opentext™

OpenText™ Structured Data Manager

Runtime Guide

Version : 26.1

PDF Generated on : February 18, 2026

Table of Contents

1. Runtime Guide

About this document

OpenText™ Structured Data Manager provides powerful tools to build an archive solution that copies or moves data out of your production database and into less expensive storage.

This guide provides information about:

- configuring the Web Console

- deploying business flows

- running business flows and jobs

- monitoring business flows and jobs

- accessing archived data

Intended audience

This guide is intended for:

- users configuring the Web Console

- users running business flows and jobs

Prerequisites

Prerequisites for installing this product include:

- Knowledge of the operating system

- Database knowledge

- Application knowledge

Related documentation

Document Name	Description
OpenText™ Structured Data Manager API Reference Guide	Provides reference to the available programming interfaces.
OpenText™ Structured Data Manager Certification Matrix	Provides information about supported Operating Systems, databases, browsers, software integrations and other technology stacks.
OpenText™ Structured Data Manager Concepts Guide	Explains the major concepts of database archiving in general and Structured Data Manager in particular.
OpenText™ Structured Data Manager Developer's Guide	Explains how to use the Designer component to design, build, test, and deploy your archiving projects.
OpenText™ Structured Data Manager Installation Guide	Explains how to install the product.
OpenText™ Structured Data Manager Release Notes	Lists any items of importance that were not captured in the regular documentation.
OpenText™ Structured Data Manager Troubleshooting Guide	Explains how to diagnose and resolve errors, and provides a list of common errors and solutions.
OpenText™ Structured Data Manager Tutorial	Provides step-by-step instructions to build a sample archiving module, deploy, run, and troubleshoot errors in it.
OpenText™ Structured Data Manager Upgrade Guide	Explains how to upgrade the product and archive schema generated by the earlier versions of the product.
OpenText™ Structured Data Manager Discovery Guide	Explains the purpose, how to install and use Discovery.

1.1. Introduction

Managing structured data is the act of moving or retiring data from an active database to an archive data store. An active database is typically an online transaction processing (OLTP) database and is also sometimes referred to as a production database. An archive data store is another database (sometimes called an archive database) or a set of files (CSV or XML). Moving means copying the specified parts of the active database to the archive data store and deleting it from the active database.

Optimization is desirable for many reasons, but two of the most common ones are:

- **To retain data for the long term while keeping it accessible for the purposes of corporate governance, eDiscovery, and/or legal hold.** Database to file movement enables you to move or copy large amounts of data from a database to structured files.

- XML files provide an open, standards-based format that can be opened long after the original application has been retired. As such, it provides an excellent method of storing data for long term retention.
- Comma separated values (CSV) and JavaScript Object Notation (JSON) files provide a simple format that many applications can import.

- **To reduce the footprint of your active database.** After databases and their associated applications have been in use for some time, they often become very large, which slows performance and increases the need for additional, expensive hardware. Archiving older, infrequently accessed data from the active database enables you to move eligible data to less expensive storage while also reorganizing the active database. This reduction in footprint tends to improve the performance of the active database.

Before you begin

Before you begin performing the tasks in this guide, you should:

1. Go through *OpenText™ Structured Data Manager Concepts Guide* to become familiar with the concepts, software, and how you plan to use it.

2. Install the Structured Data Manager software according to the instructions in the *OpenText™ Structured Data Manager Installation Guide*.

3. Go through *OpenText™ Structured Data Manager Tutorial*. The tutorial enables you to get hands on with the product quickly and exposes you to many of the most commonly used features.

4. Use the *OpenText™ Structured Data Manager Developer's Guide* to design and develop or customize your archive solution.

Web Console overview

Most of the work of archiving data is performed in the Web Console. The Web Console is a powerful browser-based interface used to:

- Configure the repository

- Create environments

- Create and modify users

- Deploy business flows

- Launch business flows to move data

- Monitor running business flows and their history

The Web Console enables you to remotely manage multiple environments for the movement of data.

1.2. Tutorials

This part provides a set of tutorials that build on the basic tutorial described in *OpenText™ Structured Data Manager Tutorial*. It includes:

- Configure the Web Console
- Deploy and run business flows
- Upload to Vertica
- Perform multiple data movements
- View eligibility analytics

1.2.1. Configure the Web Console

Before you can deploy, run, and monitor your business flows to move data, you must configure the Web Console. The Web Console is the primary runtime interface to Structured Data Manager.

This tutorial walks you through the process of configuring the Web Console, including:

> **Note**
>
> Before you begin this tutorial, make sure that you have installed Structured Data Manager according to the *OpenText™ Structured Data Manager Installation Guide*.

- Start the Web Console
- Configure the third-party libraries
- Install the repository
- Create an environment
- Configure settings
- Manage the home directory
- Create and edit Web Console users and groups (optional)

1.2.1.1. Start the Web Console

The Web Console must be running before you can connect to it in your browser. After you have started the Web Console, you can perform operations such as installing the repository, creating environments, and managing users.

> **Note**
>
> When you install Structured Data Manager, the Installer gives you options to start the Web Console and launch its interface in your browser when the installation completes. If you chose those options and the Web Console is already running, you need not perform the steps in this section.

To start the Web Console

1. You can start the Web Console in either of the ways described in Start the Web Console.

2. Launch the Web Console.

1.2.1.2. Configure the third-party libraries

To establish a database connection with databases like Oracle, DB2, or Sybase, their respective libraries must be installed and configured explicitly before you begin repository installation and environment creation.

> **Important**
>
> Starting with version 25.3.0, previously bundled third-party libraries are no longer included in the SDM installer. The excluded libraries are:
>
> - db2jcc4-*.jar
> - jconn4-*.jar
> - ojdbc11-*.jar
> - xdb6-*.jar
> - xmlparserv2-*.jar
>
> Refer Setup the third-party libraries to use these additional jars.

1.2.1.3. Install the repository

Structured Data Manager stores metadata in a repository to facilitate the deployment and running of business flows, and administering the Web Console.

You can install the repository as described in Install the repository.

Login to Web Console

You can login to Web Console as described in Login to Web Console.

1.2.1.4. Create an environment

Before you can deploy your business flows, you must create a deployment environment for them with the appropriate characteristics, such as the source database credentials and available data movement methods.

> **Important**
>
> Prerequisite to create a new DB2 environment on zOS
>
> When connecting to a mainframe DB2 database (such as DB2 for z/OS, DB2 for iSeries, and DB2 for VM/VSE), you must have the licensed JAR files.These jar files are not free, you must purchase the DB2 Connect product which includes the following licensed jar file in the activation package:
>
> - db2jcc_license_cisuz.jar
>
> - db2jcc4.jar
>
> **Once the above .jar files are available, perform the below steps to ensure that your DB2 connections on zOS work correctly:**
>
> 1. Place the `db2jcc_license_cisuz.jar` file under the following folders:
>
>   - `<install_dir>/obt/lib` folder.
>
>   - `<install_dir>/obt/webconsole/apache-tomcat/webapps/WebConsole/WEB-INF/lib>` folder.
>
> 2. Restart the Web Console.
>
> **Simplified approach**
>
> SDM provides a utility to copy the additional `.jar` files as `copy_jar`. You can run the `copy_jar.batch` file on Windows or `copy_jar.sh` file on Linux system. While running this script, you need to provide the path of the additional file to be copied.
>
> This utility manages a similar requirement for JDBC connection in Designer. For more information, refer to **Create a connection** section in *Developers* guide.

> **Note**
>
> Prerequisite to create a new Oracle, DB2 or Sybase environment
>
> Before creating new environment, make sure that all their respective libraries are properly configured. See, Setup the third-party libraries for more information.

To create an environment

1. Launch the Web Console in your browser and login as the administrative user you created when you installed the repository, Install the repository.

2. Click **Environment Management** from the Web Console home page.

3. If you have not previously created an environment, the Web Console automatically prompts you to create one when you open it. Otherwise, from the Web Console home page, go to **Environment Management** and then click **New**. The New Environment wizard is displayed.

4. Specify a name for the new environment, for example, `Oracle_OLTP` or `SQLServer_OLTP`.

5. For environment configuration, specify whether you want an **Intrusive/Standard** or **Non-Intrusive** (default selection is **Non-Intrusive**). For more information, see Manage environments.

   > **Note**
   >
   > - For D2D environment, select Intrusive/Standard environment configuration.
   > - PostgreSQL and Sybase are supported only for Non-Intrusive environment.

6. Choose a user group from the **Authorized Group** list to which you want to grant access to the environment. Members of this group will have privileges on the environment.

> **Tip**
>
> If you want to give other groups privileges on the environment, change the group membership, or modify the privileges of the group, see Create or edit a user group.

7. Choose the user group to which you want to grant access to the user group. See Create and edit Web Console users and groups (optional) for more information.

8. Select the appropriate **Source Database Type** from the list of values for your active database.
   The available fields change depending on the selected database.

> **Note**
>
> - For Azure SQL environment, select SQL Server source database type.
> - Azure SQL D2D SI does not support the following capabilities:
>   - Environment creation via JDBC URL as source database type
>   - Reload
>   - Create Archive Access
>   - D2F 2-tier and 3-tier
>   - Deployment of D2D Business flow having Views
>   - D2D masking
>   - Deletion of Azure SQL D2D SI environment

9. Enter the information relevant to your database type, and click **Next** after entering the information.

> **Tip**
>
> Check **Remember** if you want to store the credentials and not have to re-enter them during deployment of cartridges and business flows. If you chose **JDBC URL** for database type, you should not choose **Remember**.

DB2

Field	Description
Administrative User	**For intrusive/standard environments.** Enter the name of a user with administrative privileges. This user needs to be able to perform database operations such as CREATE and DELETE in addition to being able to read and copy data.
User	**For non-intrusive environments.** Enter the name of a user that has at least CONNECT and SELECT ON TABLE privileges. If you plan to perform DELETE operations after archiving, this user must also have that privilege on the source database.
Password	Enter the password for the specified user.
Host	Enter the host name of the machine where your DB2 database is installed.
Port	Enter the port number of your DB2 database.
Database	Enter the name of the DB2 database you want to connect to.

JDBC URL

Selecting **JDBC URL** indicates that you want to connect to a database using a JDBC URL string. A common use of the JDBC URL is for generic JDBC non-intrusive connections. One required usage is to connect to an Oracle database with Real Application Clusters (RAC).

Field	Description
User	Enter the administrator user name.
Password	Enter the administrator password.

URL:	Enter the connection URL for the database. JDBC URLs are often used to connect to an Oracle database with RAC.
	
	For example:
	
	`jdbc:oracle:thin:@(DESCRIPTION=(SDU=32768)`
	`(enable=broken)(LOAD_BALANCE=yes)(ADDRESS=`
	`(PROTOCOL=TCP)(HOST=gvu2707.london.com)`
	`(PORT=1525))`
	`(ADDRESS=(PROTOCOL=TCP)`
	`(HOST=gvu2923.london.com)(PORT=1525))`
	`(CONNECT_DATA=(SERVICE_NAME=SAPDEMI)))`

Oracle

Field	Description
Administrative User	**For intrusive/standard environments**. Enter the name of a user with administrative privileges. This user needs to be able to perform database operations such as CREATE and DELETE, in addition to being able to read and copy data.
User	**For non-intrusive environments**. Enter the name of a user that has at least SELECT ANY TABLE and CREATE SESSION privileges. If you plan to perform DELETE operations after archiving, this user must also have that privilege on the source database.
Password	Enter the password for the specified user.
Host	Enter the name of the machine where your Oracle database is installed.
Port	Enter the port number of your Oracle database, for example, 1521.
Service Name	Enter the name of the Oracle instance you want to connect to.

PostgreSQL

Field	Description
User	Enter the administrator user name.
Password	Enter the password for the specified user.
Host	Enter the host name of the machine where your PostgreSQL database is installed.
Port	Enter the port number of your PostgreSQL database.
Database Name	Enter the PostgreSQL database name.

SQL Server

> **Tip**
> For SQL Server, ensure that the SQL Server Browser service is running and you are using a fixed port rather than a dynamic one. See Manage the home directory.

Field	Description
Authentication	**For intrusive/standard environments.** Select SQL Server Authentication or Windows Authentication from the list of values. ○ **SQL Server Authentication** indicates that you will use a SQL Server login and password to connect to the database. ○ **Windows Authentication** indicates that you will use the Windows operating system login for the machine to connect to the database. If you choose Windows Authentication, you only need to fill in the **Host**, **Port**, and **Instance Name**. Not available for Azure SQL Database connections.
Administrative User	**For intrusive/standard environments with SQL Server Authentication.** Enter the name of a user with administrative privileges on the database. This user needs to be able to perform database operations such as CREATE and DELETE in addition to being able to read and copy data. Not required for the Windows Authentication option.
User	**For non-intrusive environments.** Enter the name of a user that has at least SELECT ON privileges. If you plan to perform DELETE operations after archiving, this user must also have that privilege on the source database. Not required for the Windows Authentication option.
Password	Enter the password for the specified user. Not required for the Windows Authentication option.
Host	Enter the host name of the machine where your SQL Server database is installed. For Azure SQL , enter the server details where Azure SQL database is running. For example, azureServerName.database.windows.net
Port	Enter the port number of your SQL Server database.
Instance Name	Enter the name of the SQL Server instance that contains the database you plan to use. For Azure SQL, enter the same value as provided in the **Database Name** field.
Database Name	Enter the name of the SQL Server/ Azure SQL database to be used as source for archiving.

Sybase

Field	Description
User	Enter the administrator user name.
Password	Enter the password for the specified user.
Host	Enter the host name of the machine where your Sybase database is installed.
Port	Enter the port number of your Sybase database.
Database	Enter the Sybase database name.

10. For a **Non-Intrusive environment**, click **Finish** on the **Summary** page.

Add database to database capability

11. Optionally select the **Include Database to Database Capability** check box. (Database to database capability is only available for Oracle, SQL Server, and DB2.) By default, every environment includes database to file movement. If you also want to move data from database to database in this environment, you must choose this option.

If you do not choose the database to database option, skip to step 15.

12. If you selected **Include Database to Database Capability** in the previous step, choose the radio button that corresponds to the database location where you want to store your archived data:

> **Note**
>
> - For DB2, the tablespace size for your history database must accommodate additional columns added to the history tables. The table space size should be greater than or equal to the tables space on the source database for all managed tables.
> - For DB2, complex (structured) UDTs (user defined types) are not supported.
> - For Azure SQL, only **Single Instance** is supported.

- **Archive Data within the same database (Single Instance)** means that you want to move data to a different location in the source database. In this case, the archive tables exist within the same database as the active tables.

- **Archive Data in another database (Distributed Instance)** means that you want to move data to a different database. In this case, the archive tables exist in a separate (target) database. If you choose this option, you must also provide credentials for the target database.

> **Note**
>
> For distributed instances on DB2, you must enable federation on both source and target databases. You also need to create a user mapping between the interface and relocation user so that each can access the federation (dblink). Furthermore, the archive access user and the relocation user must be mapped for the federation. See your DB2 documentation for more information about setting up federation.

13. If you chose a distributed instance (a separate target database), enter or select values for the target database. Otherwise, go to step 15. The values for the target database are similar to those you entered in step 9 for the source database.

- If you used JDBC URL or SQL Server Windows Authentication to connect to the source database, you must use the same connection method to connect to the target database.

- If you are using an Oracle database for your source database, you are also prompted to enter the following:

  - The TNS name of the source and target Oracle database service.

  - The names for the database links to be used for the connections between the source and target databases.

    > **Note**
    >
    > If `GLOBAL_NAMES` is set to true, then the database link names must be the same as the source database and target database names.

14. Click **Next**.

    The Interface User page displays.

    Create schema

15. Enter or select the following information for the interface user, and click **Next** after entering the information:

> **Note**
>
> For DB2, you should pre-create an interface user to select here. The user cannot be the same as your administrative user. The user must have CONNECT privileges on the source database. See your DB2 documentation for information about creating users and setting user privileges.

Field	Description
User	The default value is obt_if. For Oracle and SQL Server, enter the name you want to use for the interface user. For DB2, choose an existing operating system user from the list of values. The list of values is populated with users that have connect privileges on the DB2 database.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Schema	For DB2, enter the name of the schema you want to use for the interface user. The default value is obt_if.
Data Tablespace	For Oracle and DB2, select the data tablespace you want to use.
Temp Tablespace	For Oracle, select the temporary tablespace you want to use.
Primary Data Size	For SQL Server and Sybase, accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	For SQL Server and Sybase, accept the default value or enter a maximum size in MB for the log file.

16. If you chose a distributed instance, enter the following values for the **Relocation user**, and click **Next** after entering the values:

> **Note**
>
> For distributed instances with DB2, your relocation user must have CONNECT privileges on the target database. See your DB2 documentation for information about setting user privileges.

Field	Description
User	The default value is obt_reloc. For Oracle, SQL Server, and Azure SQL, enter the name you want to use for the interface user. For DB2, choose an existing operating system user from the list of values. The list of values is populated with users that have connect privileges on the DB2 database.
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Schema	For DB2, enter the name of the schema you want to use for the relocation user. The default value is obt_reloc.
Data Tablespace	For Oracle and DB2, select the data tablespace you want to use.
Temp Tablespace	For Oracle, select the temporary tablespace you want to use.
Primary Data Size	For SQL Server, accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	For SQL Server, accept the default value or enter a maximum size in MB for the log file.

17. If you chose database to database movement, enter the following information for the user that will build archive access:

> **Note**
>
> For DB2, your archive access user must have CONNECT privileges on the source database. See your DB2 documentation for information about setting user privileges.

Field	Description
Location	Select Source or Target.
User	The default value is obt_build_aa. For Oracle, SQL Server, and Azure SQL, enter the name you want to use for the archive access user.

