here's an example that shows you how to create an expression that triggers every half hour between the hours of 8:00 AM and 10:00 AM on the 5th and 20th day of every month. This trigger will not fire at 10:00 AM, it fires at 8:00 AM, 8:30 AM, 9:00 AM, and 9:30 AM.

```
"0 0/30 8-9 5,20 * ?"
```

Required settings to read metadata with SAP JCo

The following properties are used to run the SAP JCo API to call the BAPI functions required to read metadata with SAP:

```
ia.sap.hostname=<IP-address-of-Host>
ia.sap.systemid=C11
ia.sap.clientcode=800
ia.sap.sysnumber=00
ia.sap.langcode=EN
ia.sap.username=<someusername>
ia.sap.password=<somepassword>
ia.sap.group=
iasap.result.schema=urn:ia:en:xsd:sapconnector.1.0
iasap.metadata.holding.name=SAPConnector
```

SAP objects key fields and dynamic entries

The SAPConnector holding allows you to use SAP object key fields as custom and dynamic entries. Here is an example that shows you the schema:

```
<xs:element name="isMetaDataAvail" type="xs:boolean" minOccurs="0"/>
  <xs:complexType name="Custom">
    <xs:sequence>
      <xs:any minOccurs="0" maxOccurs="2048" processContents="lax"
          namespace="##any" />
    </xs:sequence>
  </xs:complexType>
```

Archive or update business data – real time archiving

Here are some important points to remember when you are working with real time archiving:

- In the `al.properties` file set values for all of the following properties:

```
ia.sap.hostname=<IP-address-of-Host>
ia.sap.systemid=C11
ia.sap.clientcode=800
ia.sap.sysnumber=00
ia.sap.langcode=EN
ia.sap.username=<someusername>
ia.sap.password=<somepassword>
ia.sap.group=
```

```
iasap.result.schema=urn:ia:en:xsd:metadata.1.0
iasap.metadata.holding.name=SAPBusinessMetaData
```

- Set the `iasap.docs.update.metadata` property to `Sync`.
- Enable `IAMetaDataSynchronousListener` in `iasap/WEB-INF/web.xml` as shown below:

```
<listener>
<listener-class>com.documentum.ei.al.jmx.mbeans.IAMetaDataSynchronousListener</
listener-class>
</listener>
```

- When any ArchiveLink created request is received by the OpenText Information Archive for SAP connector, before ingesting the original document to OpenText Information Archive, the EAS repository manger runs the `MetaDatRetrieve` utility class to get the key fields of the SAP business object of that document.
- `EASRepositoryManger` retrieves business metadata and makes updates in the `.pdi` file of the `.sip` file.
- The `.sip` file with updated business metadata is ingested into OpenText Information Archive along with the binary file of the document.

Archive or update business data – Asynchronous update of business metadata

Here are some important points to remember when you are making asynchronous updates to business metadata:

- In the `al.properties` file set values for all of the following properties:

```
ia.sap.hostname=<IP-address-of-Host>
ia.sap.systemid=C11
ia.sap.clientcode=800
ia.sap.sysnumber=00
ia.sap.langcode=EN
ia.sap.username=<someusername>
ia.sap.password=<somepassword>
ia.sap.group=
iasap.result.schema=urn:ia:en:xsd:metadata.1.0
iasap.metadata.holding.name=SAPBusinessMetaData
```

  In addition to the above properties, you must set the `iasap.docs.update.metadata` property to `Asynch`, and you must also set a trigger time in the `iasap.metadata.job.schedule` property for the `IAMetaDataJob` to run.
- `IAMetaDataMBean` schedules the job according to the scheduled time that was set in the `iasap.metadata.job.schedule` property.
- `IAMetaDataJob` identifies all documents in OpenText Information Archive that have metadata by using the status of `isMetaDataAvail` and the version fields of the document.
- Documents that have metadata are sent to `IAMetaDataRetriever`, which retrieves their respective custom fields. `IAMetaDataRetriever` is responsible for re-ingesting the updated custom fields along with the original binary file.

Retrieving metadata from SAP

Here are some points to remember when retrieving metadata from SAP:

- Using the `sapjco3.jar` file gives you the ability to run functions from JAVA classes.
- Call the `ARCHIV_GET_CONNECTIONS` function module to read metadata. It returns the following details:
  - SAP Object Id, which is a combination of the original object, company id, and fiscal year for a finance object. Similarly, for a material object it is the object id, material id, and the created date
  - SAP Object
  - Client id
  - Archive object type
  - Dynamic attributes (such as Finance invoice document, company code, fiscal will)
  - Created date
  - Format
- You can use the SAP Object and Object Id as parameters for the `SWO_GET_KEYFILEDS_FROM_ID` function, which returns the key fields of that object type for that object id.
- You can run each SAP function. In this case the run time is extended from 200 up to 500 micro seconds per run.

Note: You must have SAP Jco, which is an OS and hardware specific native library that can be downloaded from the SAP Service Market Place website.

Risks and limitations

The `SWO_GET_KEYFILEDS_FROM_ID` SAP ABAP function must be enabled for the remote-enable module to run by the OpenText Information Archive for SAP connector.

Not all fields are added as metadata. Only the key fields of the object are treated as custom fields while updating business metadata.

Example 1-6: Pdi.xml with metadata

```
<?xml version='1.0' encoding='UTF-8'?>
<documents xmlns='urn:ia:en:xsd:metadata.1.0'>
  <create>
    <compId>data</compId>
    <contRep>FF</contRep>
    <docId>TEST1</docId>
    <isMetaDataAvail>true</isMetaDataAvail>
    <custom>
      <OBJECT_ID> TEST1000000022001</OBJECT_ID>
      <COMPANY_CODE> TEST1</COMPANY_CODE>
      <AR_OBJECT> DOC</AR_OBJECT>
      <DOCUMENTNO> 1000000002</DOCUMENTNO>
      <FISICAL> 2001</FISICAL>
      <SAP_OBJECT> BKPF</SAP_OBJECT>
    </custom>
```

```
  </create>
</documents>
```

Using the IA Web App to search metadata

Here are some points to remember when searching metadata with the IA Web App.

- You must update the `pdi.xml` file with new business metadata fields. This allows you to view the business metadata fields in the search creation page.
- Create a new search with all of the required metadata fields and select those fields to retrieve them as part of your search results.

1.7.4 Configuring SSL for OpenText Information Archive for SAP

The OpenText Information Archive for SAP connector supports both HTTP and HTTPS protocols. This section details the configurations needed on both SAP and the OpenText Information Archive for SAP connector application server for Secure Socket Layer (SSL).

1.7.4.1 Prerequisites for SSL configuration

SAP HTTP

As of Release Web AS 6.20 / SAP GUI for Windows 6.20, SSL is supported in `SAPHTTP`. The latest version of `SAPHTTP` is available on the SAP Service Marketplace website. For more information, see SAP Note 164203.

SAP Cryptographic Toolkit

The SAP Java Cryptographic Toolkit must be installed to use SSL. For more information about SAP Cryptographic Library, see SAP Note 397175.

1.7.4.2 Configuring SSL for SAPHTTP

Configuring SSL for `SAPHTTP` requires the creation of an `SAPSSLC.pse` file and making `SAPHTTP` aware of the trust certificate.

Creating an `SAPSSLC.pse` file

The `SAPSSLC.pse` file contains SSL client information. Copy the SAP Cryptographic Library and the `sapgenpse` program into the directory where `SAPHTTP` is located.

Working with self-signed certificates

To create a self-signed certificate:

1. Use the following command to generate a `SAPSSLC.pse` file with the name `<saphost>` in the `c:\temp\` directory:

```
sapgenpse get_pse -noreq -p c:\temp\SAPSSLC.pse CN=<saphost>
```

The value for <saphost> can be a fully qualified host name or the IP Address of the SAP application server. When prompted for a PIN number, enter a blank PIN.

2. Use the following command to export the certificate for your new SAPSSLC.pse file:

```
sapgenpse export_own_cert -p c:\temp\SAPSSLC.pse -o c:\temp\sapserver.cer
```

You must import this certificate as trusted in the Application Server keystore as detailed in "Configuring SSL on the OpenText Information Archive for SAP application server" on page 59.

3. Use the following command to import the certificate of the Application Server for your SAPSSLC.pse file as a trusted certificate:

```
sapgenpse maintain_pk -a c:\temp\tomcat.cer -p c:\temp\SAPSSLC.pse
```

In the above command, tomcat.cer is the certificate of the Application Server. For more information on how to create the tomcat.cer certificate file, see "Configuring SSL on the OpenText Information Archive for SAP application server" on page 59.

Alternatively, the certificate can also be imported using STRUST.

4. Use the following command to list the certificates in the SAPSSLC.pse file:

```
sapgenpse maintain_pk -l -p c:\temp\SAPSSLC.pse
```

Working with certificates from a certificate authority (Verisign, Thawte, or Trustcenter)

Follow these steps to obtain a certificate from your Certificate Authority (CA):

1. Create an SAPSSLC.pse file with a Certificate Signing Request (CSR) using the following command:

```
sapgenpse get_pse -p c:\temp\SAPSSLC.pse -r <CERT_REQUEST.csr> CN=<saphost>
```

2. Submit the CSR to your CA to get your certificate <CERT_RESPONSE.cer>

3. Include the root and intermediate certificates of the CA in to the SAPSSLC.pse file using the following command. In this example, we assume that the CA Root certificate is encoded in the c:\temp\CARoot.cer file:

```
sapgenpse maintain_pk -a c:\temp\CARoot.cer -p c:\temp\SAPSSLC.pse
```

Similarly you can include the intermediate Certificate in the SAPSSLC.pse file.

4. Use the following command to import your own certificate into the SAPSSLC.pse file.

```
sapgenpse import_own_cert -p c:\temp\SAPSSLC.pse -c <CERT_RESPONSE.cer>
```

You can also use the STRUST transaction to import the certificate.

5. Include the Root and intermediate certificates of your Application Server into the SAPSSLC.pse file. For example, when the Application Server uses VeriSign Certificates, include the VeriSign Root certificate into the SAPSSLC.pse file by using this command:

```
sapgenpse maintain_pk -a c:\temp\TomcatRoot.cer -p c:\temp\SAPSSLC.pse
```

Similarly, you can include the VeriSign intermediate Certificate of your Application Server into the SAPSSLC.pse file.

Configuring SAPHTTP

To configure SAPHTTP:

1. Copy the SAPSSLC.pse file to the following location:

```
<SYSTEM NAME>/<INSTANCE NAME>/sec
```

Caution

- SAPHTTP also requires the SAPSSLS.pse file to work. If the SAPSSLS.pse file is missing (for example when you do not have Web AS 6.10) from the above directory, copy the SAPSSLC.pse file into the same directory as the SAPSSLS.pse file.
- A file named ticket must also be present in the same directory as the SAPSSLS.pse file. If the ticket file is not present, copy this file from the SAP Cryptographic Library into the directory.
- You must also copy the sapcrypto.dll from SAP Cryptographic Library in to this directory.

2. Copy the SAPSSLC.pse, SAPSSLS.pse, ticket, and sapcrypto.dll files into the SAP GUI directory and also into the <SYSTEM NAME>/SYS/exe/run directory on the SAP server.

1.7.4.3 Configuring SSL on the OpenText Information Archive for SAP application server

The steps described below are for the Tomcat Application Server. Similar changes need to be made for the other supported application servers.

Enabling SSL on Tomcat

To enable SSL on Tomcat, add or uncomment the following lines in the <tomcat_home>\conf\server.xml file:

```
<Connector port="<SSL Port>"> maxThreads="150" minSpareThreads="25”
  maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" debug="0" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
```

```
  keystoreFile="<="<${user.home}/.keystore>"
  keystorePass="<keystore password>" keyAlias="<Certificate Alias>" />
```

Creating keystore

Use the following command to create a keystore using the JDK keytool:

```
%JAVA_HOME%\bin>keytool -genkey -alias <Certificate Alias> -keyalg RSA
Enter keystore password: changeit
```

Note: This is not your personal name.

```
What is your first and last name?
[Unknown]: <enter_host_name or IP Address of host where tomcat is running>
What is the name of your organizational unit?
[Unknown]: <enter_OU>
What is the name of your organization?
[Unknown]: <enter_Org>
What is the name of your City or Locality?
[Unknown]: <enter_City>
What is the name of your State or Province?
[Unknown]: <enter_State>
What is the two-letter country code for this unit?
[Unknown]: <enter_Country>
Is CN=tomcat-host, OU=CMA, O=OpenText, L=Sydney, ST=NSW, C=AU correct?
[no]: yes
Enter key password for <certificate_alias>
(RETURN if same as keystore password):
```

Note: The private key password and keystore password should be the same. Verify that a file called `.keystore` has been created in the user home directory (`C:\Documents and Settings\dmadmin\.keystore`).

Generating certificate from keystore – Self-signed certificates

This section describes the details of creation of a self signed certificate from the tomcat keystore. The certificate exported to the file would in-turn be imported to the `SAPSSLC.pse` file.

Create Self-signed certificate from the keystore

Create a certificate from the Java keytool using the following command:

```
keytool -selfcert  -alias <certificate_alias>  -validity <valdays>
```

Enter keystore password when prompted for the password.

Export certificate from the keystore to a file

Export the certificate to a file using the following command:

```
keytool -export -alias <certificate_alias> -file “C:\temp\tomcat.cer”
```

The command stores the self signed certificate in the `tomcat.cer` file.

You must add this certificate to the trusted list of `SAPSSLC.pse` as detailed in the Configuring SSL for `SAPHTTP` section.

Import SAP certificate to the keystore

Add the self signed certificate of the `SAPSSLC.pse` file into the trusted list of Tomcat’s keystore.

Use the following command for a certificate that is encoded in the `sapserver.cer` file:

```
keytool -import -alias sapserver -file C:\temp\sapserver.cer
```

Create CSR from the keystore

To generate the certificate from a Certificate Authority (such as VeriSign or Thawte), use the following command to create a certificate request:

```
keytool -certreq -keyalg RSA -alias <certificate_alias> -file c:\temp\certreq.csr
```

Import certificates to the keystore

To import the certificates to the keystore:

1. Send the CSR to the Certificate Authority (such as VeriSign or Thawte) to get a certificate.
2. Before importing the certificate to the keystore, include the root and intermediate certificates in the keystore. For example, for Verisign trial certificates.

   ```
   keytool -import -alias verisign-trial-root -trustcacerts -file c:\temp
   \Verisign-trial-root.txt
   ```

   ```
   keytool -import -alias verisign-trial-intermediate -trustcacerts
   -file c:\temp\Verisign-trial-intermediate.txt
   ```

3. Use the following command to import the certificate reply to the keystore:

   ```
   keytool -import -alias <Certificate Alias> -file C:\temp\Verisign-
   signed-cert.txt
   ```

4. Include the Root and intermediate certificates of the CA, from whom you received the certificate for the `SAPSSLC.pse` file, into the trusted list of Tomcat's Keystore.

   Use the same commands that were used above to import the root and intermediate certificates of VeriSign.

Now, both the Application Server and `SAPHTTP` are ready to work with the HTTPS protocol. Proceed with the OpenText Information Archive for SAP connector deployment and archiving.

1.7.4.4 Configuring SAP Content Repository to enable SSL for Archive Link communication

You must complete the following steps to configure the SAP Content Repository and enable SSL for Archive Link communication:

1. Use the `oac0` transaction code and select the content repository on which SSL must be enabled.
2. Send the certificate from the SAP content repository to OpenText Information Archive in non-SSL mode.

   Note: The distribution of certificates from the `oac0` transaction code is intended to send the signer's certificate to the communication partner in a system landscape for an SSL application. It is used for verification of data that is sent through Archive Link using the PKCS7 standard. `Send Certificate` in transaction `oac0` can only be done in non-SSL mode. It is a one time job.