For DB2, choose an existing operating system user from the list of values. The list of values is populated with users that have connect privileges on the DB2 database. |
Password	Enter the password for the user you created.
Confirm Password	Enter the password again.
Schema	For DB2, enter the name of the schema you want to use for the archive access user. The default value is obt_build_aa.
Data Tablespace	For Oracle, select the data tablespace you want to use.
History Data Source for AA on OLTP	(DB2 only) Enter the name of the history data source used for archive access. The default value is OBT_OLTP_TO_HIST.
Temp Tablespace	For Oracle, select the temporary tablespace you want to use.
Primary Data Size	For SQL Server, accept the default value or enter a maximum size in MB for the data file.
Transaction Log Size	For SQL Server, accept the default value or enter a maximum size in MB for the log file.

18. Click **Next**.

The Summary page displays for your review.

19. Click **Finish** to create the environment.

It may take several moments to create the environment. When the environment is ready, the success message should appear at the top of the page.

For the purposes of this tutorial, you can now optionally create a Web Console user Create and edit Web Console users and groups (optional), or skip directly to Deploy and run business flows.

1.2.1.4.1. SDM database user privileges

During the environment creation, the following is the list of SDM database user privileges for the D2D (Database to Database) and D2F (Database to File) archiving functions:

D2D (Intrusive)

	OBT_BUILD_AA	OBT_IF	OBT_REP	OBT_AA
ADMINISTER DATABASE TRIGGER				✓
ALTER ANY	✓			
ALTER ANY PROCEDURE	✓	✓		
ALTER ANY INDEX		✓		
ALTER ANY TABLE	✓	✓		
ALTER ANY TRIGGER	✓	✓		
ALTER SESSION	✓	✓	✓	✓
ANALYZE ANY		✓		
CREATE ANY INDEX	✓	✓	✓	
CREATE ANY PROCEDURE	✓	✓		✓
CREATE ANY SYNONYM	✓	✓	✓	✓
CREATE ANY TABLE		✓		
CREATE ANY TRIGGER		✓		
CREATE ANY VIEW	✓			✓
CREATE DATABASE LINK	✓	✓		✓
CREATE PROCEDURE	✓	✓	✓	
CREATE PUBLIC SYNONYM		✓		
CREATE SEQUENCE	✓	✓	✓	
CREATE SYNONYM	✓			
CREATE SESSION	✓	✓	✓	✓
CREATE TABLE	✓	✓	✓	✓
CREATE VIEW	✓	✓	✓	
DELETE ANY TABLE	✓	✓		✓
DROP ANY INDEX	✓	✓		
DROP ANY PROCEDURE	✓			✓
DROP ANY SYNONYM	✓	✓		

	OBT_BUILD_AA	OBT_IF	OBT_REP	OBT_AA
DROP ANY TABLE		✓		
DROP ANY VIEW	✓			
DROP PUBLIC SYNONYM		✓		
EXECUTE ANY PROCEDURE	✓	✓		✓
EXECUTE ANY TYPE	✓	✓		✓
LOCK ANY TABLE	✓			✓
INSERT ANY TABLE	✓	✓		✓
SELECT ANY DICTIONARY	✓	✓	✓	✓
SELECT ANY SEQUENCE	✓	✓		✓
SELECT ANY TABLE	✓	✓		✓
UPDATE ANY TABLE	✓			✓
UNLIMITED TABLESPACE		✓		

D2F (non-intrusive)

	OBT_BUILD_AA	OBT_IF	OBT_REP	OBT_AA
ADMINISTER DATABASE TRIGGER				
ALTER ANY				
ALTER ANY PROCEDURE				
ALTER ANY INDEX				
ALTER ANY TABLE				
ALTER ANY TRIGGER				
ALTER SESSION			✓	
ANALYZE ANY				
CREATE ANY INDEX				
CREATE ANY PROCEDURE				
CREATE ANY SYNONYM				
CREATE ANY TABLE				
CREATE ANY TRIGGER				
CREATE ANY VIEW				
CREATE DATABASE LINK				
CREATE PROCEDURE			✓	
CREATE PUBLIC SYNONYM				
CREATE SEQUENCE			✓	
CREATE SYNONYM			✓	
CREATE SESSION			✓	
CREATE TABLE			✓	
CREATE VIEW			✓	
DELETE ANY TABLE				
DROP ANY INDEX			✓	
DROP ANY PROCEDURE				
DROP ANY SYNONYM				

	OBT_BUILD_AA	**OBT_IF**	**OBT_REP**	**OBT_AA**
DROP ANY TABLE				
DROP ANY VIEW				
DROP PUBLIC SYNONYM				
EXECUTE ANY PROCEDURE				
EXECUTE ANY TYPE				
LOCK ANY TABLE				
INSERT ANY TABLE				
SELECT ANY DICTIONARY			✓	
SELECT ANY SEQUENCE				
SELECT ANY TABLE				
UPDATE ANY TABLE				
UNLIMITED TABLESPACE				

D2F (Intrusive)

	OBT_BUILD_AA	**OBT_IF**	**OBT_REP**	**OBT_AA**

	OBT_BUILD_AA	OBT_IF	OBT_REP	OBT_AA
ADMINISTER DATABASE TRIGGER				
ALTER ANY				
ALTER ANY PROCEDURE		✓		
ALTER ANY INDEX		✓		
ALTER ANY TABLE		✓		
ALTER ANY TRIGGER		✓		
ALTER SESSION		✓	✓	
ANALYZE ANY		✓		
CREATE ANY INDEX		✓	✓	
CREATE ANY PROCEDURE		✓		
CREATE ANY SYNONYM		✓		
CREATE ANY TABLE		✓		
CREATE ANY TRIGGER		✓		
CREATE ANY VIEW				
CREATE DATABASE LINK		✓		
CREATE PROCEDURE		✓	✓	
CREATE PUBLIC SYNONYM		✓		
CREATE SEQUENCE		✓	✓	
CREATE SYNONYM		✓	✓	
CREATE SESSION		✓	✓	
CREATE TABLE		✓	✓	
CREATE VIEW		✓	✓	
DELETE ANY TABLE		✓		
DROP ANY INDEX		✓	✓	
DROP ANY PROCEDURE				
DROP ANY SYNONYM		✓		

	OBT_BUILD_AA	OBT_IF	OBT_REP	OBT_AA
DROP ANY TABLE		✓		
DROP ANY VIEW				
DROP PUBLIC SYNONYM		✓		
EXECUTE ANY PROCEDURE		✓		
EXECUTE ANY TYPE		✓		
LOCK ANY TABLE				
INSERT ANY TABLE		✓		
SELECT ANY DICTIONARY		✓	✓	
SELECT ANY SEQUENCE		✓		
SELECT ANY TABLE		✓		
UPDATE ANY TABLE				
UNLIMITED TABLESPACE		✓		

Note

Only users who have the write access to the source can insert, update (In-place masking), or delete the data.

In case of D2F, a user must have the write access to the destination location. For example, Local File System, and Content Manager.

1.2.1.5. Configure settings

The Web Console includes a number of other settings that you can configure from the **Settings** menu.

For more information on these settings, see the Web Console Help system.

Set up email access

You can set up the Web Console to send out email alerts for various events. Before you can send email using the Web Console, you need to configure your Server Email settings.

To configure the settings

1. Navigate to **Settings** in the Web Console.
2. Click **Email Server**.
3. The Email Settings page opens.
   To require authentication (password), select **Authentication Required**.
4. Enter the email settings:
   - Web Server URL: The URL for the server where the Web Console is running. This is the URL that other machines need to connect to the Web Console, for example, `http://<host>:8080/WebConsole`.
   - Email Host: The host name of the mail server you want the Web Console to use for sending messages, for example, smtp.gmail.com.
   - Email Port: The mail server port number, for example, 465.
   - Email User Name: The email account name, for example, tom.baker@hiscompany.com.
   - Email Password: The password to the email account. This field is visible only if you selected **Authentication Required**.
   - Email From: The email address that will appear in the From part of the email. It might be the same as User Name, but you could also enter a different name here.
   - Additional Settings: Enter any additional email settings as `key=value` pairs, separated by semicolons (;).
     For example:
     `mail.smtp.auth=true;mail.smtp.starttls.enable=true;mail.smtp.host=smtp.example.com;mail.smtp.p`
     `mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory;mail.smtp.socketFactory.port=465;`
5. Click **Apply**.
6. To test whether your server settings are correct and working, navigate to the following URL:
   http://<host>:8080/WebConsole/main/mailTest

> **Note**
>
> You must be an administrator for this page to work for you, and your email information must be set up at
> `http://<host>:8080/WebConsole/settings/info`.

Manage indexing server settings

On the Indexing Server Settings page, you can specify the indexing server and options to use when running indexing cartridges.

To manage indexing server settings

1. From the Web Console home page navigate to **Settings > Indexing Server**. The Index Server page displays.

2. Select the **Indexing Server Type**:

   1. *Knowledge Discovery (IDOL)*

      1. Select the protocol to employ when communicating with the Knowledge Discovery (IDOL) server, **HTTP** or **HTTPS**. See the *Knowledge Discovery Administration Guide*, "Set Up an SSL Connection," for more information about HTTPS and Knowledge Discovery (IDOL).

      2. Enter the **Server Host** where Knowledge Discovery (IDOL) is located.

      3. Enter the **Server Action Port** and **Server Content Port** for Knowledge Discovery (IDOL), for example 9000 and 9001.

      4. If you wish to enable TEA encryption, check **Enabled** and enter your TEA encryption key in the provided field.

         For more information about TEA, see the OpenText™ Structured Data Manager *Developer's Guide.*

5. Enter the **Default Knowledge Discovery (IDOL) Database Name**. You can enter a different database name in the cartridge editor for each cartridge to override this default.

6. Choose a database security option:

   - **Allow users with data access privileges to query any Knowledge Discovery (IDOL) database** means that any user who has the data access privilege granted to them can also query any Knowledge Discovery (IDOL) database. See Manage groups for more information on privileges.

   - **Grant users Knowledge Discovery (IDOL) database query access** means that you must explicitly grant users query access to Knowledge Discovery (IDOL) databases. If you choose this option, when you select **Apply**, a new link appears called **Set security by database**. Following the link takes you to a page where you can manage query access by Knowledge Discovery (IDOL) database and user.

7. Optionally, modify the Groovy script that specifies the entry ID.

2. *Apache Solr*

   1. Enter the **URL** where the server is located.

   2. Enter the **Entry Reference ID**.

3. **Elastic search**

   1. The **URL** field is auto filled. You must provide the appropriate URL where the Apache Solr instance is installed and the service is started.

   2. The **Entry reference ID** field will have the appropriate URL by default. Using this field you can provide any IP address where the Structured Data Manager is installed to perform free text search in that machine.

   3. Click **Apply** to save settings.

   To set up the Elastic search:

   - Refer to https://www.elastic.co/guide/en/elasticsearch/reference/6.1/zip-targz.html

   - You can test that your Elasticsearch node is running by sending an HTTP request to port 9200 on localhost using the following URL http://localhost:9200/

   - In Web Console the URL requires the index to be specified, for example: http://localhost:9200/sdm-index.

   > **Note**
   >
   > The index specified must be unique for each repository.

3. Click **Apply**.

Configure the trusted store for certificates

For locations that require secured communications, you need to store the certificates from trusted Certificate Authorities (CAs).

To store the certificates

1. Navigate to **Settings** in the Web Console.

2. Click **Certificate Issuers**.

3. Enter the certificate settings.

   - **File Path:** The file that contains all of the certificates from CAs that are trusted. Any certificate not issued by one of these issuers will not be accepted by the HTTPS processing. Enter the full path name to this file.

   - **Password:** The password for the trust store.

   - **Type**: The type of trust store (JKS, PKCS12, and so on).

4. Click **Apply**.

Masking Server settings

Structured Data Manager now supports the Format Preserving Encryption (FPE) and Secure Stateless Tokenization (SST) provided by Data Privacy and Protection Appliance (DPP Appliance (SDA)). The Masking Server Settings page includes a default public instance so you can try FPE/SST. However, OpenText recommends that you purchase the on-premises DPP Appliance (SDA) product.

Prerequisites

The Data Privacy and Protection SimpleAPI (SimpleAPI) must be installed on your server.

Connecting to SimpleAPI from SDM

To connect any versions of SimpleAPI from SDM, follow the below steps:

1. Stop the Web Console and Designer.

   > **Note**
   >
   > For a fresh installation, you must start the Web Console once and then stop the Web Console.

2. Install any version of SimpleAPI on SDM machine.

   For example: 6.22.0 or 6.20.0 or 6.0.0

   > **Tip**
   >
   > Step 3 is optional for SimpleAPI version 6.22.0

3. Run `replace_jar.bat/sh`, to replace the `vibesimplejava.jar` file in the SDM folders with the `vibesimplejava.jar` file in `<SimpleAPI_Install>\lib` folder

   where <SimpleAPI_Install> is the location where SimpleAPI is installed.

   For example:

   Go to `<install_dir>\obt\bin`

   Run the command

   `replace_jar.bat/sh vibesimplejava*.jar <SimpleAPI_Install>\lib\vibesimplejava.jar -all`

4. Start the Web Console and Designer.

DPP Appliance (SDA) Settings

For authentication, DPP Appliance (SDA) uses shared secret, username/password and certificates as different authentication mechanisms.

With SDM, shared secret authentication is used during de-identification/encryption of the data as well as during identification/decryption of the data while performing the undo business flow for IPMC. The username/password along with "LDAP + Shared Secret" mechanism is used during identification/decryption of the data. For more information on authentication mechanisms and required setup of DPP Appliance (SDA), see *OpenText™ Data Privacy and Protection Appliance Admin Guide* .

On the Masking Server Settings page, you can specify the masking server and options.

To manage masking server settings

1. Navigate to **Settings** in the Web Console.

2. Click **Masking Server**.

   The DPP Appliance (SDA) Settings page opens.

3. Specify the DPP Appliance (SDA) settings:

   1. **Masking Product.**

      - **FPE.** Indicates that SDM will use DPP Appliance (SDA) Format Preserving Encryption (FPE) as the masking server.

> **Note**
>
> Ensure that a DPP server (machine where DPP Appliance (SDA) is installed) is defined in Web Console so that the SDM can communicate with the DPP Appliance (SDA) .

2. **Shared Library/DLL Path.** Specify the path to `vibesimplejava.dll`, based on your installation of SimpleAPI installation directory.

3. **Server Base URL**. Specify the base URL of the DPP Appliance (SDA)
   `protocol://<serverDNS:Port>`
   where

   Example
   http://mySecureDataServer.myDomain.com
   https://mySecureDataServer.myDomain.com

   > **Note**
   >
   > The default value of **Server Base URL** points to the public instance of DPP Appliance (SDA) . This value is provided to get an idea of the masking or unmaking functionality. The Data Access Cartridge (DAC) with pseudo columns having these DPP Appliance (SDA) access or protect functionality will not perform any encryption/decryption when the public instance is used as it does not support the username/password based authentication. To get this feature in DAC, the customer must have the private instance of DPP Appliance (SDA) .

   - protocol is HTTP or HTTPS
   - serverDNS is the server DNS or IP address
   - Port is the port number of the DPP Appliance (SDA)

4. **Default Identity.** Specify the default identity value used, if the per format identity is not provided. This specified identity value is also used for AUTO format while testing the connection.

5. **Shared Secret.** Enter the shared secret in the **Shared Secret** box.

6. **Allow Short FPE?** For some of the DPP Appliance (SDA) formats the input data length of one character will result into an error `FPE input with a length too short to encrypt was encountered`. You can overcome this by setting **Allow Short FPE** to **Yes**. We recommend you to set this to **Yes** to avoid runtime errors while executing business flows with DPP Appliance (SDA) masking functions.

7. Enter the data to be masked or unmasked in the **Input Data** box.

   This is an optional field, not required if you use the **Test Connection**, but required test the masking or unmasking.

8. Click **Test Connection** to test the connection to the masking server.

9. Click **Test Mask** to test the data masking.

10. Click **Test Unmask** to test the data unmasking.

    The Output Data box displays the masking or unmasking results.

    1. Optionally, click **Clear** to clear the values in the **Input Data** box.

    2. Click the arrow button to copy the value from the **Output Data** box to the Input Data box.

       This allows you to quickly verify the masking and unmasking feature.

4. Click **Apply.**

   > **Note**
   >
   > The special characters ., <, >, :, ", /, \, |, ?, and * and blanks are not allowed in the Function Prefix box.

   The DPP Appliance (SDA) Format - SDM Functions page opens. This page allows you to enter the display names for the FPE/SDM functions and also for the per format identity.

5. Enter a function prefix in the **Function Prefix** box.

6. Click **Generate** to generate the functions.
   Once you generate the functions, the page status bar displays shows the number of functions generated and the directory

name where the Groovy files are stored.

Use this location to import the FPE/SST functions into SDM Designer.

1.2.1.6. Manage the home directory

Structured Data Manager stores its program data, such as configuration, log, and archive files, in a location that is separate from its program files. This home directory is determined at installation time:

- By default, Structured Data Manager attempts to configure its home directory in the standard location for application data on the operating system. For example, on Windows, the Installer tries to use `C:\`.

After installation, you can change the home directory by following the instructions in [Change the home directory](#).

Example

Configuration files might be stored in:

`<OBT_HOME>/config`

> **Tip**
>
> On Windows, the `ProgramData` directory may be hidden by default. You need to show that directory in order to find the log files. See your MS Windows documentation for information on displaying hidden folders and files.

Change the home directory

To change the home directory for Structured Data Manager:

1. Stop all Structured Data Manager processes, including Web Console, Designer and AQS.

2. Open *<install_dir>*`\obt\config\obt.env` in a text editor and change existing path to the new path.

   For example:

   `OBT_HOME=C:/SDM/OBTHOME`

   > **Note**
   >
   > Notice that, even on Windows, the directory separator is a forward slash (/).

3. In your file system, copy all of the directories and files except `aqsdatasources` from your current home directory to the new one that you just specified in `obt.env`.

4. For `aqsdatasources`, run the following command from *<install_dir>*`\obt\bin`:

   `oacommandexec -m <data src name> <new location>`

   For example:

   `oacommandexec -m xmlArchive D:\home1\aqsdatasources\xmlArchive`

5. Restart Structured Data Manager processes, such as Web Console and AQS.

Move the OBTHOME to a new folder (same or different drive)

To move the folder OBTHOME for Structured Data Manager:

1. Stop all Structured Data Manager processes, including Web Console, Designer and AQS.
2. Copy the existing folder *OBTHOME* to a new folder *NEW_OBTHOME*.
   For example:
   On Windows
   from `C:\SDM\OBTHOME` to `H:\NEW_OBTHOME`
   On Linux
   from `/home/user/SDM766/OBTHOME/` to new folder `/home/user/NEW_OBTHOME`
3. Open *<install_dir>*`\obt\config\obt.env` in a text editor and change existing path to the new path.
   For example:
   On Windows
   `OBT_HOME=H:/NEW_OBTHOME`
   On Linux
   `OBT_HOME=/home/user/NEW_OBTHOME`

4. Restart Structured Data Manager processes, such as Web Console, Designer and AQS.

1.2.1.7. Create and edit Web Console users and groups (optional)

In most cases, more than one user needs to perform functions in the Web Console, but not every user needs the same privileges. For example, some users might only need to run jobs while others may need to deploy business flows, set parameter values, and run jobs. Web Console provides a user model that enables you to create as many users as you need, and grant or withhold privileges. For cases where you have a many users to manage, you can also create groups to make assigning privileges and access easier.

Create or edit a user group

Groups are used to assign privileges and environments to users.

> **Note**
>
> Typically, you should have at least one group that has all privileges enabled and has at least one member. By default for new installations, Structured Data Manager creates such a group for you and names it Administrators. The user you created when you installed the repository is assigned to this group. Similarly, for new default installations, Structured Data Manager creates a group called Users that has the necessary privileges to run business flows and change parameters.

To create or edit a user group

1. Launch the Web Console in your browser and log in. The Web Console home page is displayed.

2. Navigate to **User Management > Users > Groups**. The Groups page opens.

   If you have not previously created any other groups, you should only see the Administrators and Users groups created by default during initial configuration Web Console.

3. Click **New Group** or click on an existing group.

   The Create/Edit Group page opens.

4. Enter or modify a **Group Name**.

   > **Note**
   >
   > For an LDAP group, use underscores in place of spaces in the name.

5. Use the **Add** and **Remove** buttons to specify which of the following available privileges you want to assign to this group. For the purposes of this example, you do not need to make any changes.

Privilege	Description
Administrative	Indicates the user is the Administrator. Having the Authenticated Administrator privileges does not imply any of the other privileges. Users with this privilege still require the other privileges in order to perform those functions.
Environment Management	Can create and edit environments and locations, and grant permissions for users to access an environment.
User Management	Can create, edit, and delete Web Console users, and grant or revoke permissions for users to access environments.
Access Discovery	Can log in to Discovery.
Access Web Console	Can log in to the Web Console.
Access data	Can view application data, for example, run data access cartridges. Note that, if you grant a user only this privilege and no data access cartridges are deployed, the user will not be able to do anything in the Web Console.
Access extension	Can access Web Console extensions. **Note** If the extension specific privileges are available then they take priority over this general privilege for extensions. This is available to support the legacy extensions which could have been developed when extension specific privileges were not available. **Tip** See the *OpenText™ Structured Data Manager Developer's Guide* > Appendix A: Extensions> Tile and Access Control for more information on extension specific privileges.
Business flow deployment	Can deploy business flows directly into the environments they have access to.
Call web service	Can launch and schedule jobs through Web services, for example, deploying into an environment from Designer using a remote connection.
Change/update application parameter	Can set configuration parameters at the cartridge level. Can set business flow parameters. Cannot set database to database and database to file configuration parameters that apply to all cartridges, but can view the parameters.

Privilege	Description
Change/update system parameter	Can set database to database and database to file configuration parameters that apply to all cartridges. Can set database to database and database to file configuration parameters at the cartridge level. Cannot set business flow parameters, but can view the parameters.
Download data	Allows the group users to download query results from DAC and Free Text Search as HTML, XML, PDF, and CSV.
Run jobs	Can run business flows and other Structured Data Manager jobs.

6. Click **Members**.

7. Use the **Add** and **Remove** buttons to select which users to make members of this group. For the purposes of this example, make the user you just created in Create or edit a user a member of this group.

8. Click **Environments**.

9. Use the **Add** and **Remove** buttons to select the environments to which this group has access. For the purposes of this example, give the group access to the environment you created in Create an environment.

10. Click **Save**.

Create external groups

1. Create following external groups
   - SDM_ADMINS
   - SDM_USERS
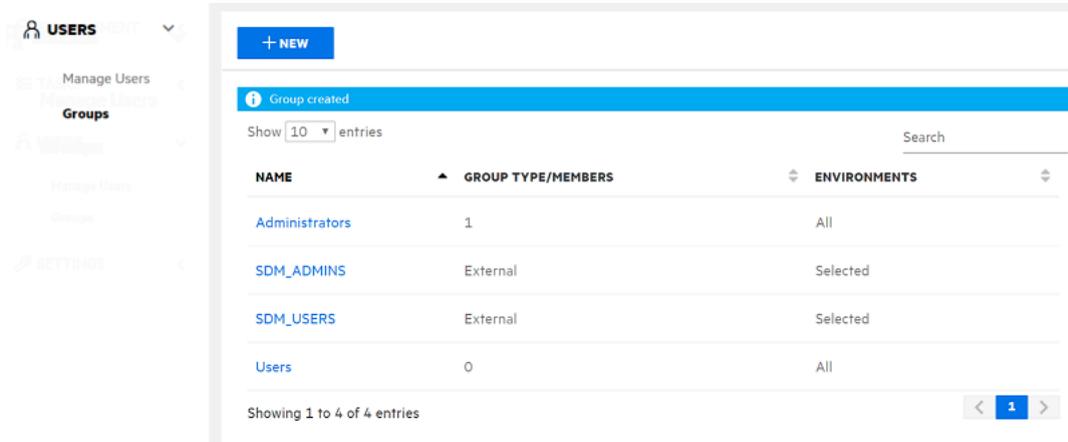
2. Group SDM_ADMINS has all the privileges giving it administrator role.
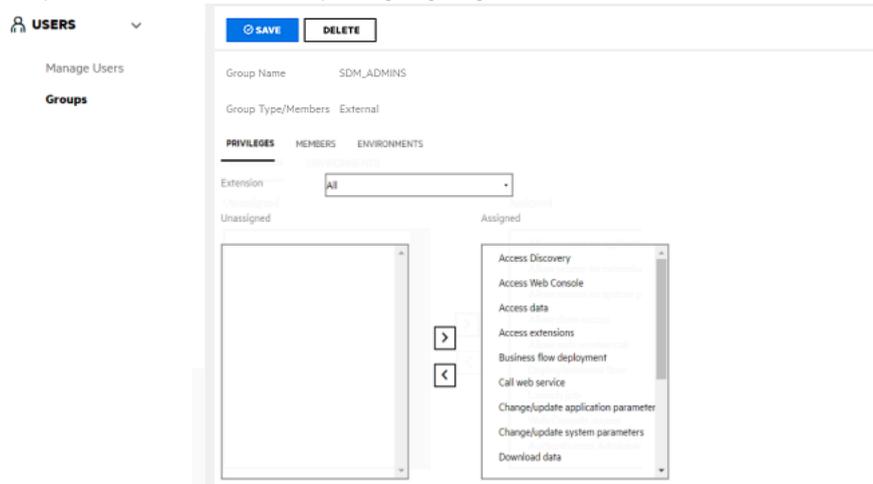
3. SDM_USERS group has no administrative privileges
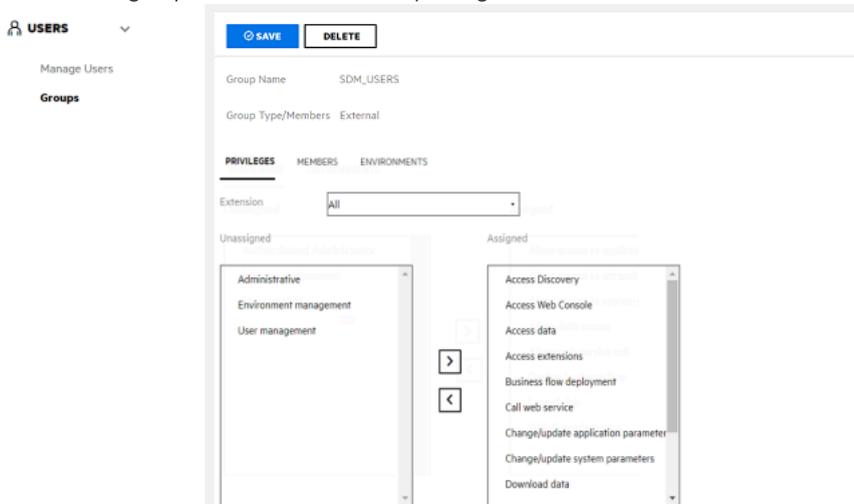
Create or edit a user

Privileges are granted to the groups. Therefore, a Web Console user who belongs to the designated group inherits the group privileges of each environment.

To create or edit a user

1. Launch the Web Console in your browser and login. The Web Console home page is displayed.
2. Navigate to **User Management > Users > Manage Users**.
   The User List page is displayed. If you have not previously created any other users, you should only see the user you created when you installed the repository, for example, `admin`.
3. Click **New** or an existing user.
   The Create/Edit User page displays.
4. If external authentication (such as LDAP) is enabled, you see the **Type** field on this page. If you see it, choose **Built-in** for the purposes of this tutorial. Otherwise, you can skip this step.

   > **Tip**
   > Even if you have external authentication (such as LDAP) configured, you may want to maintain at least one built-in user that you can use to login should your LDAP directory go down for any reason.

   > **Note**
   > A Web Console user can be a native Web Console user or an external user, when the external authentication (such as LDAP) is enabled. If the user is an external user (for example, they belong to an external group), then the Web Console group name must be exactly equal to the external group name.

5. Enter at least the following values:

   - **Login Name:** tom_baker

- **Full Name:** TOM BAKER

- **Password**: welcome

- **Enabled**: checked

6. Use the **Add** and **Remove** buttons to specify to which of the available groups this user should belong. For the purposes of this example, you do not need to make any changes.

7. Click **SAVE** at the top of the page.

The Show User page opens.
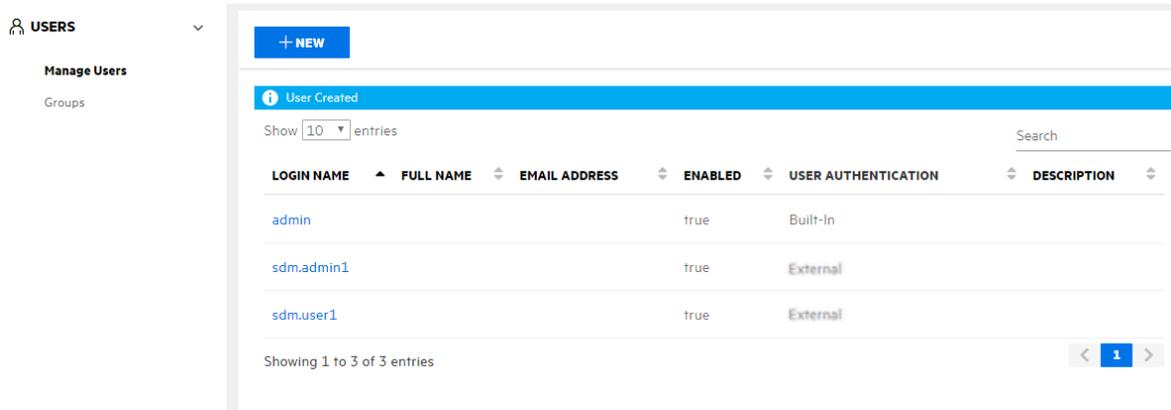
Verify the new Web Console user

To verify the Web Console user

1. Click **Logout**.

2. Log back in as the user you just created.

On the Web Console home page, notice that the **User Management** option does not appear because the user does not have permission to create users.

3. Click **Logout**.

4. Log back in as the Web Console administrator.

Login with **sdm.user1** credentials and observe that administrative privileges are not available on the Web Console home page.
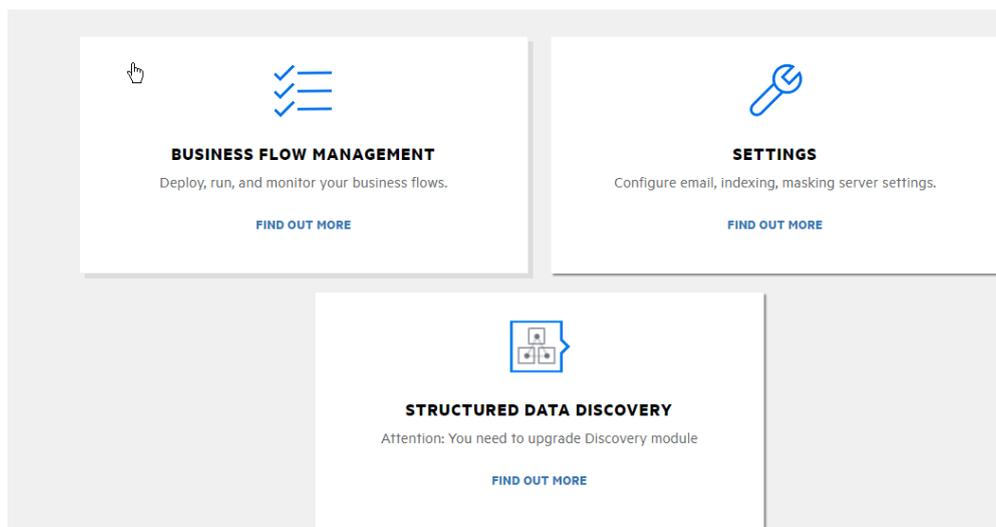
Login with **sdm.admin1** to see that all the administrative privileges are available on the Web Console home page.

BUSINESS FLOW MANAGEMENT

Deploy, run, and monitor your business flows.

FIND OUT MORE

ENVIRONMENT MANAGEMENT

Manage Environments

FIND OUT MORE

USER MANAGEMENT

Manage individual users and user groups.

FIND OUT MORE

SETTINGS

Configure email, indexing, masking server settings.

FIND OUT MORE

STRUCTURED DATA DISCOVERY

Attention: You need to upgrade Discovery module

FIND OUT MORE

1.2.2. Deploy and run business flows

After your Web Console is properly configured, as described in Configuring the Web Console, you are ready to begin deploying and running business flows. Business flows are a series of activities to be run in sequence. These activities include archive, copy, and reload cartridges, as well as other business logic in the form of Groovy scripts.

During and after the running of your business flows, you can monitor their status and review their history within the Web Console.

Before you begin

Before you can run a business flow, it must first be built in Designer along with the required cartridges and, if necessary, models and parameters. After you build the business flow, you need to deploy it in the environment where you plan to run the business flow. You can deploy a business flow from Designer or, if the deployment files are available, from the Web Console.

The tutorial in this section requires the following prerequisites:

- Before you can perform the steps in this tutorial, you must have configured the Web Console. See Configure the Web Console for more information about configuring the Web Console.

- For this tutorial, two Oracle databases (distributed instance) are used. One contains the active data and the other the archive data. If you are using only one database (single instance) and/or another type of database, such as Sybase or SQL Server, you will notice some small differences in the dialogs but the basic principles are the same.

- This Web Console tutorial uses a business flow that was created in a Designer tutorial in the *OpenText™ Structured Data Manager Developers Guide*. If you have not built that tutorial yourself yet, you need to do so before you proceed with this chapter.

- If you have not already done so, generate the deployment files for the archive, undo, and reload business flows from Designer:

  - **Orders_D2D_BF**

  - **Undo_Orders_D2D_BF**

  - **Reload_Orders_D2D_BF**

Place the deployment files in a location that is accessible from the Web Console installation you are using. You will need this file path later on when you import the business flow in the Web Console. See the *OpenText™ Structured Data Manager Developers Guide* for more information on how to generate deployment files.

Deploy business flows

To deploy a business flow

1. Launch the Web Console in your browser and login.

2. From the Web Console home page, click **Business Flow Management > Tasks > Deployment > Deploy**.

3. If you have more than one environment in the Web Console, make sure that you have selected the correct one as the active environment. You can determine which environment is the active one and, if necessary, change it by going back to Web Console home page and click **Environment Management > Environment > Manage**.

4. If you see **Orders_D2D_BF**, **Undo_Orders_D2D_BF**, and **Reload_Orders_D2D_BF** in the list of business flows, select them. Otherwise, you must import these business flows and then select them. If necessary, click **Import** and follow the instructions in the online help to import the business flow into Web Console.

   > 💡 Tip
   >
   > If the list of business flows is long, you can filter the list with the **Show latest versions only** and **Show newly imported business flows only**.

1. Click **Next**. The credentials page displays for you to enter your source and target database credentials, and define the access passwords.

   For Oracle

DEFINE PASSWORDS FOR TARGET AND ARCHIVE ACCESS ACCOUNTS

The passwords are used to provide access to Target and Archive Access schemas.

Target Access Password: *   ••••

Confirm Target Access Password: *   ••••

Archive Access Password: *   ••••

Confirm Archive Access Password: *   ••••

For SQL Server

DEFINE PASSWORDS FOR TARGET AND ARCHIVE ACCESS ACCOUNTS

The passwords are used to provide access to Target and Archive Access schemas.

Target Access Password: *   ••••

Confirm Target Access Password: *   ••••

Archive Access Password: *   ••••

Confirm Archive Access Password: *   ••••

2. Enter the username and password for the source database administrator. These credentials are needed to access the active database.

3. Because a distributed instance (a separate archive database) is assumed for this tutorial, enter the username and password for the target database administrator. In a single instance environment where the archive database and source database are one and the same, you do not need to perform this step because the source and target credentials are the same.

4. Create and confirm passwords for the target access user and the archive access user.

5. Click **Next**.

6. For Oracle, select the tablespaces for each business flow you are deploying.

   For SQL Server, enter the data and log size.