3. Edit the Content Repository to mention the SSL port, then do perform the following steps:
   a. Enter `%HTTPS` in the transaction code field.
   b. Select `HTTPS` required in both `HTTPS` on front end and `HTTPS` on backend.
   c. Save the configuration.

1.8 Using the OpenText Information Archive for SAP connector

1.8.1 Working with archived documents in the SAP GUI

1.8.1.1 Archiving documents

The OpenText Information Archive for SAP connector must be configured before you can begin archiving documents. For more information, see Configuration and administration.

To archive documents using the SAP GUI:

1. Start the SAP GUI and connect to SAP R/3.
2. In the transaction code field, run any ArchiveLink enabled transaction code to archive documents using the SAP GUI; for example:

   ```
   fbo3
   ```

3. Enter the search criteria and press **Enter**.
4. Select **System > Services for Object**.

5. Click the **Create** icon.
6. From the pop-up menu that appears, select **Store business document**.

   The **Archive from Frontend** dialog box appears.
7. In the **Document Type** pane, double-click on the required document type.

   This determines the document type that you intend to archive.

   The **Storing Files in Documents** dialog box appears.
8. Select the document from your file system.
9. Click **Open**.
10. Click the **Enter** icon.

    The document is archived according to the configuration of the document type.

For more information, see the SAP Help Portal website.

1.8.1.2 Finding and viewing linked documents

To view a linked document in the SAP GUI:

1. Start the SAP GUI and connect to R/3.
2. Enter the transaction code that shows all newly released documents, or the transaction code that shows the material basic data.
3. Select the tab for document data.

   In some SAP modules, such as the Customer module, you can select **Extras > Document Data** from the menu. The Linked document DIR is displayed.

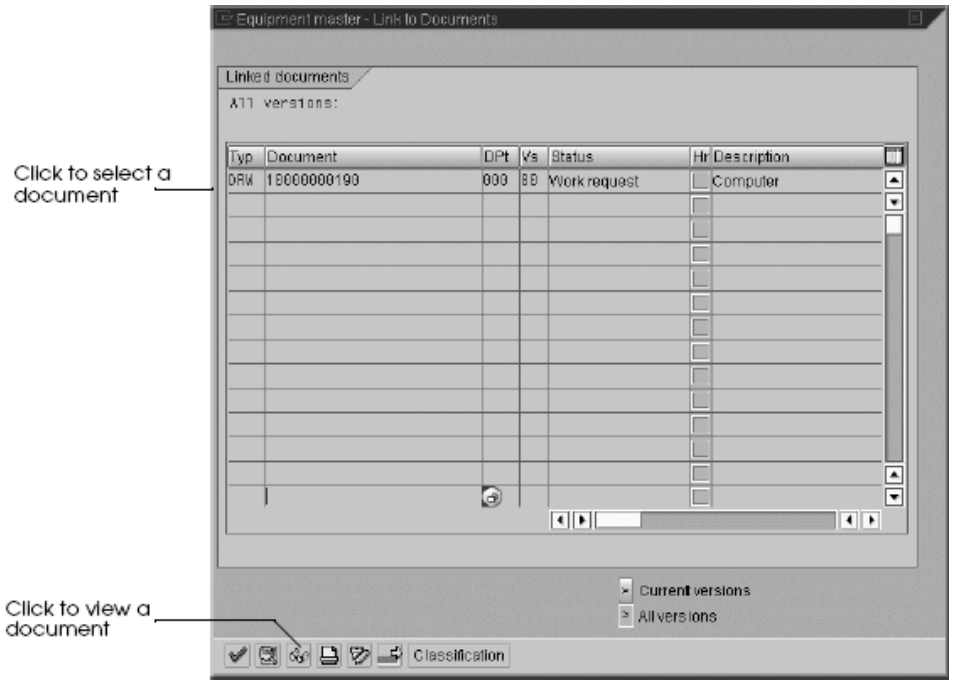


Figure 1-1: Equipment Master screen

4. Select the document.

5. Click **Display**.

   The document is displayed using the viewer application.

More information about the Document Finder component of SAP, see the SAP Help Portal website.

1.8.1.3 Finding and displaying Print Lists

To find and display Print Lists in the SAP GUI:

1. Start the SAP GUI and connect to your R/3 server.

2. In the transaction code field, run the following transaction:

   ```
   oadr
   ```

3. To find a particular Print List, enter your search parameters in the appropriate fields.

   The SAP GUI returns a list of items that matched your search parameters.

4. Click an item in the list to select it; then click **Display from Archive**.

   The archived Print List is retrieved from the repository and displayed in the SAP GUI.

When a Print List is displayed in the SAP GUI, you have the following options:

- **Adding and editing notes**
- **Free search**

  Free search refers to a full-text search.
- **Attribute search**

  Attribute search can be used for indexed Print Lists.

For more information on Print Lists in SAP, see Print Lists on the SAP Help Portal website.

1.8.2 Searching

1.8.2.1 Searching for documents in OpenText Information Archive

Follow these steps to search for documents in OpenText Information Archive:

1. Log in to the IA Web App.
2. Select **SAPConnector**.
3. Select **Create New** from the **Add Search** list box.
4. Optional Select **Import from ZIP file** to import an existing search zip file from the holdings that are already installed.
5. Enter a name for your new search in the **Search Name** field.
6. Select **SAPConnector-aic** in the **Archival Collection** list box.
7. Click **Next**.
8. Select **Show in Form** for all fields and click **Next**.
9. Select the items that you want included in the search results.
10. Click **Finish**.

Your new search is ready for use. When the document that you are searching for is in the OpenText Information Archive repository, it will also be present in the search results. Otherwise no result for that document is displayed.

1.8.2.2 Searching metadata with the IA Web App

Here are some points to remember when searching metadata with the IA Web App.

- You must update the `pdi.xml` file with any new business metadata fields before performing this type of search. This allows you to use the business metadata fields in the search creation page.
- Create a new search with all of the required metadata fields and select those fields to retrieve them as part of your search results.

Performing a metadata search

To perform a metadata search:

1. You must import the `Search` zip file that is present in the SAPConnector holding. Login to OpenText Information Archive and select the **SAPConnector** application.
2. Import the search file using **Add search > Import from zip file**.
3. Select and open the `Search SAP document from docid.zip` file.

   You should see the `metadata` and `SAP document from docid` search options that were imported.
4. You can view the search results using the **Search SAP document from docid.zip** option.

   You can click on the **+** symbol to view the metadata of each corresponding document listed.

1.9 Appendix: Advanced installation options

1.9.1 Installing the Apache and Tomcat connectors

The OpenText Information Archive for SAP connector may be installed on a Tomcat server where Apache and Tomcat are configured to use load balancing between multiple Tomcat sessions. When your installation scenario is based on a load-balancing configuration, in addition to installing Apache, you must download (from the Web) and install `mod_jk`, which is a plug-in that handles all communication between Apache and Tomcat.

1.9.2 Configuring Apache and Tomcat for load balancing

Use the following steps as broad guidelines while configuring Apache and Tomcat for advanced scalability.

- Obtain the installation file for a supported version of Tomcat from the Web and save it to a location where you can retrieve it.
- Install the supported Tomcat application server.

  Depending on your load-balancing requirements, you can install Tomcat on two or more hosts.
- Obtain the installation files for Apache from the Web and save them to a location where you can retrieve them.

  Install Apache.
- Obtain the installation files for the `mod_jk` plug-in and perform the installation. The `mod_jk` plug-in handles the communication between Apache and Tomcat.
- In Apache:
  - Edit the Apache `httpd.conf` configuration file.

    Enter the path to the `mod_jk` plug-in.
  - Create a `workers.properties` file. In this file:
    - The number of workers that you create should be equal to the number of Tomcat instances that have been installed.
    - Specify the percentage of load that will be balanced by each worker.
- In each Tomcat instance, edit the `server.xml` file to disable the stand-alone configuration.

  This ensures that each Tomcat instance is a worker that is running to perform the Servlet requests that are coming from Apache. You now have a working load-balancing configuration using Apache and Tomcat.