   For Oracle

   « BACK   NEXT»

   DEFINE TABLESPACES

   Select Cartridges and choose tablespaces below

CARTRIDGES	HISTORY DATA	HISTORY TEMP	HISTORY INDEX	ARCHIVE ACCESS DATA	ARCHIVE ACCESS TEMP
☐ All					
order_cc 1.0.0.0	USERS (34 MB) ▾	TEMP (< 1 MB) ▾	USERS (34 MB) ▾	USERS (34 MB) ▾	TEMP (< 1 MB) ▾
order_c 1.0.0.0	USERS (34 MB) ▾	TEMP (< 1 MB) ▾	USERS (34 MB) ▾	USERS (34 MB) ▾	TEMP (< 1 MB) ▾

   For SQL Server

   « BACK   NEXT»

   DEFINE DEVICE SIZES

   Select Cartridges and enter device sizes below

Cartridges	History Data Size	History Log Size	Archive Data Size	Archive Log Size
☐ All				
d2d 1.0.0.0	10	5	10	5

7. Click **Next**. The Summary page displays for confirming the settings for this deployment.

For Oracle

«BACK	FINISH

SUMMARY

Business Flows to be deployed:

CostUpdateHistArch 1.0.0.0

Contains Cartridges:

ORDER_CC 1.0.0.0
History Data Tablespace: USERS
History Temp Tablespace: TEMP
History Index Tablespace: USERS
Archive Access Data Tablespace: USERS
Archive Access Temp Tablespace: TEMP

ORDER_C 1.0.0.0
History Data Tablespace: USERS
History Temp Tablespace: TEMP
History Index Tablespace: USERS
Archive Access Data Tablespace: USERS
Archive Access Temp Tablespace: TEMP

For SQL Server

«BACK	FINISH

SUMMARY

Business Flows to be deployed:

d2d_bf 1.0.0.0

Contains Cartridges:

D2D 1.0.0.0
History Primary Data size: 50
History Transaction Log size: 25
Archive Access Primary Data size: 10
Archive Access Transaction Log size: 5

8. Click **Finish** to deploy the business flows. It may take some time for all three business flows to deploy.

Run a business flow

After you have deployed your business flows, you can run them.

To run a business flow

1. In **Tasks**, click **Launch > Business Flow**. The Business Flows page displays the list of deployed business flows, which you can run.

> 💬 **Note**
>
> The undo business flow, Undo_Orders_D2D_BF, is not displayed in this list. An undo business flow is associated with and can only be run in relationship to another business flow. For example, you can only run Undo_Orders_D2D_BF after Orders_D2D_BF has run. In Run an undo business flow, you will walk through the process of running an undo business flow.

2. Select the **Orders_D2D_BF** business flow. The Job Launch page displays for the selected business flow with its runtime parameter, `Min_Months_to_Retain`. The default value, 24, for the parameter is fine for this example.

3. Click **Run** to run the business flow.

4. Click **OK** to confirm.

5. Click **Monitoring** from the menu at the top of the page. The Business Flow Status page displays the business flow status and any currently running commands.

6. Click **Refresh** until the status of the business flow changes to COMPLETED. This status indicates that the business flow has completed successfully.

7. Query your archive database. The schema named DEMARC_HIST should contain tables populated with the archived rows from DEMARC.

8. Query your active database. The DEMARC tables should no longer contain the rows that were moved to DEMARC_HIST.

Run an undo business flow

Suppose that, after running your archive business flow and moving the data, you realize that you need to bring it all back into the active database. For example, perhaps you included too much data in the archive. In this circumstance, you want to quickly return the active and archive databases to the state they were in before you ran your archiving business flow. To return the moved data to the active database, you can run an undo business flow. You must create an undo business flow in Designer and deploy it to your environment. See the *OpenText™ Structured Data Manager Developer's Guide*.

To run an undo business flow

1. Return to the **Web Console > Business Flow Management > Tasks > Monitoring** page.

2. Click the **Orders_D2D_BF** business flow to navigate to the Run Details page. On the Run Details page, you should see an **Undo** button above the business flow status table. This button only appears if an undo business flow is deployed.

3. Click **Undo**.

4. Click **OK** to confirm.

5. Click **Monitoring** from the left pane. The Business Flow Status page displays the business flow status and any currently running commands.

6. Click **Refresh** until the status of the business flow changes to COMPLETED.

7. Query your archive database. The schema named DEMARC_HIST should have been returned to its state prior to having run the archive business flow. In this particular case, the tables should all be empty now.

8. Query your active database. The DEMARC schema should have been returned to its state prior to having run the Orders_D2D_BF business flow.

Run a reload business flow

Just because you archive a transaction today does not mean that it will not become active again at a later date. For example, you might have archived an order that later becomes active again due to a product recall or a customer payment issue. In such cases, you might need to bring the transaction back to the active database for further activity. To return archived data to the active database based upon some specified criteria, such as an order number, you can run a reload business flow. You must create a reload business flow in Designer and deploy it to your environment. See the *OpenText™ Structured Data Manager Developer's Guide*.

> **Tip**
>
> It is always a best practice to create and deploy a reload business flow for each of your database to database archiving business flows. For more information about how to create a reload business flow, see the *OpenText™ Structured Data Manager Developer's Guide*.

Before you begin

If you followed the steps in Running an undo business flow, the target database no longer contains any data for you to reload. Before performing the steps in this section, you must re-run the Orders_D2D_BF business flow to populate the target database:

Follow the instructions in Running a business flow to launch the archive business flow again. After it successfully completes, you can proceed with the rest of this tutorial.

To run a reload business flow

1. Return to the **Web Console > Business Flow Management** page.

2. In **Tasks**, click **Launch > Business Flow**. The Business Flows page displays the list of deployed business flows that you can run.

3. Select the **Reload_Orders_D2D_BF** business flow. The Job Launch page displays for the selected business flow with its runtime parameter, `OrderID_to_reload`.

4. For `OrderID_to_reload`, enter an ORDERID that you are certain was archived by the Orders_D2D_BF business flow, for example, 1402. If necessary, you can also query DEMARC_HIST in the archive database to find an ORDERID that was archived. Enter one of the ORDERID values that you find in the DEMARC_HIST.ORDER_HEADER table.

5. Click **Run**.

6. Click **Confirm**.

7. Click **Tasks > Monitoring > Overview**. The Business Flow Status page displays the business flow status and any currently running commands.

8. Click **Refresh** until the status of the Reload_Orders_D2D_BF business flow changes to COMPLETED.

9. Query the DEMARC_HIST.ORDER_HEADER table to ensure that the specified ORDERID has been removed.

10. Query the DEMARC.ORDER_HEADER table to ensure that the specified ORDERID has been restored.

11. If you plan to attempt other tutorials at this point, you need to perform one of the following procedures to restore the DEMARC sample data in the source database to its original state. Otherwise, you may get inconsistent results when you attempt the other tutorials.

    - Recreate the DEMARC schema by reloading the data with the load_demo script according to the instructions in Chapter 2, Configuring the Demarc data, in *OpenText™ Structured Data Manager Tutorial*.

    - Undo the Orders_D2D_BF the same way that you did in Run an undo business flow.

> **Note**
>
> During D2F archiving, Structured Data Manager stores dates in UTC. These dates can be translated to any other timezone when reloading (undoing) or uploading to a database including the AQS cache.

1.2.3. Upload to Vertica

Uploading data from your applications to Vertica provides superior access and analytics. With Structured Data Manager, you can set up jobs to take data from your database applications and quickly load the data into Vertica.

Introduction

To archive data to Vertica, you just need a Structured Data Manager instance installed and running on a system with access to Vertica and the necessary privileges to write to the file system and insert data into the Vertica instance.

In the Web Console, you create a location that points to your Vertica database, which becomes the destination for the business flow.

> **Tip**
>
> When uploading to Vertica, you should pre-create the schema into which you want to upload, for example `create schema demarc`.

Bear in mind the following considerations before uploading data into Vertica:

- Character columns (such as VARCHAR, TEST, NTEXT, IMAGE, CLOB, and CHAR) cannot exceed 65000 bytes.

- Binary types (such as BLOB, BINARY, and VARBINARY) cannot exceed 12500 bytes.

- When running the same job to Vertica multiple times and inserting/modifying rows in existing tables, you need a data movement key for each table. Otherwise, only the first run will succeed and all of the subsequent runs will fail. In most cases, Structured Data Manager can automatically select a unique key to use as the data movement key. In cases where it cannot, you may have to manually select one. For more information about data movement keys, see the chapter on working with models in the *OpenText™ Structured Data Manager Developers Guide*.

- If your source database is Sybase and the tables you plan to archive make use of user-defined data types, you must ensure that those data types are available to the interface database created by Structured Data Manager. For example, you might define the data type in the interface database or the model databases.

Create a Vertica location

To create a Vertica location to which you can upload data

1. Launch the Web Console in your browser and log in.

2. Create a new environment or make the environment from which you want to run your business flow the active environment, **Environment Management > Environment > Manage**.

3. Click **Environment Management > Environment > Active Environment > Locations**.

4. Click **New**.

5. Enter a name for the location, for example `salesorders_vertica_db`.

6. For **Type**, choose **Vertica**.

7. Enter the connection information for your Vertica database.

Deploy the business flow

To upload to Vertica, you need a business flow with a database to file archive activity and an upload activity.
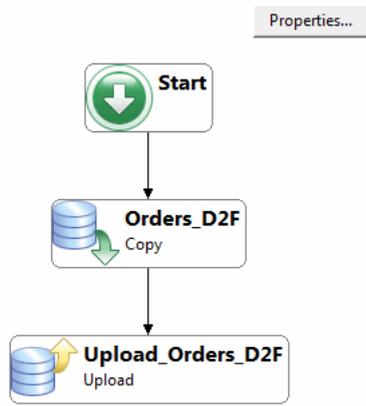
Deploy the business flow from Designer or Web Console. See the *OpenText™ Structured Data Manager Developers Guide* or [Deploy business flows](#).

Specify a schema mapping file

When uploading to Vertica, you typically want to insert your data into a specific schema. For example, in our example, we want the data to go into a schema named `demarc` that you create in Vertica. If you do not specify a schema, the data will go into a general area of Vertica and you will likely have issues with subsequent upload operations.

Depending on your source database, you might also need to perform some mapping of the schema for the Vertica database. For example, suppose that your source data is in SQL Server and uses `DEMARC.dbo`, but, in Vertica, you simply want the schema to be `demarc`. A simple schema mapping file would take care of that difference.

To map your schema for upload to Vertica

1. Before running the upload to Vertica, open a vSQL session and create a schema called `demarc`:

```
salesorder=> create schema demarc;
CREATE SCHEMA
salesorder=> select schema_name from schemata;
schema_name
-------------
v_internal
v_catalog
v_monitor
public
demarc
5 rows)
```

2. Create a simple schema mapping file in a text editor called, for example, `upload_vertica.properties` with the following contents:
   `DEMARC.dbo=demarc`

3. In the Web Console where you plan to run your database to file business flow and the upload to Vertica, go to **Business Flow Management > Tasks > System Parameters**.

4. Click **Cartridge**.

5. Click the cartridge from which you plan to upload data, for example, `Orders_D2F`.

6. Scroll down to **File Naming >Schema Mapping File for Upload** and enter the path and file you created in [step 2](#), for example:

```
C:\vertica_schema_mapping\upload_vertica.properties
```


7. Click **Apply**.

Run the business flow

To run your database to file business flow

1. Go to **Business Flow Management > Tasks > Launch > Business Flow.**

2. Click the business flow you want to run.

3. Ensure that the **Destination DB Location** is properly set to the Vertica location you created earlier.

4. Click **Run**. When prompted, confirm that you wan to run the business flow. The business flow runs, archiving you data to file and then uploading it to Vertica.

Access the archived data

If the archive developer creates and deploys data access cartridges, users can use them to explore the archived data. Pre-defined data access is available by clicking **Data Access** from the home page in the Web Console. For information on how a developer can pre-create this kind of data access, see the *OpenText™ Structured Data Manager Developers Guide*. Otherwise, you can simply query the data with vSQL to confirm that it was inserted into the database in the correct schema.

> **Tip**
>
> In order to run data access cartridges, users only require the Access Data privilege.

> **Note**
>
> To run a data access cartridge against Vertica, you must create a location for Vertica and the Vertica instance that you reference must be accessible from that instance of the Web Console.

For more information about Vertica, see the following manuals:

- *Vertica SQL Reference*

- *Vertica Administrator's Guide*

- *Vertica Programmer's Guide*

If you have a customer account, you can obtain these and other Vertica documentation from my.vertica.com.

1.2.4. Perform multiple data movements

Records from your database applications may be required for many different purposes, such as:

- **Compliance.** You may need to store a read-only copy of the data in a system like Consolidated Archive for compliance purposes.

- **Data analytics.** You may want to upload the data to another database location, such as Vertica, in order to perform analytics on it.

- **Long term retention and query.** You may put infrequently accessed records into a file system where you can use the Archive Query Server to maintain SQL access to such records more cheaply than you would by keeping them in your database.

To meet these requirements, you can move data from a variety of sources to multiple targets in one job run using Structured Data Manager.

Introduction

A typical configuration that archives data to multiple remote locations consists of the following:

- A gateway instance of Structured Data Manager running on a system with a data movement configuration that enables it to send data to various stores that you require, such as a network file system, Content Manager, Consolidated Archive, and Vertica.

- An application instance of Structured Data Manager with one or more environments connected to your source, online transaction processing databases. One or more of the environments in this local instance include a location that points to the gateway Structured Data Manager instance.

When the application instance targets jobs to the gateway instance, the gateway Structured Data Manager picks up the data and performs the data movements specified in its configuration. This architecture enables you to satisfy multiple purposes from one archive run.

Configure Structured Data Manager

To configure Structured Data Manager for this case, you need two instances:

- Configure a gateway instance of OpenText™ Structured Data Manager

- Configure an application instance of OpenText™ Structured Data Manager

Configure a gateway instance of Structured Data Manager

To configure your remote Structured Data Manager

1. Install Structured Data Manager as per the *OpenText™ Structured Data Manager Installation Guide*.
2. Start the Web Console and install a repository as per the instructions in Install the repository.

   > 💡 **Tip**
   > Use PostgreSQL for the repository.

3. Create an environment for database to file as per the instructions in Create an environment.

   > 💡 **Tip**
   > Make the environment non-intrusive and use a JDBC URL to a PostgreSQL database for the source.

4. Create a Web Console user specifically to run the data movement jobs for the application instance of Structured Data Manager, for example, `data_move_admin`. For more information, see Create and edit Web Console users and groups (optional).
5. Create locations to which you want to move data, for example Content Manager and Vertica. See Manage environments for more information.
6. Create a new directory in the home directory, under `/config/upload`, using the user name you created in step 4, for example:

   `<OBT_HOME>\config\upload\data_move_admin`
7. Create a data movement configuration file in the directory you created in Step 6. Use the sample file, `username.xml.sample`, which is found in `/config/upload/username`, as a template. Name the file for the target

configuration you specify in Step 9. For example, create a file called `sdm2sdm.xml`:

`<OBT_HOME>\config\upload\data_move_admin\sdm2sdm.xml`

Following is a simple example of the file contents:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<fileLocation environment="OLTP_Safe_Copy"
    location="LOCAL_ARCHIVE_FS" preserve="false"/>
<dataMovements>
  <dataMovement destination="sdm_file_archive"/>
  <dataMovement destination="salesorders_vertica_db"/>
</dataMovements>
</configuration>
```

The first element, `fileLocation`, tells Structured Data Manager where it may place files for data movement for the specified environment, `OLTP_Safe_Copy`. `LOCAL_ARCHIVE_FS` is a file system location defined in the Web Console and, by setting `preserve` to false, you let Structured Data Manager know that it should remove the files after the data movement is done.

The `dataMovement` elements tell Structured Data Manager where to move the archived data. Each dataMovement element references a location defined in the Web Console. For the purposes of this example, we created two such locations in step 5:

- The first location is called `sdm_file_archive`, which happens to be a file system path.
- The second location is `salesorders_vertica_db`, a Vertica database. The archived data will be uploaded to this database.

> **Note**
>
> In the case of Vertica, you should pre-create the schema to which you plan to upload the data. For example, suppose that the archived data came from an Oracle schema called DEMARC. You would create a DEMARC schema in the Vertica database to hold the uploaded data before running the job.

These two examples do not require it, but some locations may require a data transformation. Following is an example of a data transformation within a data movement:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
<fileLocation environment="demarc"
  location="LOCAL_ARCHIVE_FS" preserve="false"/>
<beforeDataMovements>
  println("Hello");
</beforeDataMovements>
<dataMovements>
  <dataMovement destination="LOCAL_TEMP_FS">
    <dataTransformation outputFormat="xml"
    batchSize="1">
  </dataTransformation>
  </dataMovement>
</dataMovements>
<afterDataMovements script="sayHello.groovy"/>
</configuration>
```

Notice also in this case the Groovy scripts to be run before and after the data movements. You can use Groovy to perform preparation and cleanup activities before and after you actual data movements.

Configure an application instance of Structured Data Manager

To configure your local Structured Data Manager

1. Install Structured Data Manager as per the *OpenText™ Structured Data Manager Installation Guide*.
2. Start the Web Console and install a repository according to the instructions in Install the repository.
3. Create an environment (standard or non-intrusive) for database to file as per the instructions in Create an environment.

4. For the environment you just created, go to the Environment page, **Environment Management > Environment > Active Environment**.
5. Click **Locations** to create a location that points to your remote Structured Data Manager.
6. Click **New**.
7. Enter the name to assign to the location, for example, `SDM_Safe`. The location name is case sensitive.
8. For **Type**, choose **Remote Gateway**.
9. Enter the properties for the location:
     - **Remote WebConsole URL** is the URL to the Web Console for your remote Structured Data Manager. For example, `http://rovitto-linux.svs.softwaregrp.net:8080/WebConsole`.
     - **Remote WebConsole user** is the user name on the gateway Web Console that you created in step 4 to run the data movements, `data_move_admin`.
     - **Remote WebConsole password** is the password associated with the remote Web Console user.
     - **Proxy Server Host** is the host name of the proxy that must be used to reach a target Structured Data Manager instance outside of your firewall.
     - **Proxy Server Port** is the port on the proxy server host that you are using to get outside of your firewall.
     - **Use SSL with client certificate** should be false, unless you have https configured for the Web Console. See Configuring HTTPS for information on how to set up https.
     - Optionally, enter **Target Configuration Name**, for example, `sdm2sdm`. This name corresponds to the configuration file name on the gateway Structured Data Manager that you created in step 7.
10. Click **Create**.

Run business flows

To launch your business flow on the application instance and send the data to the gateway instance

1. Deploy one or more database to file business flows in the local Structured Data Manager, where the cartridges optionally reference an indexing cartridge, For more information on deployment, see the *OpenText™ Structured Data Manager Developers Guide*.
2. Make sure the destination for the business flows is the location you created for the remote Structured Data Manager.

> **Tip**
> To set the destination, go to Business Flow Management > Tasks > System Parameters > Business Flow and choose Global BF Settings or a specific business flow.

3. Run your business flows. See the *OpenText™ Structured Data Manager Tutorial* or the Web Console help system for more information about running business flows.
4. Confirm that your archive has successfully run and the data pushed to all the specified locations in your data movement configuration file.
5. For example, in this case, check the file system for your gateway instance and query your Vertica database.

1.2.5. View eligibility analytics

During and after the running of your business flows, you can monitor their status and review their history within the Web Console. Furthermore, if enabled, you can examine any cartridge's eligibility analytics to see which records would be moved or copied based upon the selected rules.

This tutorial takes you through the process of viewing eligibility analytics, including:

- Deploying business flows

- Enabling eligibility analytics

- Running business flows

- Monitoring status

- Viewing eligibility analytics

Before you begin

Before you can run a business flow, it must first be built in Designer along with the required cartridges and, if necessary, models and parameters. After you build the business flow, you need to deploy it in the environment where you plan to run it. You can deploy a business flow from Designer or, if the deployment files are available, from the Web Console.

The tutorial in this section requires the following perquisites:

- Before you can perform the steps in this tutorial, you must have configured the Web Console. See Configure the Web Console for more information about configuring the Web Console.

- This Web Console tutorial uses a business flow (**Orders_D2F_BF**) that was created in a Designer tutorial in Chapter 5, Eligibility analytics, of the *OpenText™ Structured Data Manager Developers Guide*. If you have not built that tutorial yourself yet, you need to do so before you proceed with this tutorial.

- Make sure that the **Orders_D2F_BF** business flow is deployed from Designer (right click it and choose **Deploy**, then select the **Generate** radio button). Place the deployment file in a location that is accessible from the Web Console installation you are using.

- Deploy the **Orders_D2F_BF** business flow to the Web Console instance where you plan to run it, just as you would any other business flow.

Enable eligibility analytics

Using eligibility analytics can affect performance, so the default behavior at runtime is to not generate eligibility analytics. Therefore, if you want to generate the analytics, you must specifically change a parameter to enable them.

> **Note**
>
> You must also have enabled eligibility analytics in the rules in your model in Designer before deploying the business flow, and the model must contain a pause after the selection step in order for you to see them. For more information, see the eligibility analytics chapter in the *OpenText™ Structured Data Manager Developers Guide*.

1. Launch the Web Console in your browser and log in.

2. From the Web Console home page, click **Business Flow Management > Tasks > System Parameters > Database to File**.

3. Scroll down to the section of the page entitled **Performance**.

4. For the Eligibility Analytics Configuration parameter, select **Enable the eligibility analytics**.

Performance			
Combined delete count *	ⓘ	100	DEFAULT
Combined statement count *	ⓘ	50	DEFAULT
Data movement Batch size *	ⓘ	50000	DEFAULT
Eligibility Analytics Configuration	ⓘ	Enable the eligibility analytics. ▾	DEFAULT
Maximum number of parallel workers *	ⓘ	1	DEFAULT
Populate Index Tables	ⓘ	true ▾	DEFAULT
Populate User-Index Tables	ⓘ	true ▾	DEFAULT
Selection Batch size *	ⓘ	500000	DEFAULT

5. Click **Apply**.

Run a business flow

After you have enabled eligibility analytics, you can run the business flow.

1. Click **Tasks > Launch > Business Flow** from the menu at the top of the page.

   The Business Flows page displays the list of deployed business flows.

2. Select the **Orders_D2F_BF** business flow.

   The Job Launch page displays for the selected business flow with its runtime parameters, `Min_Months_to_Retain` and `Min_Selected_Rows`. The default values for the parameters are fine for this example.

3. Click **Definition** to view the Business Flow Definition page.

   The Business Flow Definition page displays the activities that this business flow will run when you launch it. Notice the activity labeled Suspend Job. After records have been selected (but not moved) by the first activity, the business flow pauses, giving you the chance to review the eligibility analytics.

   After you have reviewed the analytics and determined whether to continue, you can continue the business flow or cancel it.

4. Click **Back** to return to the launch page for the business flow.

5. Click **Run** to run the business flow.

6. Click **OK**.

7. Click **Monitoring** from the menu at the top of the page.

   The Business Flow Status page displays the business flow status and any currently running commands.

8. Click **Refresh** until the status of the business flow changes to SUSPENDED. This status indicates that the business flow has paused as per the interrupt included when designing the business flow.

View eligibility analytics

With the business flow paused, you have the opportunity to review the eligibility analytics to determine whether you wish to continue running and move the data, or cancel the job without moving any data.

1. From the Business Flow Status page, click Orders_D2F_BF to navigate to the Run Details page.
   Notice how the Archive Selection Step and the Job Paused Step have completed, and the overall business flow is suspended.

2. From the Run Details page, click **Eligibility Analytics**.

   The Eligibility Analytics page displays. Eligibility analytics are generated on a per-cartridge basis. Therefore, this page displays the cartridges in the business flow for which you can view eligibility analytics.

   Notice the drop-down list next to the title of the table and next to each rule in the table.

3. Click the list next to the title of the table and select **Selection Data**.

The eligibility analytics for the cartridge displays.

4. If not already selected, select **All Data** in the left navigation pane.
   - Rows that were excluded by a rule appear in red and the name of the rule that caused their exclusion appears in the **Excluded By** column.
   - Rows that were selected for movement appear in black and the **Excluded By** column is empty.

> **Tip**
>
> To see only rows that were included (those that appear in black), select Selection Only. To see only rows that were excluded by the rules (those that appear in red), select Exclusion Only.

In some cases, you may find it useful to save the eligibility analytics for further review or future reference or both. You can easily save this data into an easily accessible format like XML or comma separated values (CSV).

5. Under **Export to**, click **CSV** to save the data to a CSV file.

   CSV files can be opened or processed by many standard programs, such as spreadsheet applications. The operating system dialog for saving to a file displays.

6. Save the file under a name and location of your choice.

7. Open the CSV file in a spreadsheet program of your choice to further review the data.

8. (Optional) Explore the other options on the Eligibility Analytics page:
   - Return to the Run Details page by clicking **Monitoring > Orders_D2F_BF**.
   - Click **Eligibility Analytics**.
   - Click the various lists and select the other options to see what information is available to you.

Continue the paused business flow

For the purposes of this tutorial, assume that you are satisfied with the eligibility analytics review. You are now prepared to continue the business flow and actually move the data from the active database to the archive file.

1. Return to **Web Console > Business Flow Management > Tasks > Monitoring > Overview**.
2. Click the **Orders_D2F_BF** business flow to navigate to the Run Details page.
3. Click **Continue** to complete the paused business flow.
4. Click **Monitoring > Overview** to return to the Business Flow Status page.
5. Click **Refresh** until the job finishes successfully.
6. Click the **Orders_D2F_BF** business flow to navigate to the Run Details page.
7. On the Run Details page, you can see the status of the individual activities within the business flow. In this case, all activities should have completed successfully and have a status of COMPLETED.
8. Click **Row Counts** to view the number of rows that were selected and deleted.

Run the undo business flow

In order to return the sample database to its original state, you should run the Undo_Orders_D2F_BF business flow.

1. Return to the **Web Console > Business Flow Management** and click **Tasks > Monitoring** in the top menu on the page.
2. Click **OK**.
3. Click **Monitoring** from the menu at the top of the page.

   The Business Flow Status page displays the business flow status and any currently running commands.

4. Click the **Orders_D2F_BF** business flow to navigate to the Run Details page.

   On the Run Details page, the **Undo** button is displayed above the business flow status table.

5. Click **Undo**.

> **Tip**
>
> Alternatively, you could recreate the original DEMARC schema by reloading the data with the load_demo script according to the instructions in the OpenText™ Structured Data Manager Tutorial.

1.3. Reference guide

This part provides an advanced task reference to assist you in performing specific tasks within the Web Console and from the command line. It includes:

- Start and configure the Web Console
- Secure the Web Console
- Manage environments
- Manage business flows
- Run business flows and jobs from the command line
- Map schemas for upload or AQS cache
- Datatype mapping list for AQS Cache and Upload
- Perform history schema reconciliation
- Use AQS to access database to file output

1.3.1. Start and configure the Web Console

The Web Console is the primary run time interface to Structured Data Manager. You use it to deploy, run, and monitor jobs, and administer environments and users. The Web Console requires a repository in which to store its metadata. Hence, the first time you start the Web Console, you are prompted to install or upgrade the repository.

- Start and stop the Web Console
- Setup the third-party libraries
- Install the repository
- Upgrade an existing repository
- Add Web Console nodes
- Use log files

1.3.1.1. Start and stop the Web Console

Before you can connect to the interface for the Web Console server in your Web browser and perform tasks, you must start the Web Console server. You can also stop the Web Console server when it is not in use or when performing maintenance activities that require it be taken down.

Start the Web Console

You can start the Web Console from the Start menu on Microsoft Windows or from the command line on any platform.

Before starting the Web Console

- Before running the Web Console, ensure that you are logged into your operating system as a user with appropriate administrative privileges.

- Increase browser session timeout

  > **Note**
  >
  > Browser session timeouts after Web Console remain idle. If you want to increase your browser session timeout modify value of session timeout attribute of session config element in `web.xml` available under `<install_dir>\obt\webconsole\apache-tomcat\conf`

Start Web Console from the program group

> **Note**
>
> **Running Web Console on Different Port**
>
> By default, the Web Console runs on port 8080. If the default port 8080 is already in use by another application, you need to start the Web Console on another port. To do this, modify the value of port attribute of Connector element in `server.xml` (the Tomcat configuration file) available under `<install_dir>\obt\webconsole\apache-tomcat\conf`.

After installing Structured Data Manager, start the Web Console from:

Start > All Programs ><Structured Data Manager_Pgm_Group> > Start Web Console

where <Structured Data Manager_Pgm_Group> is the program group where you installed Structured Data Manager.

A command window opens and closes automatically after the Web Console is started.

Start Web Console from the command line

1. Open a command window and navigate to the directory that contains the Web Console script.

   `cd <install_dir>/obt/bin`

   where *<install_dir>* is the location where you installed the Structured Data Manager.

2. Enter the start command:
   On Windows:
   `webConsole.bat start`
   On UNIX:
   `./webConsole.sh start`
   Structured Data Manager uses Tomcat to run the Web Console.

   - On Windows, a new window is spawned for the Tomcat process.

   - On Linux, a Tomcat process is started.

   Check the Tomcat process and logs in your installation directory to determine if the Web Console started successfully.

> **Tip**
>
> First time when you launch the Web Console, the Structured Data Manager will perform some initial configuration, which may take a longer duration. Hence, ensure you do not interrupt this process. If the Web Console fails to launch or you inadvertently interrupt the initiation process, see the *OpenText™ Structured Data Manager Troubleshooting Guide*.

Start Web Console as a Windows service

1. On Windows, install the Web Console service, if it is not already installed.

   `<install_dir>\obt\bin\webConsoleService.bat install`

   > **Tip**
   >
   > To uninstall the service, run `webConsoleService.bat uninstall`.

2. To start the Web Console service run the command:

   `<install_dir>\obt\bin\WebConsoleService.bat start`

   > **Note**
   >
   > You can also start and stop the Structured Data Manager service from **Control Panel > Administrative Tools > Services**. By default, the service is configured to start automatically during system start. You can also start it manually. Ensure that the user specified to run the service has appropriate administrative privileges on the system.
   >
   > Apart from this, you can stop the Web Console service by running the command:
   >
   > `<install_dir>\obt\bin\WebConsoleService.bat stop`

Launch the Web Console

1. Launch a supported browser, and connect to Web Console using the following URL:
   `http://<hostname>:<port>/WebConsole`
   where *<hostname>* is the name of the machine on which you installed Structured Data Manager, and *<port>* is the port number you specified in the previous step.
2. If you are starting the Web Console for the first time after installation of Structured Data Manager, you are prompted to either install a new repository or upgrade an existing one. See Install the repository or Upgrade an existing repository.

Stop the Web Console

You can stop the Web Console from the Start menu on Windows or from the command line on any platform.

Stop the Web Console from the Start menu

Stop Web Console from the Start menu:

Start > All Programs ><Structured Data Manager_Pgm_Group> > Stop Web Console

where <Structured Data Manager_Pgm_Group> is the program group where installed SDM.

A command window opens and closes automatically after the Web Console is started.

Stop the Web Console from the command line

To stop the Web Console, use the following command:

`webConsole.bat/sh stop`

> **Note**
>
> Before installing the repository, you can set the password policies in `<SDM_dir_HOME>/config/WebConsole.properties` file. For more information, see Set password security

1.3.1.2. Setup the third-party libraries

To explicitly use Oracle, DB2 and Sybase libraries, follow the below steps:

1. Download the libraries from the location mentioned in the below table for your operating system:

Libraries	Location
Oracle	https://mvnrepository.com/artifact/com.oracle.database bc/ojdbc11/23.5.0.24.07 https://mvnrepository.com/artifact/com.oracle.database ml/xmlparserv2/23.5.0.24.07 https://mvnrepository.com/artifact/com.oracle.database ml/xdb6/18.3.0.0
Db2	https://mvnrepository.com/artifact/com.ibm.db2/jcc/12.1 0
Sybase	https://mvnrepository.com/artifact/jdbc.sybase/jconn4/1 0

2. Use Copy Jar utility to copy the jars.

> **Note**
> - On a fresh installation, the **WebConsole** folder will not initially appear under the `<install_dir>\obt\webconsole\apache-tomcat\webapps` directory.
> - To generate it, you must start the Web Console once and then stop the Web Console.

The jars get copied to the following locations:eb Console once and then stop the Web Console.

- `<install_dir>\obt\lib`

- `<install_dir>\obt\ui\plugins\noneclipsedependencies`

- `<install_dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

  You are now able to create or access the respective database environments on the Web Console.

> **Note**
> Core Data Discovery & Risk Insights (CDDRI) users integrating through **sdm-api** must explicitly copy the libraries required to create environment in Discovery to `<install_dir>\ob\webconsole\apache-tomcat\fusionapps\sdm-api\WEB-INF\lib` directory.

1.3.1.3. Install the repository

Structured Data Manager stores metadata in a repository to facilitate the deployment and running of business flows, and administering the Web Console. You can create the repository in an Oracle, or PostgreSQL database. You have the option to connect with the standard Oracle, or PostgreSQL connection formats.

After you install SDM, and start and launch the Web Console, you are prompted to install a new repository or upgrade an existing one. For information about upgrading an existing repository, see Upgrade an existing repository.

> **Note**
>
> For information about database prerequisites for the repository, see the *OpenText™ Structured Data Manager Installation Guide* and the OpenText™ Structured Data Manager *Certification Matrix*.

To install the repository

1. Launch the Web Console in your browser.

   The Repository installation screen displays:
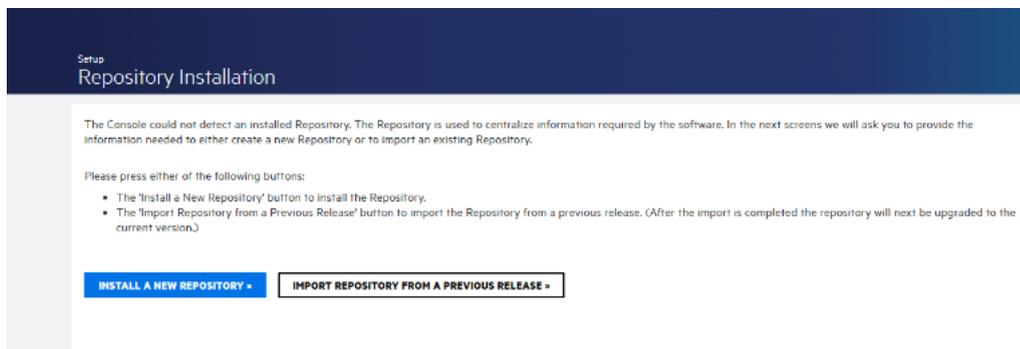
2. Click **Install a New Repository**.

   The Repository Database: Administrator page displays.

3. Select the RDBMS type for your repository and enter the appropriate information:

   > **Important**
   >
   > Oracle and SQL Server will be deprecated from SDM 7.63. Future versions of SDM will support only PostgreSQL as the repository type. Hence, it is recommended to use PostgreSQL as the repository type.

   - Oracle means an Oracle database and connect to it using the basic Oracle connection method.

     > **Note**
     > Prerequisite for Oracle repository
     > Make sure that all required libraries are configured before selecting the Oracle repository. See, Setup the third-party libraries for more information.

   - PostgreSQL means use a PostgreSQL database.

     > **Note**
     >
     > You can only have one Structured Data Manager repository per PostgreSQL database. If necessary, create another database, which can be on the same server.

   Oracle

   An example for Oracle:

REPOSITORY DATABASE: ADMINISTRATOR

The administrator account is used by the deployment assistant to create users and grant access to resources.

Repository Type: Oracle (Deprecated)   ▼

Warning: Future versions of SDM will not support **Oracle** or **SQL Server** as repository type, use **PostgreSQL**.

User:*   system

Password:*   Enter system user password

Host:*   Host IP or Name

Port:*   1521

Service Name:*

* Required Fields   « BACK   NEXT »

1. Enter the following administrator information:

Field	Description
User	Enter the administrator username.
Password	Enter the password for the specified user.
Host	Enter the name of the machine where your Oracle database is installed.
Port	Enter the port number of your Oracle database, for example, 1521.
Service Name	Enter the name of your Oracle database, for example, ORCL.