1.10 Appendix: Troubleshooting

1.10.1 Connection issues between SAP and the OpenText Information Archive for SAP connector

If you see the following error in the `alserver.log` file, SAP probably closed the connection to or stopped listening to the OpenText Information Archive for SAP Connector: "An existing connection was forcibly closed by the remote host." SAP normally does this when it reaches its maximum time-out value for ICM/HTTP resources, which is usually 60 to 300 seconds.

Possible causes for timing out include the following:

- There are no threads available to process the request on the application server that hosts the OpenText Information Archive for SAP connector.
- The requested resource (for example, a document) is very large in size.
- There is a network delay.

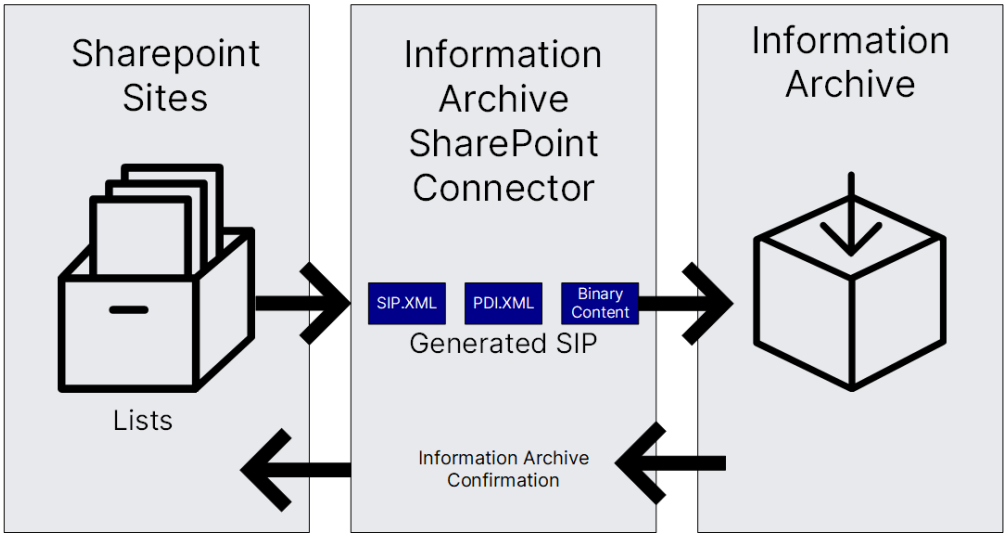
Possible solutions include the following:

- Increase the maximum number of threads on the application server.
- Increase the JVM memory on the application server.
- Prevent any network delays from occurring during processing.
- On SAP, change the maximum time-out value to -1 (unlimited time) to listen for the response.

Chapter 2

OpenText Information Archive SharePoint connector

OpenText Information Archive SharePoint connector (SharePoint Connector) is a command-line data extraction and transformation utility that extracts list items from SharePoint sites and generates SIPs for OpenText Information Archive to ingest. OpenText Information Archive can then send back a confirmation to SharePoint. The following diagram illustrates how SharePoint connector works with SharePoint and OpenText Information Archive.



SharePoint connector extracts list items from SharePoint sites, saves the attributes of extracted items in the resulting PDI file (`eas_pdi.xml`), and creates the SIP descriptor (`eas_sip.xml`) based on the settings in the configuration file.

You can use SharePoint connector with SharePoint 2010, 2013, 2016, and SharePoint Online as a part of an Office 365 suite.

The SharePoint connector distribution package contains the following components:

Components	Description
`InfoArchive.SharePoint.Connector.exe`	Connector executable file.

Components	Description
XSD files	Schemas to validate the SIP and PDI files produced by the connector. The SharePoint holding is delivered with OpenText Information Archive (`tools > applications` folder).
DLL files	Class libraries for running the executable file.
`config` folder	Configuration file template and CAML query template.

2.1 What's new

In OpenText Information Archive SharePoint connector, the structure of PDI files has changed from release 3.2, 3.2 patch 8, 4.0, 16 EP3 and 16 EP4:

Starting with version 3.2, the `Versions` element of a PDI file displays all versions of a document, including the latest version. The value in the `EASFile` element(s) of previous version(s) is prefixed with a version label, for example, `2.0_25c111e0-22a3-4cf1-b968-c5101284e666.rtf`. The value in the `EASFile` element of the last version does not have the prefix, for example, `25c111e0-22a3-4cf1-b968-c5101284e666.rtf`.

2.1.1 PDI structure changes since Version 3.2, Patch 8

2.1.1.1 Field display name

The field's display name is added to the XML element:

```
<Last_x0020_Modified displayName="Modified"> 2015-08-25T11:13:19Z </Last_x0020_Modified>
```

Structure before version 3.2 patch 8:

```
<Last_x0020_Modified>2015-08-25T11:13:19Z</Last_x0020_Modified>
```

2.1.1.2 User objects

```
<Author displayName="Created By" datatype="People">
  <People>
    <ID>12</ID>
    <UniqueId displayName="Unique Id">z19499c1-ab11-305z-1zz1-1573z92214zz</
UniqueID>
    <Title displayName="Name">John Doe (CORP/office)</Title>
    <EMail displayName="Work email">john.doe@johndoe.com</EMail>
  </People>
</Author>
```

Structure before version 3.2 patch 8:

```
<Author>
  <ID>12</ID>
    <UniqueId displayName="Unique Id">z19499c1-ab11-305z-1zz1-1573z92214zz</
UniqueID>
    <Title displayName="Name">John Doe (CORP/office)</Title>
    <EMail displayName="Work email">john.doe@johndoe.com</EMail>
</Author>
```

2.1.1.3 Lookup field

```
<LookupField name="Lookup_x0020_Col1" displayName="Column1">
  <LookupValue id=”572”>0</LookupValue>
</Lookup_x0020_Col1>
```

Structure before version 3.2 patch 8:

```
<Lookup_x0020_Col1 xmlns=" ">
  <LookupValue>0</LookupValue>
  <LookupId>572</LookupId>
</LookupField>
```

2.1.1.4 Lookup field (List Of)

The `list_url` attribute refers to the lookup field's associated list.

```
<LookupField name="Lookup_x0020_Column1" displayName="Column1" list_url="/site/
List">
  <LookupValue id="20">value</LookupValue>
  <LookupValue id="7">another value</LookupValue>
</LookupField>
```

Structure before version 3.2 patch 8:

```
<Lookup_x0020_Column1 xmlns=" ">
  <LookupValue>value</LookupValue>
  <LookupId>20</LookupId>
</Lookup_x0020_Column1>
<Lookup_x0020_Column1 xmlns=" ">
  <LookupValue>another value</LookupValue>
  <LookupId>7</LookupId>
</Lookup_x0020_Column1>
```

2.1.1.5 Managed items

Before version 3.2 patch 8, the following fields were not extracted:

```
<ManagedMetadata name="Column1_x" displayName="Column1 Managed Metadata">
  metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="City" TermGuid=
"84569b41-7163-4626-a471-812ed976886d" WssId="11"/>
  <metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="Keyboard"
TermGuid
="dbbd37b9-64dc-499f-9ed3-cbc312a17d43" WssId="8"/>
  <metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="Desktop"
TermGuid=
"72ca1aab-ef35-4b87-8b6d-ed26ae4a88ba" WssId="12"/>
  <metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="Desk"
TermGuid=
"c7fbce97-20c0-46b2-ad31-5e434201b7c7" WssId="13"/>
  <metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="Office"
TermGuid=
"de24c67e-ed73-4ca7-b0a9-800356fd355c" WssId="14"/>
  <metadata _ObjectType_="SP.Taxonomy.TaxonomyFieldValue" Label="Paris"
TermGuid=
"1fdfb8d0-ef83-4e86-8d1e-544387d0fd3f" WssId="15"/>
</ManagedMetadata>
```

2.1.2 PDI structure changes since Version 4.0

2.1.2.1 Multiple choice check box

The Multiple Choice Check Box is now extracted:

```
<Value_x0020_MultiChoice displayName="Value MultiChoice">Enter Choice #
1,Enter Choice #2,Enter Choice #3,Enter Choice #4,Enter Choice #5
</Value_x0020_MultiChoice>
```

2.1.2.2 HyperLink type element

The Description field is added to the HyperLink Element:

```
<attributeUrl displayName="Document Link" datatype="Hyperlink">
  <Description>this is the description</Description>
  <Url>https://mysite.sharepoint.com/myssite/_layouts/15/
      DocIdRedir.aspx?ID=11ZABCDP5ZZ2-21-5</Url>
</ attributeUrl >
```

2.1.3 PDI structure changes since Version 16EP3

2.1.3.1 Site and list

```
<Site url="/Project/Team1">Project Team1</Site>
<List>Documents</List>
```

2.1.3.2 User permissions

SharePoint Online:

```
<Permissions>EmptyMask,ViewListItems,AddListItems</Permissions>
```

SharePoint 2010:

```
<Roles>
  <Role Name="Limited Access" Type="Guest" />
</Roles>
```

2.1.4 Changes since Version 16EP3 Patch 1

2.1.4.1 CSOM changes

For 16EP3 patch 1, the Client-Side Object Model (CSOM) API was upgraded.