2. Click **Next**.

The Repository Database: User page displays.

REPOSITORY DATABASE: USER

Create the repository user. Repository user is a database user for logging in to the repository schema.

User:*	obt_rep
Password:*	Password
Confirm Password:*	Confirm Password
Data Tablespace:	XDB (1 MB) ▼
Temp Tablespace:	TEMP (4867 MB) ▼

Repository Database: Encryption Key

The Encryption Key is used as Repository Master Password

Encryption Key:*	Encryption Key
Confirm Encryption Key:*	Confirm Encryption Key

Directories

Home Directory:*	D:\Test\7.6_Screenshots\OBT
Base Archive Directory:*	D:\Test\7.6_Screenshots\OBT\archive

* Required Fields   « BACK   NEXT »

3. Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created. See Special characters for repository password to use special characters in password.
Confirm Password	Enter the password again.
Data Tablespace	Select the data tablespace you want to use.
Temp Tablespace	Select the temporary tablespace you want to use.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.

PostgreSQL

An example for PostgreSQL:

REPOSITORY DATABASE: ADMINISTRATOR

The administrator account is used by the deployment assistant to create users and grant access to resources.

Repository Type:	PostgreSQL ▼
User:*	postgres
Password:*	••••••
Host:*	localhost
Port:*	5432
Database Name:*	postgres
JDBC URL	jdbc:postgresql://localhost:5432/postgres?ssl=true&tcpKeepAlive=true
	ssl = true −
Additional Parameters	tcpKeepAlive = true −
	+

* Required Fields   « BACK   **NEXT »**

1. Enter or select the following administrator information:

Field	Description
User	Enter the database administrator username.
Password	Enter the database administrator password.
Host	Enter the name or address of the machine where the database is installed.
Port	Enter the port number of your PostgreSQL database. For example: 5432
Database Name	Enter the name of the PostgreSQL database, for example, `sdm_repos`. You can only have one repository per PostgreSQL database.
Additional Parameters	(PostgreSQL only) Click on the **+** button if you want to pass any additional parameters while connecting to the database. To remove the added parameter, click - button. The formed JDBC URL is displayed to review the same when you use this option.

2. Click **Next**.

The Repository Database: User page displays.

REPOSITORY DATABASE: USER

Create the repository user. Repository user is a database user for logging in to the repository schema.

User:*   obtrep

Password:*   Password

Confirm Password:*   Confirm Password

Repository Database: Encryption Key

The Encryption Key is used as Repository Master Password

Encryption Key:*   Encryption Key

Confirm Encryption Key:*   Confirm Encryption Key

Directories

Home Directory:*   D:\Test\7.6_Screenshots\OBT

Base Archive Directory:*   D:\Test\7.6_Screenshots\OBT\archive

* Required Fields   « BACK   NEXT »

3. Select or enter the following information for the repository user:

Field	Description
User	Enter the name you want to use for the repository user. The default value is obt_rep.
Password	Enter the password for the user you created. See Special characters for repository password to use special characters in password.
Confirm Password	Enter the password again.
Encryption Key	Enter the encryption key you want to use.
Confirm Encryption Key	Enter the encryption key again.
Home Directory	Modify the home directory for Structured Data Manager or accept the default value.
Base Archive Directory	Modify the directory where archive data is stored or accept the default value.

4. Click **Next**.

The Console Administrator page displays.

CONSOLE ADMINISTRATOR

The Console Administrator is initially the only user authorized to connect to the Web Console.

User Name*	admin
Password*	Password
Confirm Password*	Confirm Password
Real Name	Users display name
Description	User Description
Email	Contact email address
Phone Number	Contact phone number

5. Select or enter the following information:

> **Note**
>
> By default, this user is assigned to the Administrators group, which has all privileges enabled.

Field	Description
User Name	Enter the name you want to use for the Web Console administrator. The Web Console administrator is used to log on to the Web Console.
Password	Enter the password for the administrator.
Confirm Password	Enter the password again.
Real Name	Optionally, enter the name of the administrator user.
Description	Optionally, enter a description.
Email	Optionally, enter an email address.
Phone Number	Optionally, enter a phone number.

6. Click **Next**. The Summary page appears for your review.

7. Click **Finish**.

It may take several moments for the repository creation to complete. When the repository is created, the following message displays:

```
You have successfully installed the repository. Please do NOT close your browser. Web
Console will restart.
```

After the Web Console restarts, the login screen displays.

> **Tip**
>
> If for some reason the repository installation should fail, you need to stop and start the Web Console before trying again.

8. Enter the Web Console administrator user name and password that you just created.

9. Click **Login**.

You are now logged in to the Web Console and can create a new environment.

4. Click **Next**.
The Summary page displays.

5. Click **Finish** to create the repository, or click **Back** to make changes.
The following message displays:
`You have successfully installed the repository. Please do not close your browser. Web Console will restart.`

After the Web Console restarts, the login screen displays.

Special characters for repository password

Below are the supported and unsupported special characters you can use for the repository password.

Supported special characters	Unsupported special characters
`` ` `` - ; ~ _ < ! = > @ + , # [. $] ? ^ { : & } / * | ( )	% \ " '

Login to Web Console

To login to Web Console:

1. (Once the login screen is displayed after Web Console restart) Enter the Web Console administrator user name and password that you just created as per the previous section Install the repository.
2. Click **Login**.

You are now logged in to the Web Console and can create a new environment.

1.3.1.4. Upgrade an existing repository

The first time that you start the Web Console after installing Structured Data Manager, you must install a new repository or import one from an earlier version. For information on installing a new repository, see Install the repository.

> **Note**
>
> If you have used SDM Designer before upgrading the repository and made modifications to `outerbay.properties` file then these changes will be available for you to refer to in `outerbay.properties.pre_upgrade` file.

For information on upgrading an existing repository, see the *OpenText™ Structured Data Manager Upgrade Guide*.

1.3.1.5. Add Web Console nodes

In some cases, you may want to have additional Web Console nodes on other machines that operate upon the same repository. Having multiple Web Console nodes enables you to perform operations against the same repository from any one of several different machines, thus avoiding any performance bottlenecks on the primary Web Console node. By copying certain files from your first or primary node of the Web Console, you can effectively clone the primary node on other machines.

> **Note**
>
> This procedure assumes that you have already installed and fully configured one instance of Structured Data Manager according to the instructions in the *OpenText™ Structured Data Manager Installation Guide* and Install the repository.

To configure an additional Web Console node

1. From the primary node, where you first installed Structured Data Manager and configured the Web Console, make copies of the following files in the Structured Data Manager program data directory and move them to a location accessible to the machine where you want to create an additional node:

   - `<prime_app_data>\SDM\OBTHOME\config\connection-sources.xml`

   - `<prime_app_data>\SDM\OBTHOME\config\hpdbarchiving.auth`

   - `<prime_app_data>\SDM\OBTHOME\config\sql_tuning.properties`

2. On the machine where you want to create the additional Web Console node, use the Installer to create and populate an installation directory tree just as you would for a new installation. Follow the instructions in the *OpenText™ Structured Data Manager Installation Guide*, but do not start the Web Console.

3. Apply all of the patches that were applied to the original tt tree.

> **Note**
>
> Be aware that the patches will attempt to copy new files into the tt tree and that some patches will then attempt to execute scripts to perform tasks (such as compiling packages in databases). However, these scripts will fail when executed on servers that are strictly Web Console servers; this is not a problem, though, because the critical task that must be completed is simply that the files are copied.

1. Overwrite the following files on the clone machine with the copies you made from the primary node:

   - `<clone_app_data>\SDM\OBTHOME\config\connection-sources.xml`

   - `<clone_app_data>\SDM\OBTHOME\config\hpdbarchiving.auth`

   - `<clone_app_data>\SDM\OBTHOME\config\sql_tuning.properties`

   5. Start the cloned Web Console node as described in Start and stop the Web Console.

   6. Confirm that you are operating upon the same repository as the primary node. You should not be prompted to install or upgrade a repository as you normally would for a new installation of the Web Console. Furthermore, if you are logged in as the administrator user, you should be able to see any environments that were created from the primary node.

1.3.1.6. Use log files

When reviewing job status or diagnosing an issue, it can sometimes be helpful to view the complete information provided in log files for Structured Data Manager. If you are regularly consulting the log files for a particular purpose, you may also want to adjust the logging properties to include or exclude certain information.

For detailed information on log files, see *Chapter 2: Log files* in OpenText™ Structured Data Manager *Troubleshooting Guide*.

View log files

Error messages from the Web Console and SDM are captured and appended to log files. By default, the log files are saved to the home directory under `log`. See Manage the home directory.

To view log files

1. Navigate to the directory containing the log file.

   Example

   ```
   cd <OBT_HOME>/log/
   or
   cd <OBT_HOME>/log/<envName>/
   ```

2. Open one of the following log files using a text editor of your choice:

File name	Description
obt.log	Captures logging information that is not specific to an environment. For example, repository install or Web Console. New lines appended to the log file are displayed as they appear. Multiple log files are numbered in sequence, for example, obt.log1.
env_<envName>.log	Captures logging information that is specific to an environment but not to a run. For example, deploying a business flow. The environment log file is located in the home directory under `log/<envName>`. See [Manage the home directory](#).
<envName>_run_<runID>.log	Captures logging information for a specific business flow run. The business flow log file is located in the home directory under `log/<envName>`.
pdm_server_<runID>.log where <runID> is the run ID of the job as displayed on the Web Console.	The pdm_server_<runID>.log file is used for advanced data selection. The log file contains the generated statements, execution plans, and statistics from the execution of the advanced data selection.
GroovyScript.log	Captures logging information that is specific to errors in the Groovy scripts. The log file is located in `<OBT_HOME>/log/BusinessFlow@<NameBusinessFlow>` where `<NameBusinessFlow>` is the name of your business flow.
tomcat logs: localhost_access_log.<date> catalina.<date> localhost.<date> host-manager.<date> manager.<date> Where, <date> is the date on which the log file is generated.	Captures logging information that is specific to tomcat. The log file names are appended with the date. The log files are located in `<install_dir>\obt\webconsole\apache-tomcat\logs`.

Edit the obt.log logging properties

You can edit the `log4j2.properties` file to change the following logging properties for the `obt.log` file:

- Where the log files are kept
- What information is logged
- Maximum size of the log file

- How much logging information is kept

By default, the `obt.log` file is limited to 100 MB. When it exceeds the default limit, the `obt.log` file is renamed to a backup file, and a new `obt.log` file is created. By default, a total of nine backup files are kept, limiting the log files to a 1000 MB maximum size.

To edit the obt.log properties

1. Navigate to `config` under the home directory, which contains the `log4j2.properties` file. See Manage the home directory.

2. Create a backup copy of `log4j2.properties` that you can roll back to in case of any problems.

3. Open the `log4j2.properties` file with a text editor of your choice.

4. Make your desired changes to the file.

5. Save the `log4j2.properties` file. The changes are applied automatically.

You can edit the `outerbay.properties` file to control the number of log files.

To set the number of log files

1. Navigate to `config` under the home directory, which contains the `outerbay.properties` file. See Manage the home directory.

2. Create a backup copy of `outerbay.properties` that you can roll back to in case of any problems.

3. Open the `outerbay.properties` file with a text editor of your choice.

4. Set the `envBasedLogging` and `runBasedLogging` properties to control the number of logging files as follows:

   - To have only the log file `obt.log` , set both `envBasedLogging` and `runBasedLogging` to false.

   - To have the log file `<envName>.log`, set `envBasedLogging` to true.

   - To have the log file `<envName>_run_<runID>.log`, set `runBasedLogging` to true.

   - To have all three logs, set both `envBasedLogging` and `runBasedLogging` to true.

5. Save the `outerbay.properties` file. The changes are applied automatically.

1.3.2. Secure the Web Console

Web Console is a powerful browser based user interface to SDM, hence securing it is an important aspect. You can secure the communication with Web Console by configuring HTTPS, and restrict the access to Web Console functions by controlling the access at group level, enforcing strict password controls, and leverage existing enterprise level user management infrastructure by integrating it with LDAP server.

- Configure HTTPS
- Protocols and Ciphers
- User Management
- LDAP Integration
- Import a certificate
- SAMLv2 Integration
- Enhance password security
- Configure Knowledge Discovery (IDOL) security
- Restrict Web services for cloud deployments

1.3.2.1. Configure HTTPS

Before you start the configuration, make sure you have the following files ready:

- CA signed certificate file (Base64 encoded .pem or .cer)

- Private key (corresponding Base64 encoded private key)

Now that you have all the required files, perform the following steps to configure HTTPS:

1. The Web Console runs on an instance of Tomcat, which is located in `<install_dir>\obt\webconsole\apache-tomcat`. The SSL can be implemented in a similar way as other Tomcat instance. For detailed information, see Tomcat documentation and the Tomcat `server.xml` configuration file.

   > ⚠️ **Important**
   >
   > When configuring Tomcat for SSL you should
   >
   > - avoid using the APR implementation because it uses the OpenSSL engine by default.
   >
   > - use only the supported protocols and ciphers, see Protocols and Ciphers.

2. Stop the WebConsole.

3. Go to `<install_dir>\obt\webconsole\apache-tomcat\conf` and open `server.xml` in a text editor.

4. Add the following lines under the tags `Connector > SSLHostConfig`

   ```
   <Certificate
   certificateFile="<Certificate_File_Path>"
   certificateKeyFile="<Certificate_Key_File_Path>"
   certificateKeyPassword="<Certificate_Key_Password>"
   type="UNDEFINED" />
   ```

   > 📝 **Example**
   >
   > ```
   > <Connector
   > port="8443"
   > protocol="org.apache.coyote.http11.Http11NioProtocol"
   > maxThreads="150"
   > SSLEnabled="true"
   > scheme="https"
   > secure="true" >
   > <SSLHostConfig
   >  protocols="TLSv1.3,+TLSv1.2"
   >  ciphers="TLS_AES_128_GCM_SHA256,
   >     TLS_AES_256_GCM_SHA384,
   >     TLS_CHACHA20_POLY1305_SHA256,
   >     ECDHE-ECDSA-AES128-GCM-SHA256,
   >     ECDHE-RSA-AES128-GCM-SHA256,
   >     ECDHE-ECDSA-AES256-GCM-SHA384,
   >     ECDHE-RSA-AES256-GCM-SHA384,
   >     ECDHE-ECDSA-CHACHA20-POLY1305,
   >     ECDHE-RSA-CHACHA20-POLY1305,
   >     DHE-RSA-AES128-GCM-SHA256,
   >     DHE-RSA-AES256-GCM-SHA384,
   >     DHE-RSA-CHACHA20-POLY1305"
   > honorCipherOrder="true"
   > certificateVerification="none">
   > <Certificate
   >     certificateFile="C:/Certificates/Certificate.cer"
   >     certificateKeyFile="C:/Certificates/Certificate.key"
   >     certificateKeyPassword="P@ss1234"
   >     type="UNDEFINED" />
   > </SSLHostConfig>
   > </Connector>
   > ```

5. Save the file and start the WebConsole.

> **Note**
>
> - When you enable HTTPS, for Designer to work with Web Console (for remote or SAML deployment of cartridges/business flows), you need to add CA certificate or certificate chain to cacert of JRE's cacerts file on both the Designer machine and the Web Console machine. For the complete steps, see Import a certificate.
>
> - To enable SSL for WebConsole using self signed certificate, see Enable SSL for WebConsole Utility.

1.3.2.2. Protocols and Ciphers

Supported Protocols

- TLS 1.3
- TLS 1.2

Supported Ciphers

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- ECDHE-ECDSA-AES128-GCM-SHA256
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-CHACHA20-POLY1305
- ECDHE-RSA-CHACHA20-POLY1305
- DHE-RSA-AES128-GCM-SHA256
- DHE-RSA-AES256-GCM-SHA384
- DHE-RSA-CHACHA20-POLY1305

1.3.2.3. User Management

In most cases, more than one user needs to perform functions in SDM, but not necessary for every user to have the same privileges. For example, some users might only need to run jobs while others may need to deploy business flows, set parameter values, and run jobs. To meet this, the Web Console provides a user model that enables you to create multiple users, and grant or withhold the privileges. Such privileges are assigned at the group level to manage the access control in an optimized way. Most of the enterprises today manage the users through directory management software by using LDAP (for example, OpenLDAP or ActiveDirectory) or SAMLv2 (e.g. WSO2). The SDM provides a seamless integration with LDAP and SAMLv2 IdP or in other words LDAP or SAMLv2 IdP is used as a user store. The users or groups which belongs to only SDM (and not part of LDAP or SAMLv2 user store) are known as built-in users or built-in groups respectively.

1.3.2.4. LDAP Integration

Before integrating SDM with LDAP make sure that you have access control policies defined and the LDAP user store has been configured accordingly. Let us assume that the SDM users are classified into two categories as follows:

1. Administrators (who have access to all the functionality provided by SDM)

2. Users (who have restricted access to the functionality provided by SDM)

Based on the above categorization you can have the following two groups in LDAP.

1. **SDM_ADMIN**
2. **SDM_USERS**

Note that with such a distinction, we make sure that the users who are added to the group **SDM_USERS** (in LDAP) will have restricted access to the SDM functionality.

With such a distinction, we make sure that the users who are added to **SDM_USERS** will never get access to SDM functionality which is not supposed to be given access to other than few people of your organization.