The SharePoint Connector requires the latest Client Components SDK (Version 16), which can be downloaded from the Microsoft website.

2.1.4.2 Version 16 EP4: Enabling the wizard on the SharePoint holding schema

To create new holdings using the holding wizard (16EP4), the PDI produced by the connector needs to be re-factored with an XSL template. A sample XSL template and matching XSD schema are delivered with the connector to transform the PDI file. The XSL moves some common elements to new `Objects/Object/base` node and leaves all the other elements (SharePoint internal and custom attributes) to the new `Objects/Object/other` node.

To ingest the data and use the holding wizard, this XSL has to be applied and set in the configuration file (refer to the `config > Sample.config` file provided with the connector (notably, the following `xslt` and `xsd` configuration properties:

```
<add key="xsd" value=".\resources\sharepoint_1.0.xsd" />
<add key="xslt" value=".\resources\SharePointToIA.xsl" />
```

The `Objects\Object\base` element contains: `Type`, `Title`, `Author`, `Editor`, `Document (versions)`, `FileDirRef`, `FileLeafRef`, `FileType`, `Site (url and name)`, `List`, `Modified` and `Created dates`.

To move elements from `Objects\Object\other` to `Objects\Object\base`, edit the above XSL and XSD.

2.2 Configuring and running SharePoint connector

2.2.1 Prerequisites

Make sure you install the following components:

- Microsoft SharePoint Client Components SDK
- Microsoft .NET Framework 4.5

Note: SharePoint connector does not have to run on an OpenText Information Archive host. However, you must install the SharePoint sample holding on the OpenText Information Archive host where you want to archive the SIPs generated by SharePoint connector.

The SharePoint connector is independent of the IA Server. We strongly recommend using the latest version of the SharePoint connector, even when running an older version of OpenText Information Archive. You can generate SharePoint PDI files with the 21.2 version and ingest the files with any version of OpenText Information Archive.

2.2.2 Preparing configuration files

Configuration files provide all the properties related to SharePoint connector run, including properties for establishing a connection with a SharePoint site, properties for formatting the generated SIPs, properties for locating items to extract, and so on. You can find a sample configuration file in `Sample\Sample.config`.

You have two options when preparing configuration files for a SharePoint site:

- *One local file*: All properties saved in a local file
- *One local file and one remote file*: Properties for establishing a connection saved in a local file, and other properties in a remote file on a SharePoint site

If you set the `doTracking` property to `TRUE` in the configuration file, you must save configuration files in the `config` folder of the SharePoint connector package. If `doTracking` is `FALSE`, you can save configuration files elsewhere in the SharePoint connector package.

2.2.2.1 Preparing one local configuration file

If you plan to use a single local configuration file for SharePoint connector, the file must contain *all* properties for running SharePoint connector.

Complete the following steps to prepare a local configuration file:

1. Open `Sample.config` in a text editor.
2. Locate the `appSettings` element.
3. Modify the following properties:
   - `host`: The hostname of your SharePoint site. If SharePoint is hosted on a port other than `80` (`http`) or `443` (`https`), you must specify the port number as a part of the host. For example, `myhost:8080`.
   - `loginName`: The name you use to log in to the SharePoint site
   - `workingDir`: The directory where you save generated SIPs
   - `site`: A relative URL to a SharePoint site. For example,
     ```
     <add key="site" value="/sites/demo" />
     ```

     Note: These properties are the minimal set of the properties you need to modify. Refer to the Configuration properties chapter for more information about configuration properties.
4. Optional Modify the properties in the `log4net` element to change the logging settings.
5. Save the file.

2.2.2.2 Preparing one local configuration file and one remote configuration file

The local configuration file used with a remote file should at least contain the following properties:

- `configurationFileUrl`
- `authenticationMode`
- `domain`
- `protocol`
- `host`
- `loginName`
- `site`
- `encryptionKeysFolder`
- `password`

Note: The `configurationFileUrl` property, which points to a document on a SharePoint site, is only needed when a local configuration has a remote counterpart.

The remote configuration file must contain all other properties that are not specified in the local configuration file.

2.2.3 Custom queries

It is possible to define a custom query to extract specific items from the SharePoint lists targeted in the configuration file.

This custom query is defined in a dedicated file and is linked to the main configuration file as a `.caml` file. To be loaded, the `.caml` file name should contain the full configuration file name with the `.caml` extension and should be saved in the same folder than the configuration file:

- Configuration file: `./config/Sample.config`
- Custom query file: `./config/Sample.config.caml`

The `caml` query file is unique to each configuration file and can contain only a single query. If the configuration file targets more than one list, this custom query will be applied to all the lists.

The query syntax is SharePoint Collaborative Application Markup Language (CAML). The SharePoint connector `.caml` query file contains the WHERE clause of the CAML query. A sample is provided with the connector in the `config` directory.

Scenarios

1. To extract documents from the 'Folder A/Folder B' subpath of the Documents list defined in the main configuration file (`lists` property):

```
.caml file query:
<Contains>
  <FieldRef Name='FileDirRef'/>
  <Value Type='Text'>Folder A/Folder B</Value>
</Contains>
```

   The connector will submit this query to the SharePoint list:

```
<View Scope='RecursiveAll'>
  <Query>
    <Where>
      <Contains>
        <FieldRef Name='FileDirRef'/><Value Type='Text'>
         Folder A/Folder B</Value>
      </Contains>
    </Where>
  </Query>
  <RowLimit>5000</RowLimit>
</View>
```

   Note: The query submitted to the SharePoint server is added to the log file generated during the extraction.

   The `<View Scope='RecursiveAll'>` setting can be changed with the `QueryViewScope` configuration property (refer to the lists section for more information).

2. To extract all the PDF documents from the Documents list defined in the main configuration file:

```
<Eq>
  <FieldRef Name="File_x002O_Type"></FieldRef>
  <Value Type="Text">pdf</Value>
</Eq>
```

3. To extract all the documents except the PDF files:

```
<Neq>
  <FieldRef Name="File_x002O_Type"></FieldRef>
  <Value Type="Text">pdf</Value>
</Neq>
```

4. To extract all the PDF documents from the Folder B subpath (two query constraints: folder path and file extension):

```
<And>
  <Eq>
    <FieldRef Name="File_x002O_Type"></FieldRef>
    <Value Type="Text">pdf</Value>
  </Eq>
  <Contains>
    <FieldRef Name='FileDirRef'/>
    <Value Type='Text'>Folder A/Folder B</Value>
  </Contains>
</And>
```

For more details on the CAML query syntax, refer to Microsoft SharePoint documentation.

2.2.4 Generating SIPs from a SharePoint site

The first time you run SharePoint Connector on a SharePoint site, you must perform the following steps:

1. Generating private and public keys
2. Encrypting a password
3. Extracting items from the SharePoint site

After you run SharePoint Connector for the first time on a SharePoint site, you only need to perform Step 3 for the subsequent runs on the same SharePoint site.

2.2.4.1 Generating private and public keys

Open a command prompt window and run the following command:

```
InfoArchive.SharePoint.Connector.exe --generateKeys -c
<config_file_relative_path>
```

If you save a configuration file `test.config` in the `config` folder, the relative path to the configuration file is: `.\config\test.config`.

Generated keys are saved in the folder specified by the `encryptionKeysFolder` property in the configuration file.

2.2.4.2 Encrypting a password

Open a command prompt window and run the following command:

```
InfoArchive.SharePoint.Connector.exe --pwd -c <config_file_relative_path>
```

SharePoint connector prompts you to enter the password. SharePoint connector encrypts and saves the password to the `password` property in the configuration file.

Note: You must generate keys before encrypting the password, or the keys are invalid. If you reset the keys after the password is encrypted and saved to the configuration file, you must reset the password.

2.2.4.3 Parallel site extractions

To extract data from a farm with a lot of sites, the connector can be configured to extract sites in parallel to reduce extraction time.

Set `maxParallelSiteExtractions` to a number from `1` to a reasonable limit (up to `30`). The value will depend on the SharePoint server and the server resources that run the connector.

2.2.4.4 Extracting items from SharePoint sites

Open a command prompt window and run the following command:

```
InfoArchive.SharePoint.Connector.exe -c <config_file_relative_path>
```

Generated SIPs are saved in the directory specified by the `workingDir` property in the configuration file.

Note: If the status of the latest extraction is not `STORAGE`, SharePoint connector does not proceed to extract items. To extract items that have been rejected or invalidated, use the following command line to force extract items:

```
InfoArchive.SharePoint.Connector.exe -c <config_file_relative_path> --forceExtraction
```

In that case, the rejected or invalidated items will not be extracted again unless they have been updated since the last extraction.

2.2.5 Tracking the status of extracted items

To avoid redundant extractions and ingestions, SharePoint connector creates tracking lists on the SharePoint site based on the `doTracking` setting in the configuration file.

The SharePoint account for performing tracking tasks must have **Manage Lists** permission with these permission levels: Full Control, Design, and Manage Hierarchy. For more information, see *On Premises SharePoint Server use permissions and permission levels* on the Microsoft website.

With tracking lists, the extraction and ingestion are incremental, which means extracted items are not included in the extraction scope before they are archived by OpenText Information Archive. Once an item is archived (tracking status = STORAGE), SharePoint connector brings the item back to SharePoint connector's work scope. If errors occur during the extraction, SharePoint connector removes the tracking items from tracking lists and returns an error code.

The generated list is similar to a database table, containing the following columns:

- **Title**: DSS ID
- **Version**: The version of the item
- **ListGuid**: The list GUID of the item
- **ListName**: The list name of the owning list
- **ExtractionDate**: Date/time when the item is extracted
- **ExecutionDate**: Date/time when SharePoint connector runs
- **ObjectId**: Object GUID
- **SIP**: SIP sequence number
- **Status**: The status of the item

- **Eia_Id**: Internal key used by SharePoint connector

An item can be in one of the following lifecycle statuses. Each status corresponds to a certain stage of an extracted item.

- **EXTRACTED**: An instance of SharePoint connector is extracting from, or the last run has been abruptly terminated.
- **TRANSFER**: SharePoint connector generates SIPs successfully.
- **RECEIPT**: OpenText Information Archive receives the SIP.
- **STORAGE**: OpenText Information Archive ingests the SIP.
- **REJECT**: OpenText Information Archive rejects the SIP.
- **INVALID**: OpenText Information Archive invalidates the SIP.

You can update tracking statuses in the following ways:

- Using the OpenText Information Archive confirmation files
- Running an update status command with a proper DSS ID and a sequence number specified

2.2.5.1 Enabling report generation for item extraction

The connector can generate a `.csv` file report at the end of the run. The report provides the number of successful and failing items, files, and attachments extracted. If the connector failed to extract items, a list of the failing items is provided in the report.

See the Configuration properties section to enable the report generation.

2.2.5.2 Using OpenText Information Archive confirmation files to update status

You can update item statuses using the confirmation files when you receive, ingest, reject, invalidate, or purge the owning AIP. Use the following arguments to update the status:

`--updateTrackingFile`
Update the status based on a confirmation file. For example, `InfoArchive.SharePoint.Connector.exe --updateTrackingFile <path_to_a_confirmation_file>`

`--updateTrackingFolder`
Update the status based on a folder containing confirmation files. For example, `InfoArchive.SharePoint.Connector.exe --updateTrackingFolder <path_to_the_folder_of_confirmation_files>`

The SharePoint connector looks for producer and SIP ID in the confirmation file. The Sip ID matches the ID with the DSS column in the tracking list. The producer identifies the local configuration file. It enables the connector to identify which configuration file to load in order to update the site and tracking lists. The producer value is generated during the data extraction, it contains the configuration filename. The configuration filename length should not exceed 32 (length without the `.config` extension).

Enabling confirmation on the holding with the default settings produces confirmation files with all the required fields. Settings can be changed to reduce the output with the following required fields:

- `producer`
- `id`
- `seqno`
- `conf_type`

SharePoint connector looks up the SIP ID in the confirmation file and matches the ID with the DSS column in the tracking list. When a match is found, SharePoint connector performs the following tasks:

- Updating the corresponding **Status** column in the tracking list based on the confirmation file. For example, the status is updated to <*STORAGE*> based on the following confirmation event.

  ```
  <eas_conf_type>storage<eas_conf_type>
  ```

- Renaming the confirmation file name extension to `.confirmed` if the update is successful.
- Renaming the confirmation file name extension to `.discarded` if the update fails or is not applied.

The tracking list defined in the configuration file is automatically excluded from SharePoint connector extraction scope.

2.2.5.3 Running an update status command to update status

In case confirmation files are not available, you can update items status by running an update status command. An AIP can be identified by its DSS ID and the sequence number. You can use the following arguments in the command:

`--updateTracking`
The type of the task.

`--dssId`
The DSS which the SIP belongs to.

`--status`
The target status. You must use one of the following values:

1. TRANSFER

2. RECEIPT
3. STORAGE
4. INVALID / REJECT

You can only promote an item to the subsequent status of its lifecycle.

`--producer`
Configuration file alias. For example, if the configuration file is `test.app.config`, the configuration file alias is `test.app`.

`--seqNo`
(Optional) The sequence number of the AIP in the belonging DSS. If not specified, all AIPs in the DSS are updated.

The following command updates the `2nd` SIP in DSS `SharePoint_2014929_DQGTMHZPFIRWY` to the `RECEIPT` status, and the configuration file is `config\test.app.config`:

```
InfoArchive.SharePoint.Connector.exe --updateTracking --dssID
SharePoint_2014929_DQGTMHZPFIRWY --status  RECEIPT --producer test.app --seqNo
2
```

2.2.5.4 Deleting tracking records

Use the delete command to delete tracking records of all SIPs in a DSS. You perform deleting tasks in the following scenarios:

- Listed items are archived and removed from the SharePoint site.
- The latest generated DSS is rejected or one of the AIPs in the DSS is invalidated.

Note: If one SIP in a DSS is invalidated, SharePoint connector extracts all items in the DSS in the subsequent run.

You can delete tracking records using the following arguments:

`--deleteTracking`
The type argument indicating the type of the task.

`--dssId`
The DSS which the AIP belongs to.

`--producer`
Configuration file alias. If the config file is `test.app.config`, the configuration file alias is `test.app`.

The following command deletes all tracking records for DSS `sharepoint_2014929_DQGTMHZPFIRWY`:

```
InfoArchive.SharePoint.Connector.exe --deleteTracking --dssID
sharepoint_2014929_DQGTMHZPFIRWY --producer test.app
```

2.3 SharePoint connector return codes

The following return codes help you locate and troubleshoot issues of SharePoint connector.