On the LDAP side, we make sure that we add the following users to the **SDM_ADMIN** group:

- sdm.admin1

- sdm.admin2

Similarly, add the following users to **SDM_USERS** group:

- sdm.user1

- sdm.user2

With the above groups and users, our LDAP tree structure will appear as shown below:
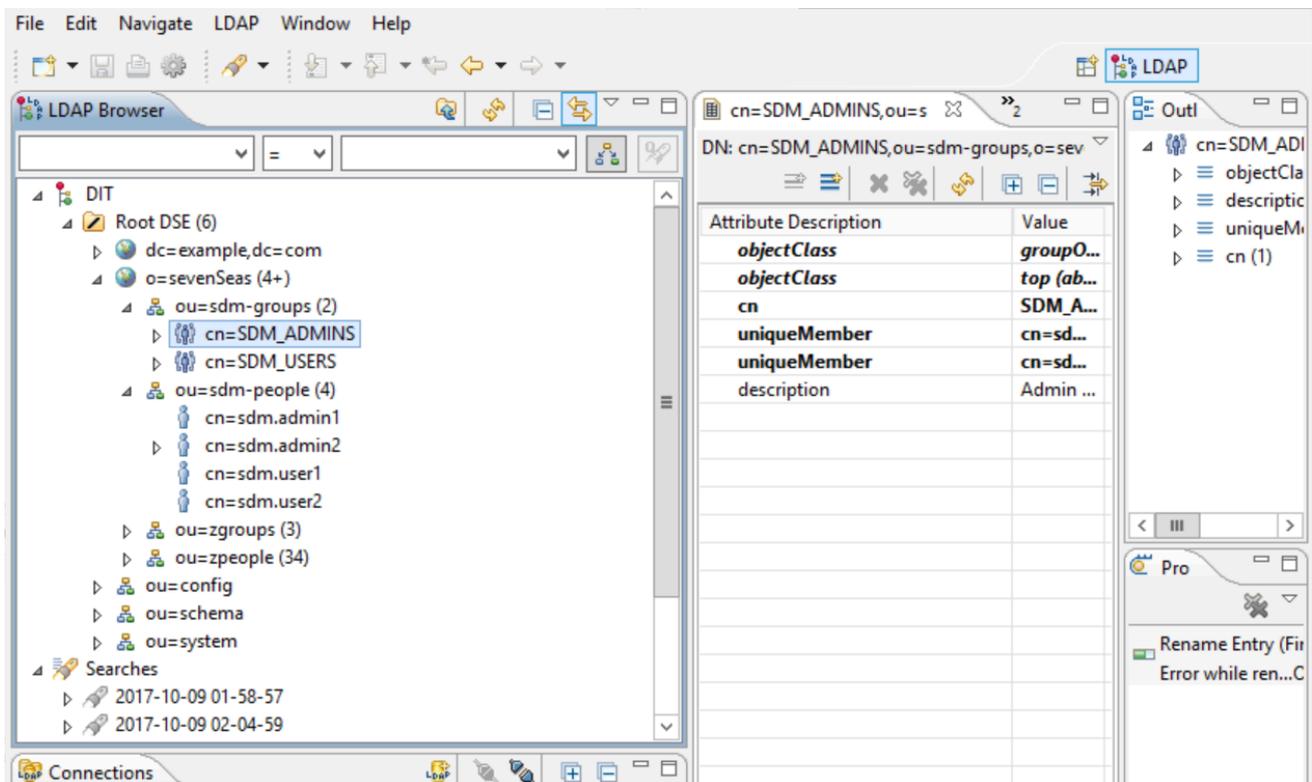
SDM Configuration for LDAP

If you connect to LDAP over SSL (that is, use ldaps:// in your URL), you need to make sure that you import root CA certificate as below.

> **Note**
>
> The instructions assumes the use of LDAP as a best practice for security. Also, you can implement LDAP without SSL if your environment permits and security is not a major concern.

1.3.2.5. Import a certificate

To import a certificate for the Web Console/Designer to use:

For an easy import of certificates SDM provides importCert utility.

`importCert.bat` (Windows) / `importCert.sh` (Linux).

> **Note**
>
> The following formats are supported for certificates: `.cer`, `.pfx`, and `.pem`.

1. Import the certificate using the command as below:

   `importCert.bat/.sh` *<Path to Certificate File> <Alias>*

   where *<Alias>* is optional.

2. You are now prompted to enter the password:

   For .cer and .pem

   - Enter keystore password

     (default password is `changeit`)

   For .pfx

   - Enter destination keystore password

     (default password is `changeit`)

   - Enter the source keystore password

     (password provided while creating certificate)

3. Type yes to successfully import the certificate to the keystore.

Update webConsole.properties

Now that you have done all of the preparation, you are ready to cut over to LDAP or LDAPS for access control:

1. Stop the Web Console. See Start and configure the Web Console.

2. In a text editor, open `webConsole.properties`, which is stored in the application data directory.

3. Uncomment the LDAP configuration and update the values for your environment. Note that `security.ldapServer` can be `ldap` or `ldaps`. For example:

> **Note**
>
> The LDAP settings that you need to include may vary depending upon your LDAP setup. Consult with your LDAP administrator to ensure you have the right settings specified in `webConsole.properties`.

```
security.ldapServer=ldap://ldap.sevenSeas.com:10389
security.ldapManagerDn=uid=admin,ou=system
security.ldapManagerPassword=manage
security.ldapSearchBase=ou=sdm-people,o=sevenSeas
security.ldapSearchFilter=(cn={0})
security.ldapRetrieveDatabaseRoles=true
security.ldapRetrieveGroupRoles=true
security.ldapGroupSearchBase=ou=sdm-groups,o=sevenSeas
security.ldapGroupSearchFilter=(uniquemember={0})
security.ldapAttribute.userRealName=cn
security.ldapAttribute.email=mail
security.ldapPasswordAttributeName=password
```

Example for Active Directory configuration:

```
security.ldapServer=ldap://sdmqa.internal.net:389
security.ldapManagerDn=cn=Administrator,cn=Users,dc=SDMQA,dc=com
security.ldapManagerPassword=mypassword
security.ldapSearchBase=cn=users,dc=SDMQA,dc=com
security.ldapSearchFilter=(sAMAccountName={0})
security.ldapRetrieveDatabaseRoles=true
security.ldapRetrieveGroupRoles=false
security.ldapGroupSearchBase=
security.ldapGroupSearchFilter=
security.ldapAttribute.userRealName=cn
security.ldapAttribute.email=mail
```

4. Start the Web Console. See Start and configure the Web Console.

Include the LDAP manager password

1. If you want to perform authenticated LDAP searches, you need to enter your LDAP manager password in `webConsole.properties` as follows:

   `security.ldapManagerPassword=password`

   > **Note**
   >
   > All **Basic Latin Unitt** ASCII special characters except a backslash (\) are supported for an LDAP manager password in the `webConsole.properties` file.

2. With `ldapManagerPassword` set, administrators are prompted during their next login to choose whether to encrypt the password in the `webConsole.properties` file.
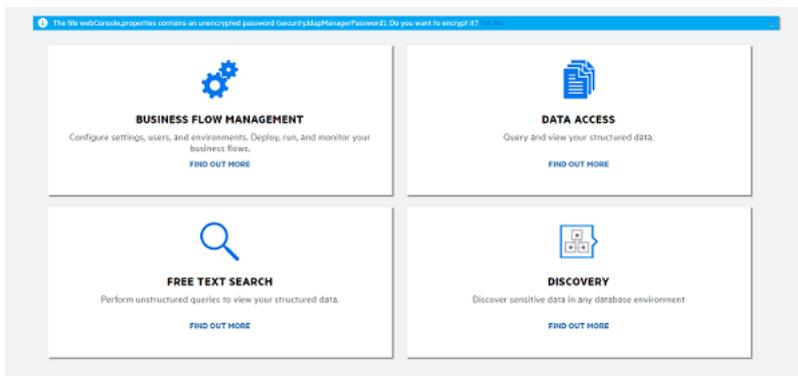
   > **Note**
   >
   > If you have multiple administrators, each one is prompted during the login.

   - If an administrator selects **Yes**, the `ldapManagerPassword` setting with the password in clear text is removed from `webConsole.properties` and the `ldapEncryptedManagerPassword` is appended to the end of the file:

     `security.ldapEncryptedManagerPassword=<encrypted value>`

     > **⚠ Caution**
     >
     > Encryption of the LDAP manager password in `webConsole.properties` is not intended as a substitute for file security. All of your Structured Data Manager configuration files, including `webConsole.properties`, should be stored in secured directories with restricted access.

   - If an administrator selects **No**, the `ldapManagerPassword` setting with the password in clear text remains intact and the administrator will receive no further prompts to encrypt it.

1.3.2.6. SAMLv2 Integration

Before integrating SDM with SAMLv2 make sure the user store, Identity Provider (IdP) and SDM Web Console as a Service Provider (SP) has been configured properly. For more information, see SDM Configuration for SAMLv2.

Let us assume that the SDM users are classified into two categories as follows:

1. Administrators (who have access to all the functionality provided by SDM)

2. Users (who have restricted access to the functionality provided by SDM)

Based on the above categorization you can have the following two Roles/Groups created in SAMLv2:

1. SDM_ADMIN
2. SDM_USERS

Groups with same name will need to be created in SDM Web Console as well, with various permission restricted on **SDM_USERS** group.

Note that with such a distinction, we make sure that the users who are added to the Role/Group **SDM_USERS** (in SAMLv2) will have restricted access to the SDM functionality.

With such a distinction, we make sure that the users who are added to **SDM_USERS** will never get access to SDM functionality which is not supposed to be given access to other than few people of your organization.

On the SAMLv2 IdP side, we make sure that we add the following users to the **SDM_ADMIN** Role/Group:

- sdm.admin1
- sdm.admin2

Similarly, add the following users to **SDM_USERS** Role/Group:

- sdm.user1
- sdm.user2

1.3.2.6.1. SDM Configuration for SAMLv2

Before you start with configuration, export the metadata files for both Identity Provider (IdP) and Service Provider (SP).

Export Service Provider (SP) Metadata

When you configure SAMLv2 for Structured Data Manager, you might need to provide SP metadata to your IdP.

Use the following sample SP metadata as a reference. Replace the placeholders with values that are specific to your SDM environment before you share the metadata with your IdP administrator.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor
        xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
  ID="${SP_ID}"
  entityID="${SP_ENTITY_ID}">

        <md:SPSSODescriptor
    AuthnRequestsSigned="true"
    WantAssertionsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

                <md:KeyDescriptor use="signing">
                        <ds:KeyInfo
                                xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                                <ds:X509Data>
                                        <ds:X509Certificate>
        ${SP_SIGNING_CERTIFICATE_BASE64}
        </ds:X509Certificate>
                                </ds:X509Data>
                        </ds:KeyInfo>
                </md:KeyDescriptor>
                <md:KeyDescriptor use="encryption">
                        <ds:KeyInfo
                                xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                                <ds:X509Data>
                                        <ds:X509Certificate>
        ${SP_ENCRYPTION_CERTIFICATE_BASE64}
        </ds:X509Certificate>
                                </ds:X509Data>
                        </ds:KeyInfo>
                </md:KeyDescriptor>

                <md:SingleLogoutService
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="${SP_SLO_URL}" />

                <md:SingleLogoutService
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
    Location="${SP_SLO_URL}" />

                <md:NameIDFormat>
    urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
    </md:NameIDFormat>
                <md:NameIDFormat>
    urn:oasis:names:tc:SAML:2.0:nameid-format:transient
    </md:NameIDFormat>
                <md:NameIDFormat>
    urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
    </md:NameIDFormat>
                <md:NameIDFormat>
    urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
    </md:NameIDFormat>
                <md:NameIDFormat>
    urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
    </md:NameIDFormat>

                <md:AssertionConsumerService
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
```

```
        Location="${SP_ACS_URL}"
        index="0"
        isDefault="true" />

                <md:AssertionConsumerService
        Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
        Location="${SP_ACS_URL}"
        index="1" />

        </md:SPSSODescriptor>
</md:EntityDescriptor>
```

Placeholder	Description
`${SP_ID}`	Unique identifier for the Service Provider
`${SP_ENTITY_ID}`	Entity ID of the SDM Service Provider
`${SP_SIGNING_CERTIFICATE_BASE64}`	Base64-encoded X.509 certificate used for signing SAML requests
`${SP_ENCRYPTION_CERTIFICATE_BASE64}`	Base64-encoded X.509 certificate used for assertion encryption
`${SP_SLO_URL}`	Single Logout (SLO) endpoint URL of SDM
`${SP_ACS_URL}`	Assertion Consumer Service (ACS) endpoint URL of SDM

> **Note**
> The SP metadata values might vary depending on your SAMLv2 setup. Consult your IdP administrator to ensure that you use the correct endpoints, certificates, and identifiers.

Now that you have done all of the preparation, you are ready to cut over to SAMLv2 for access control:

1. Stop the Web Console.
2. In a text editor, open `<install_dir>/config/webConsole.groovy`.
3. Uncomment the SAMLv2 configuration and update the values for your environment.

4. Set the following property `grails.plugin.springsecurity.saml.active` to true.

   > **Note**
   > The SAMLv2 settings that you need to include may vary depending upon your SAMLv2 setup. Consult your IdP administrator to ensure you have the right settings specified in webConsole.groovy. Example settings are present in the configuration file already.

5. Start the Web Console.

Now you should see the SAMLv2 login page instead of SDM login page.

1.3.2.6.2. Using built-in users when SAMLv2 is enabled

When SAMLv2 is enabled default login page takes you to SAMLv2 login page. In some cases, you may like to login with built-in user's credential, you can do so by accessing the URL http://<HOST>:<PORT>/WebConsole/login/builtIn.

Role and group mapping for SAMLv2 authentication

When you enable SAMLv2 authentication, the Identity Provider (IdP) typically sends user roles or group values as part of the SAML assertion (for example, `SDM_ADMIN,SDM_USERS`).

External group requirement

Any role or group value sent through SAML must exist as an external group in SDM before the user can be authorized.

Administrators can use the built-in user login to access SDM and perform this configuration.

Required configuration

1. Identify the role or group attribute that is configured in the IdP (for example, role, groups, or a custom attribute).

2. In SDM user management, create corresponding external groups that match the role values that are sent by the IdP.

3. Assign appropriate SDM permissions to each external group.

4. Ensure that the IdP sends the role or group attribute consistently for all users.

> **Note**
>
> If an IdP provided role or group value has not been created as an external group in SDM, the user may authenticate successfully but authorization to SDM functionality will fail.

1.3.2.6.3. Global and local logout

Global logout logs out current session from the IdP & all connected SP's, whereas local logout logs out the current session from specific SP only. By default SDM provides a single logout button with a global logout behavior. If you want to have both the options (global and local logout) you can do so with SDM Web Console. To do so, set the `grails.plugin.springsecurity.saml.custom.enableLocalLogout` property in `<install_dir>/config/webConsole.groovy` file to `true`.

> **Note**
>
> 1. Make sure to restart the Web Console (See Start the Web Console) for the changes to take effect.
>
> 2. After enabling Local Logout, the users will see two logout buttons in Web Console:
>
>   1. Local Logout
>   2. Global Logout

SAML support in the Designer

To enable SAML support in the Designer and deploy a business flow with SAML, see *Deploy a business flow with SAML* in *OpenText™ Structured Data Manager Developer's Guide*.

1.3.2.7. Enhance password security

To further increase security, you can place some additional requirements on your login and password behavior.

To set password security

1. Stop the Web Console. See Stop the Web Console

2. In a text editor, open `webConsole.properties`, which is stored in the home directory under `config`. See Manage the home directory.

3. Modify the following properties as desired to control login and password behavior:

Login and password properties

Property	Description
security.user.authMaxFailedAttempts	Defines the maximum number of failed login attempts allowed before the user's account is locked. The default value is 5.
security.password.minLength	Defines the minimum required length of a password. The default is 8.
security.password.mustHaveLetter	Indicates that the password must have at least one letter.
security.password.mustHaveDigit	Indicates that the password must have at least one digit.
security.password.ruleMessage	Is the message displayed to users when their new password is rejected for some reason. Typically this message outlines the password requirements.
security.password.mustHaveSpecialChar	Indicates that the password must have at least one special character out of !@#\$%&*_+=?~-

> **Note**
>
> Security policies can be added even before installation of repository, but if the `HOME` folder is changed during the installation of repository, then you must set the password security policies again in the `webConsole.properties` file which is stored in the `<OBT_HOME>/config` directory.

4. Start the Web Console. See Start the Web Console.

1.3.2.8. Configure Knowledge Discovery (IDOL) security

If you use Structured Data Manager to index structured records for Knowledge Discovery (IDOL), you should check your Knowledge Discovery (IDOL) settings in Web Console and choose appropriate security options for communicating with Knowledge Discovery (IDOL).

To review and change your Knowledge Discovery (IDOL) settings for the Web Console

1. From the Web Console home page, click **Settings**.
2. Click **Indexing Server** from the left navigation pane. The Indexing Server page displays.
3. Select **Knowledge Discovery (IDOL)** from the drop down list as **Indexing Server Type**.
4. Select the protocol to employ when communicating with the Knowledge Discovery (IDOL) server, HTTP or HTTPS.
   For more information about HTTPS and Knowledge Discovery (IDOL) See the *Knowledge Discovery Administration Guide*, "Set Up an SSL Connection".
5. If you have configured TEA encryption for Knowledge Discovery (IDOL), you can enable it here by checking **Enabled** and entering the key in the provided field.
   TEA encryption is configured in `AutonomyIDOLServer.cfg`. For example:

   ```
   ACIEncryption