Return Code	Description
0	Success
1	SharePoint connector is started without command line arguments.
5	SharePoint connector does not extract any items. No SIP generated.
6	The `failOnAiuError` is set to false. Some items are skipped during the extraction.
210	Login error. Invalid credentials or privilege.
211	Encryption error. The encrypted password can not be decrypted, or an error occurred while generating the encryption keys.
220	Network error. Connection lost or server unreachable.
230	Invalid property value or invalid configuration file (XML syntax).
235	Query error.
241	Extraction denied. The status of one SIP of the previous extraction prevents SharePoint connector from performing a full extraction. SharePoint connector stops the extraction.
242	Parser error. An error occurred when AIUs are added to the PDI XML file.
243	File error. A referenced file is missing.
244	XSD file not found.
245	XSLT file not found.
250	Tracking error. An error occurred when creating, querying, or adding items to the tracking list.
251	Error occurred when tracking items are deleted.
252	Error occurred when tracking items are updated.
260	PDI XML file error
270	XSD validation fails.
272	XSLT validation fails.
280	ZIP error. Error occurred when ZIP files are created.
999	Unknown error occurred.

2.4 Configuration properties

2.4.1 SharePoint properties

The following outlines the SharePoint properties in the configuration file:

`authenticationMode`
The authentication mode when you access the SharePoint site. This property value is either `online` or `windows` (default value is `online`.

`domain`
The domain to which the SharePoint host belongs. If you set `windows` as the authentication mode, you must provide a domain value.

`protocol`
The protocol for establishing communication with the host. This property value is either <*http*> or <*https*> (default value is `https`).

`host`
The server on which the SharePoint site is hosted. If SharePoint is hosted on a port other than `80` (`http`) or `443` (`https`), you must specify the port number as a part of the host. For example, `myhost:8080`.

`configurationFileUrl`
When a local configuration file has a remote counterpart on the SharePoint site, `configurationFileUrl` must be specified. For example, if the URL to the remote file is: `https://myserver.sharepoint.com/sites/demo/Shared%20Documents/Demo.config`, the `configurationFileUrl` value is `/sites/demo/Shared%20Documents/Demo.config`.

`loginname`
The username you use to connect to the SharePoint site.

`encryptionKeysFolder`
The folder that holds the private and public keys, which are used to encrypt and decrypt the password.

The keys can only be generated once. If keys are regenerated after the encrypted password is saved to the configuration file, the encrypted password is not valid anymore. You must encrypt the password again.

`password`
The encrypted password that you use with the `loginname` to access a SharePoint site. Run SharePoint connector using the following command:

```
InfoArchive.SharePoint.Connector.exe -c config_file_relative_path
 --pwd
```

Enter the password according to the screen prompt. The value is populated by SharePoint Connector.

`requestTimeout`
If the connection cannot be established in a specified time (in ms), SharePoint Connector returns a time-out error (default value is `18,000 ms`).

`workingDir`
The directory where the generated SIPs are saved.

`site`
The site from which SharePoint connector extracts items. Specify a single site using the relative URL. For example, `/sites/demo`.

2.4.2 SharePoint extraction properties

`siteNavigation`
If `TRUE`, SharePoint connector extracts the root site and its sub-sites. If `FALSE`, SharePoint connector extracts the root site only. The default value is `true`.

`lists`
Specifies the list to extract items. one or more values accepted. If left empty, all lists are extracted.

`discardedLists`
The lists that are excluded from the extraction scope.

If you set a value for the `lists` property, this property is disabled, regardless of whether it contains a value or not.

`extractHiddenLists`
Whether SharePoint connector extracts hidden lists on the specified site. The default value is `false`.

`maxAiuPerSip`
The maximum number of AIUs allowed in a SIP package.

`maxCiSizePerSip`
The maximum size of content files allowed in a SIP. The size is the unzipped raw size in bytes. This value avoids files growing too big. If you do not set this property, the generated file size is determined by the value of enableZip64.

Note: On SharePoint 2010, the size of a SIP generated by SharePoint connector may exceed the limit.

`maxCiCountPerSip`
The maximum number of content files allowed in a SIP package.

If you do not set this property, the generated file size is determined by the value of enableZip64.

`enableZip64`
If `TRUE`, ZIP64 is applied. If `FALSE`, ZIP32 is applied.

- If `ZIP64` is enabled, each file size is less than 2^64 bytes, and the number of files is less than 2^64 (18,446,744,073,709,551,616).
- If `ZIP32` is enabled, each file size is less than 4 GB, and the number of files is less than 65,535.

The default value is `true`.

`discardedSites`
Lists of sub-sites to skip in the "site" hierarchy (not extracted).

Note: The parent ("site" configuration parameter) can be added to `discardedLists`. The "site" sub-sites will be extracted, not the parent site. The default value is `test1/subsite1,/test1/subsite10,/test/subsite99`.

`oneSipPerSubSite`
If set to `false`, one SIP for each sub-site equals one unique DssId / Site. With one DssId per site, you can either :

- Set different retention policies to each single site.
- Invalidate a specific sub-site without impacting the other sites extracted.

If set to `true`, one SIP for all the sites equals one unique DssId for all the sites Each will then produce from one to N `ZIP` files based on the rupture values (AIU count, CI count, Size).

The default value is `false`.

`maxParallelSiteExtractions`
From one to N parallel site extractions. For example, with one parent site and 11 sub-sites:

- If each have a similar size, set the value to `12`.
- If four are big and the others are small, set the value to at least `5` (up to `12`).

Caution
There is no maximum value, do not exceed the SharePoint throttling. Check with the SharePoint administrator.

The default value is `1`.

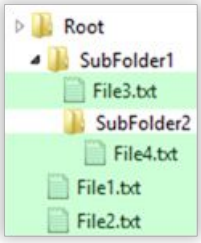
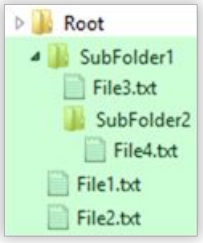
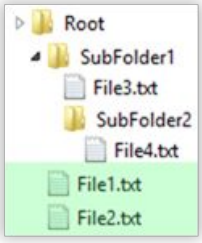
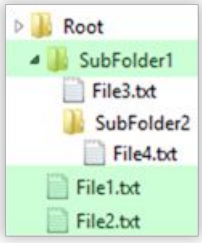
`QueryViewScope`
Specifies the scope for returning list items and list folders in a list view.

Values:

- `Recursive`: All list items in the list are extracted in the PDI file.
- `RecursiveAll`: All list items and all list folders in the list are extracted in the PDI file.
- `FilesOnly`: All list items in the current list folder are extracted in the PDI file.
- `RootItems`: All list items and all list folders in the current list are extracted in the PDI file.

  If the property is not set or empty, `RecursiveAll` is used by default.

The default value is `RecursiveAll`.

Recursive	RecursiveAll	FilesOnly	RootItems
			
AIU Count: 4	AIU Count: 6	AIU Count: 2	AIU Count: 3

2.4.3 Report generation properties

`generateReport`
A Boolean that controls the generation of a status report after site extraction. The default value is `true`.

`reportFolder`
Specifies the folder where the report is generated. If the specified folder does not exist, the system automatically creates one.

`reportFormat`
Specifies the report format. In this release, the value of this property must be `CSV`. The default value is `CSV`.

`reportColumnSeparator`
Specifies the column separator in the report. Valid values are `,` and `;`. The default value is `,`.

`getSiteStructure`
This generates the list of sites, their lists and number of items per list in the csv report. Set to true to optimize the parallel site extractions (`maxParallelSiteExtractions`). The default value is `false`.

2.4.4 PDI content properties

`sipSchema`
The SIP XML file schema.

The default value depends on which version of the product you are using:

- For OpenText Information Archive 4.0 : `urn:x-emc:ia:schema:sip:1.0`
- For OpenText Information Archive 3.2 and earlier versions: `urn:x-emc:eas:schema:sip:1.0`

`extractVersionsMetadata`
Whether SharePoint connector extracts versions metadata from the specified site. The default value is `true`.

`extractVersionsContent`
Whether SharePoint connector extracts each version of documents from the specified site. The default value is `true`.

`extractDocuments`
Whether SharePoint connector extracts documents (the content files), or just extracts the metadata. The default value is `true`.

`xsdInfoArchive`
The schema which is used to validate the generated PDI XML (before any transformation). The default value is `./resources/Sharepoint_1.0.xsd`.

`xslt`
The XSLT is used to transform the `eas_pdi.xml` files. OpenText does not provide any XSL file.

`xsdSchema`
The schema which is used to validate the generated PDI XML. (Only after XSLT processing)

`doTracking`
Whether SharePoint connector generates or updates the tracking list on the SharePoint site. The default value is `true`.

`eas_tracking_list`
The name of the tracking list. Use a descriptive name for the tracking list. Set this property value when `doTracking` is enabled.

`trackingChunkSize`
Chunk size of tracking items added or updated in the tracking list. Change the value to reduce the load of items being sent to the server. The default value is `500`.

`failOnAiuError`
If true, when some error occurs while SharePoint connector is extracting metadata or files, SharePoint connector stops extraction and returns an error code. If false, SharePoint connector skips the problematic item and continues to run.

Note: Setting this property to false only ignores errors related to metadata/file extraction. Other errors, such as failing to get a list of documents from a SharePoint server, cannot be ignored via this setting.

The default value is `true`.

`resolveUserPermissions`
Extracting user permissions can be costly and slow the extraction process down. This property allows you to disable the extraction of user permissions.

For SharePoint 2010, this will add user roles to user metadata:

```
<Roles>
  <Role Name="Limited Access" Type="Guest" />
</Roles>
```

For SharePoint 2013:

```
<Permissions>
EmptyMask,ViewListItems,AddListItems,EditListItems,
DeleteListItems,ApproveItems,OpenItems,
ViewVersions,DeleteVersions,CancelCheckout,
ManagePersonalViews,ManageLists,ViewFormPages,
AnonymousSearchAccessList,Open,ViewPages,
AddAndCustomizePages,ApplyThemeAndBorder,
ApplyStyleSheets,ViewUsageData,CreateSSCSite,
ManageSubwebs,CreateGroups,ManagePermissions,
BrowseDirectories,BrowseUserInfo,
AddDelPrivateWebParts,UpdatePersonalWebParts,
ManageWeb,AnonymousSearchAccessWebLists,
UseClientIntegration,UseRemoteAPIs,ManageAlerts,
CreateAlerts,EditMyUserInfo,EnumeratePermissions,
</Permissions>
```

The default value is `false`. Set to `true` to extract user permissions.

For the `resolveLookupFieldAssociatedList` property, add the Lookup Field's associated list.

Caution

Using this property slows down the extraction.

2.4.5 PDI content

All the item attributes are extracted in the PDI, but only a subset is available in the holding configuration wizard:

- Content type ID and name (task, document, etc.)
- Title
- Author and modifier (with ID, name, title, and e-mail)
- Document and versions, if any (document name, creation date, mimetype, size, and path)
- Site and list
- Attachments (document name, creation date, mimetype, size, and path)

To expose additional attributes, open the generated `eas_pdi.xml` file. There are two sections:

- `<base>`: Contains the exposed attributes
- `<other>`: Contains the other attributes (not exposed in the schema). These attributes might be internal to SharePoint and not relevant for use, custom attributes or specific to each item type. For example, task items have specific attributes, such as Status, Priority, PercentComplete, AssignedTo, StartDate, DueDate, etc.

Identify which attributes you need. For example, if you need the task object's Priority, Status and PercentComplete attributes and modify their values:

```
<Priority displayName="Priority">(2) Normal</Priority>
<Status displayName="Task Status">In Progress</Status>
<PercentComplete displayName="% Complete">0,33</PercentComplete>
```

Modify the `.\resources\SharePointToIA.xsl` and `.\resources\pdi-schema.xsd` files to move the attributes to the `<base>` element of the PDI file (xsl), and the XSD to reflect the new schema. Also, modify the percent value to be `33%` instead of `0,33` and the priority to be `Normal` instead of `(2) Normal`.

The modified XSD and XSL files are included in the `resources\sample` folder. The result after the modification looks like:

```
<Priority displayName="Priority">Normal</Priority>
<Status displayName="Task Status">In Progress</Status>
<PercentComplete>33%</PercentComplete>
```

2.4.6 OpenText Information Archive properties

`sipXslt`
: Allows you to transform the default SIP XML from a XSLT

`sipXsdSchema`
: Validates the transformed SIP XML: `ia_sip.xsd` for 4.x or `eas_sip.xsd` for 3.x. The `XSD` files are in the `./resources` folder. The default value is `./resources/ia_sip.xsd`.

`holding`
: The holding name in the SIP descriptor. If you plan to ingest the SIP into the SharePoint holding in SharePoint connector distribution package, do not change the default value. The default value is `sharepoint`.

`pdi_schema`
: The Schema to validate the generated PDI file, with the version number appended to the end of the Schema. The default value is `urn:x-emc:eas:schema:sharepoint:1.0`.

`pdi_schema_version`
: The Schema version.

`entity`
: The entity element in the SIP descriptor. The default value is `EAS`.

`priority`
: The priority value of the SIP descriptor. The default value is `0`.

`application`
: The application element in the SIP descriptor. Usually the same as the holding name.

`retention_class`
: The `retention_class` applied to the archived items.

`archiveId`
: During SharePoint connector's run, archive ID, date, and a random string are concatenated to form a new string, which is set as the `<id>` element value in the

SIP descriptor and also displayed in the DSS column of the tracking list. The final string cannot exceed 32 characters, so don't set this property's value too long.

2.5 Troubleshooting

SharePoint connector logs are saved in `EAS.Connector.log.<date>.log` file in the same directory as SharePoint Connector executable. The following table lists errors that you may come across.

Error	Solution
`ERROR [EAS.Connector. Sharepoint2013.UtilityClasses. ConnectorException] - [CONFIGURATION]An XML comment cannot contain '--', and '-' cannot be the last character. Line 23, position 83.`	Go to the specified location to remove `--` or `-` in XML comments.
`ERROR [EAS.Connector. Sharepoint2013.Extractor] - Login failed to https://<[url to your sharepoint server]>:System. FormatException: The input is not a valid Base-64 string as it contains a non-base 64 character, more than two padding characters, or an illegal character among the padding characters.`	Regenerate the encrypted password. If the error continues, regenerate the keys and the encrypted password following the steps in Generating SIPs from a SharePoint site.
`ERROR [EAS.Connector. Sharepoint2013.UtilityClasses. XmlTools] - The [C:\...\eas_pdi.xml] is not valid according to [Sharepoint_1.0.xsd]`	OpenText provides a default `Sharepoint_1.0.xsd` in `Sharepoint Holding\template\content`. You may need to customize the default XSD file according to your specific requirements.
`[LOGIN]Login failed to https://<[url to your sharepoint server]>:Microsoft.SharePoint. Client.IdcrlException: The identity has not been authenticated.`	The password in the configuration file is invalid. Make sure the password is correct and regenerate the encrypted password.
`Could not find the xsd Schema : [XSD file path]`	Check and update the `xsdSchema` property in the configuration file.
`[ENCRYPTION]Exception while decrypting the value: System.IO. DirectoryNotFoundException: Could not find a part of the path [encryptionKeysFolder\private.xml] [ENCRYPTION]Exception while encrypting the value: System.IO. DirectoryNotFoundException: Could not find a part of the path [encryptionKeysFolder\public.xml]`	There is an error with the key files used for the password encryption or decryption. Regenerate the keys files and the encrypted password.

Error	Solution

`[25O:TRACKING] Could not add the tracking items to the tracking list: Microsoft.SharePoint.Client. ServerException: The request uses too many resources.`	Tune the `trackingChunkSize` configuration property to reduce the number of items sent in the request.
	
	Alternatively, you can change the SharePoint server `maxObjectPaths` value to accept more items in each single request by using the following Powershell script:
	
	`$webApp = Get-SPWebApplication "site"`
	`$webApp.ClientCallableSettings.MaxObje ctPaths = 2000`
	`$webApp.Update()`