   CommsEncryptionTEAKeys=1111,2222,3333,4444

   CommsAllowUnencrypted=false

   CommsEncryptionType=TEA
   ```

6. Optionally, you can change the Security by Database setting as well:

   - **Allow users with data access privileges to query any Knowledge Discovery (IDOL)database** means that any user who has the data access privilege granted to them can also query any Knowledge Discovery (IDOL) database. See the Web Console online help for more information on privileges.

   - **Grant Knowledge Discovery (IDOL) database query access** means that you must explicitly grant users query access to Knowledge Discovery (IDOL) databases. If you choose this option, when you select Apply, a new link appears called **Set security by database**. Following the link takes you to a page where you can manage query access by Knowledge Discovery (IDOL) database and user.

7. Click **Apply**.

1.3.2.9. Restrict Web services for cloud deployments

When deployed in the cloud, you should disable deployments and job runs from the Web Console:

Make the following changes in `outerbay.properties`:

- `allow.remoteDeployment=false`
- `allow.remoteRun=false`

By changing these values to false, the server will refuse to do deployments and start business flows using Web services.

1.3.3. Manage environments

An environment is a named deployment environment associated with a source (active) database. If you plan to perform database to database operations for the source database, the environment will also include a target database or location. From an environment, users with the necessary privileges can run business flows against the source database. You can create as many environments as you wish, thus supporting multiple source and target databases from a single installation of Structured Data Manager.

Before you can run business flows, you must create at least one environment into which you can deploy the business flows.

> **Tip**
>
> Step-by-step information for the Web Console user interface can be found in the Web Console help system.

- Choose environment type
- Use JDBC URLs for database connections
- Configure standard database to file environments for DB2
- Configure parameters
- Manage locations
- Refresh test environments

1.3.3.1. Choose environment type

In Web Console, you can have multiple, different combinations of data source and target. For example, you might have ten source databases from which to archive and five locations in which to store the archived data. Environments provide the mechanism for you to uniquely identify a specific combination of source and its potential targets. When you run jobs, you run them in a particular environment, that is, against a specific combination of source and target. Database to file archiving is enabled for each environment and you can optionally add database to database archiving to the environment.

Before you start, you must consider what type of environment best meets your business requirements and policies. Structured Data Manager offers two major types of environment:

Standard environments

In a standard environment, data is archived using an interface schema created in your source database, and native or JDBC drivers to move the data. Standard environments support all cartridge types, including database to database cartridges.

When to use

A standard environment is best when:

- The data source has no restrictions on the operations that you can perform, such as DELETE.
- You require database to database archiving.
- Performance is critical.

Non-intrusive environments

In a non-intrusive environment, data is archived without creating an interface schema in the source database. Non-intrusive environments use the corresponding JDBC driver for the source database. You can use the JDBC driver that comes with Structured Data Manager for natively supported platforms. If you are using a non-native database, you must supply your own JDBC driver.

> **Note**
>
> Non-intrusive environments only support database to file archiving. You cannot run database to database cartridges in a non-intrusive environment.

Non-intrusive environments support all of the following functionality:

- Data masking

- Uploading of extracted files to supported databases.

- Use of Column Inclusion, Name Override and Custom Properties.

- Delete operations, provided it is possible to perform delete operations against the data source. If you attempt to delete something when your source does not support deletes, you will receive an error message when running the delete operation. Delete-only cartridges are supported as well, but they are subject to these same limitations.

When to use

A non-intrusive environment is best when:

- Database to database archiving is not required.

- You must copy from read-only sources, especially in cases where data is associated with older technologies that might not support basic SQL statements such as DELETE, or when the database administrator or company policy prohibits creating objects in the production database.

- Performance is not a major issue. Standard environments may perform better than non-intrusive environments in some instances. If performance is a problem and the data source is not read-only, you may want to consider a standard environment.

- None of the following behaviors presents a problem for you:

  - Selection and data movement occur simultaneously. If you split an archive activity into separate selection and data movement segments, selection is automatically skipped.

- When deploying a database to file cartridge against a non-intrusive environment, no selection tables are created. Data movement is performed directly from the source environment.

- Because there is no interface schema, Eligibility Analytics are not supported. Also, Validation is not available. If you specify Validation in a database to file cartridge, it will be ignored when the cartridge is deployed.

- Reload and undo are not supported. However, you can upload extracted XML or CSV files. After extracting data from a non-intrusive environment, you can upload the files to any of the supported database platforms.

- Advanced Selection and chaining tables are not supported. If you specify Advanced Selection, cartridge deployment will fail.

- Custom selection is supported in non-intrusive environments; however, you must include a SELECT statement against the driving table (rather than an INSERT statement). The SELECT statement must include all columns from the driving table. If all columns are not selected in the driving table, a runtime error occurs when you attempt to deploy the cartridge. The error occurs because the custom SQL is not parsed, and it cannot be determined whether all columns are included at the time of deployment.

1.3.3.2. Use JDBC URLs for database connections

You can use a JDBC URL to connect to your databases.

JDBC URLs

To use a JDBC URL for your connections, you need to enter the following information in the appropriate format:

> **Note**
>
> You can only use one JDBC driver at a time. Therefore, the URL you use to invoke the driver must be unique.

Database	Expected URL
Oracle	jdbc:oracle:thin:@<Host>:<Port>:<OracleServiceName>
SQL Server	jdbc:sqlserver://<Host>:<Port>;InstanceName=<DatabaseName> or jdbc:sqlserver://<Host>:<Port>;InstanceName=<DatabaseName>;AuthenticationMethod=auto **Note** • Copy the `SQLJDBC4.JAR` driver from `/obt/lib` to `/obt/lib/jdbc`. • The `sqlserverToPostgresql.xml` file from `..\obt\foundation\components\config\AQSCache` location needs to be copied and named as `ansiToPostgresql.xml` if the source database is sqlserver and target database is Postgresql.
Sybase	jdbc:sybase:Tds:<Host>:<Port>/<DatabaseName>
DB2	jdbc:db2://<Host>:<Port>/<DatabaseName>
MySQL	jdbc:mysql://<Host>:<Port>/<DatabaseName> **Note** Add the `mysql-connector-java.jar` to `/obt/lib/jdbc`.

Parameter	Description
Host	The fully qualified host name or address for the connection.
Port	The port number for the connection.
OracleServiceName	The Oracle database service name.
DatabaseName	The name of the SQL Server, Sybase, or DB2 database.

Oracle example

`jdbc:oracle:thin:@localhost:1521:ORCL`

Oracle RAC (Real Application Clusters)

You can use database to database archiving in an Oracle RAC environment that has more than one server, or node, connected to a database on a shared file system. To ensure connectivity even in the event of a rollover:

- The repository must be installed either on a non-RAC server or using a JDBC URL connection.
- The active database must use a JDBC URL connection.
- For distributed archive, the archive database must use a JDBC URL connection.

Contact your database administrator for the correct JDBC URL that you should use. The JDBC URL for Oracle RAC should look similar to this:

Example

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=Node1)(PORT=1521))(ADDRESS=
(PROTOCOL=TCP)(HOST=Node2)(PORT=<port>))(LOAD_BALANCE=no)(FAILOVER=yes))(CONNECT_DATA=
(SERVICE_NAME=My_Database)))
```

1.3.3.3. Configure standard database to file environments for DB2

When you create a standard database to file environment with DB2 as the source, you must set two additional properties in `<install_dir>\obt\install\product_<user>.properties`:

- `source.interface.default.schema`

- `source.interface.default.storagelocation`

Properties file example

The following is an example of `product_<user>.properties`:

```
 #Product Information
encryption.key = hello
environment.id = DefaultEnvironment
environment.description = my db2 environment
ea.product.configuration = default
##ea.product.configuration = lite

#Source Database Information
source.rdbms.name = db2
##source.rdbms.name = db2

source.conn.dbserver = db2i97a
source.conn.host = linux-qacert-08
source.conn.port = 50000
#source.conn.conntype=url
#source.conn.url = jdbc:db2://linux-qacert-08:50000/db2i97a
source.dbadmin.username = db2i97a
source.dbadmin.password = all4one

#Interface User Information
#source.interface.dbname = db2i97a_SOURCE_INTF
source.interface.default.schema=${source.interface.default.schema}
source.interface.default.storagelocation=${source.interface.default.storagelocation}
##source.interface.dbname =
source.interface.owner.name = db2rt
source.interface.owner.password = welcome
##source.interface.owner.name = db2i97a
##source.interface.owner.password = all4one
```

1.3.3.4. Configure parameters

Before you run business flows, you should review and configure the parameters that govern the running of those business flows. SDM provides parameters at the following levels:

- At the database to database level, you configure parameters that apply to all database to database business flows and cartridges. For a listing of all of these parameters, see Configuration and runtime parameters.
- At the database to file level, you configure parameters that apply to all database to file business flows and cartridges. For a listing of all of these parameters, see Configuration and runtime parameters.
- At the business flow level, you configure the user-defined configuration parameters created in Designer, and those parameters apply to all business flows. For more information, see the *OpenText™ Structured Data Manager Developer's Guide.*
- At the cartridge level, you configure parameters that apply to the particular instance of the cartridge, and override the value of the database to database level or database to file level parameter. If the same cartridge belongs to more than one business flow, each instance of the cartridge can have different parameter values.

Choose a data movement method for database to database

One database to database parameter change that you might need to make for performance reasons is the Data movement method parameter in the Performance section. SDM supports the following data movement options for database to database:

- fully transactional
- table parallel

When using the fully transactional method, all data, including LONGs, are copied and deleted in the same transaction. The data is copied directly from the OLTP tables to the history tables, and the history data is always consistent.

The fully transactional method is best when consistent access to the archived data must be maintained.

When using the table parallel method, different tables are copied in different transactions in parallel, directly from the active database to the archive. Unlike the fully transactional method, the data is deleted in a separate transaction. Because the copy and delete are in separate transactions, there may temporarily be missing or duplicate data in the history tables before the entire process is complete.

> **Note**
>
> When using the table parallel method, the transparency layer should not be accessed until the data movement process is complete, as it may display errors or show inconsistent data.

The table parallel method is best when faster performance is required, and temporarily inconsistent historical data is not an issue.

To switch between data movement methods

1. Access the Parameters page as described in Choose a data movement method for database to database.
2. Click **Database to Database** from the left navigation pane.

The Parameters - Database to Database page displays.

1. Change the value of the Data movement method parameter in the Performance section.
2. Click **Apply**. The configuration parameters are saved.

Configure user-defined business flow parameters

In Designer, you can create configuration parameters that are used as part of a business flow that is deployed into a particular environment. These parameters appear on the business flow level of the Parameters - Business Flows page. If no user-defined configuration parameters were created for your business flows, this page is empty.

For more information on user-defined parameters, see the *OpenText™ Structured Data Manager Developers Guide*, or contact your archive developer.

Configure cartridge parameters

The cartridge level parameters apply to one instance of the selected cartridge, and are displayed on the Cartridge Configuration page. Except for the file naming parameters, all cartridge level parameters are a subset of the component level parameters listed in Configuration and runtime parameters.

Because file naming parameters create the naming convention for each individual database to file cartridge, they can only be selected on the cartridge parameter pages.

1.3.3.5. Manage locations

For business flows to run correctly, each environment must have a number of locations associated with it, for example, source and target database locations and credentials, and a file location. Locations are initially specified during installation and automatically loaded into the Web Console. You can then view and manage all of these locations from the Web Console.

> **Note**
>
> If passwords are changed for any of your schemas or databases, you must edit the location and change the password.

> **Tip**
>
> While you can update location credentials within Web Console, to ensure security, you cannot change the encryption key for an environment from the Web Console. To change the encryption key, use the Password Manager utility. For more information, see Change encryption keys and passwords.

Default locations

The following are the default locations in Web Console. You can modify and add to these locations.

Location name	Description
ARCHIVE_ACCESS_DB	Is the database location and credentials where the archive access schema resides. Only for database to database archiving.
DELETE_IDX_FS	Is the local file system for the delete index.
HISTORY_DB	Is the target (archive) database location and credentials. Only for database to database archiving.
HISTRELOC_DB	Is the database location and credentials where the relocation schema resides. Only for database to database archiving.
LOCAL_ARCHIVE_FS	Is the path where database to file stores archived XML or CSV files.
LOCAL_TEMP_FS	Is the path of the temporary file space.
OBTINTF_DB	Is the database location and credentials where the interface schema resides.
OLTP_DB	Is the source (active) database location and credentials.

File archive settings

In **Settings > File Archive**, you will find the following file paths defined:

- **Archive Path** is the root path for your archive file storage. The path entered here can be referenced as `${archivePath}` elsewhere in the Web Console. For example, when defining your temporary directory, you might enter something like `${archivePath}/temp/` to indicate that you want to store temporary files in a subdirectory called `temp` inside of your root

archive path. If you later decide to change the archive path, you do not need to update your temporary directory specification. It will update automatically because you used the variable for your archive path.

- **Default archive directory** is the location where database-to-file archive files are stored. The default value is `${archivePath}/archivedata/${environment}/`, which means that the path is the root archive path plus `archivedata` plus the current environment name.
- **Default temporary directory** is the location where Web Console stores temporary files. The default value is `${archivePath}/temp/${environment}/`, which means that the path is the root archive path plus `temp` plus the current environment name.
- **Default delete index directory** is the location where indexes are stored. The default value is `${archivePath}/delete_idx/${environment}/`, which means that the path is the root archive path plus `delete_idx` plus the current environment name.

Change locations and update passwords

If you need to change a location, such as the archive file location, or update a location's credentials, for example, the source database's password, you can do so from the Web Console.

To change a location or update passwords

1. Navigate to **Environment Management > Environment > Active Environment > Locations** in the Web Console.

2. Select a location.

3. Change the location properties as appropriate.

   File system locations, such as LOCAL_TEMP_FS and LOCAL_ARCHIVE_FS, are typically expressed with variables:

   - `${environment}` represents the name of the current environment.

   - `${archivePath}` represents the archive path. You can change the archive path from **Settings > File Archive**. See Change the archive path.

5. Click **Apply**.

6. If the source database password for an environment changed, click **Environment Management > Environment > Admin Credentials** and update the password.

7. For database to database with a distributed instance, if you changed the interface schema or database password:

   1. Log in to the active database as the interface user, for example, OBT_IF.

   2. Drop the database link that points to the history relocation schema or database. This link was created when you deployed the business flow.

   3. Re-create the database link with the same name.

   4. Stop and start the Web Console.

8. For database to database with a distributed instance, if you changed the OLTP history relocation schema or database password:

   1. Log in to the history database as the relocation user, for example, OBT_RELOC.

   2. Drop the database link that points to the OLTP interface schema or database. This link was created when you deployed the business flow.

   3. Re-create the database link with the same name.

   4. Stop and start the Web Console.

> **Note**
>
> You can also change passwords using the password manager utility from the command line. For more information, see Update database location passwords.

Change the archive path

You can change the root archive path of Web Console and any locations based upon it (`${archivePath}`) will change, too.

To change the archive path

1. Navigate to **Settings > File Archive** in the Web Console.
2. Update the Archive Path. As soon as you make a change to this field, a radio button prompts you to choose between:
   - **Change for future archives** means you only want the path change to apply to future archiving operations. Existing archive files will remain in their current location.
   - **Move the data and apply for future archives** means you want to both move existing archive files to the new location and apply the path change to future archiving operations.

   > **Note**
   >
   > If you choose Move the data and apply for future archives, any AQS collections that reference the old archive path will be automatically updated for you. See Use AQS to access database to file output for more information about collections.

3. Change other settings as desired.
4. Click **Apply**.

Add locations

You may wish to create additional locations archiving. For example, in some cases you may want to archive files in a cloud location, such as Cloud Object Storage, and, in other cases, you may wish to archive to a backend located on your network.

Database to file locations

Available location types for database to file are:

- Amazon-S3
- Filesystem
- HDFS (Hadoop Filesystem)

  > **Note**
  >
  > Prerequisites for integrating Structured Data Manager and Hadoop. See Hadoop integration

- SDM Gateway, see Perform multiple data movements
- SSH
- Content Manager

  > **Note**
  >
  > Prerequisites for integrating Structured Data Manager and Content Manager. See Content Manager integration

Data access locations

Available location types for data access cartridges to query against are:

- AQS (Data Access Only)
- Vertica (Data Access Only)

Database to database locations

Available location types for database to database locations are:

- DB2
- JDBC URL
- Oracle
- PostgreSQL
- SQL Server
- Sybase

Parameterize locations

Within locations, you can make reference to business flow parameters, which allow you to dynamically set the path. For example, suppose that you want to include the runtime parameter values and the date in the path for your database to file extracts. You could enter the following for the **Directory** property of your file system location:

```
C:\myDirectory\${OA_CUST_ID}\${OA_ORD_YEAR}\csv\${new Date().format("dd-MM-yyyy")}
```

In this example, the parameters are indicated by `${}`.

- `OA_CUST_ID` and `OA_ORD_YEAR` are runtime parameters used by the business flow you are running. `OA_` is a prefix added to each parameter and the rest is the name given to the parameter when it was defined in Designer.

- `${new Date().format("dd-MM-yyyy")}` is a valid Groovy expression resulting into current date in turn provides you a way to organize your files per day.

This type of parameterization can be particularly useful when you want to separate your output files dynamically based on some set of parameters used during the business flow run.

> **Tip**
>
> You can reference any parameter that is referenceable from Groovy in the location definition as well. See the *OpenText™ Structured Data Manager Developer's Guide* for more information about parameters you can reference from Groovy.

1.3.3.6. Refresh test environments

If you are maintaining a test environment in which you design and test your models, cartridges, and business flows, you may want to periodically refresh it from your production environment. For example, if the production schema has changed or been upgraded in some way, you need to reflect those changes in your test environment. You can duplicate changes to the production environment in your test environment without reinstalling Structured Data Manager.

Prerequisites

For this process to work, your environments must meet the following criteria:

- Both environments must use the same version and configuration of Structured Data Manager.
- Both environments must use the same version and configuration of Oracle. This procedure only applies to Oracle databases.

To refresh your test environment

1. In the test environment, clone the following databases from the production environment:

    - active (source) databases

    - repository databases

    - target databases (for the database to database distributed instance configuration only)

2. In the cloned databases, update any database links to use the test environment's host names, port numbers, and so on. Otherwise, the database links will continue to resolve to the production databases rather than the test databases.

3. Manually copy the product tree and the OBT_HOME directory. The OBT_HOME directory is defined in *<install_dir>*`/obt/config/obt.env.`

4. In the replicated location, edit the *<install_dir>*`/obt/config/obt.env` to reset the OBT_HOME for the replicated environment, and make a copy of `obt.env` as `env2.obt.env.`

5. Check that the `OBT_HOME/config/directories.properties` file has the correct settings for the replicated environment.

6. Navigate to the following directory:

    `<install_directory>\obt\foundation\components\replicate`

    where, <install_directory> is the location where you installed Structured Data Manager.

7. Create a text file for the environment properties. The default name for this file is `replicate.properties`, which is defined in `replicate_environment.xml` as `<property name="properties.file" value="replicate.properties"/>`.

    The file must contain the following properties:

Property	Description
env1.repository.password	The repository password for the original environment.
env1.encryption.key	The encryption key for the original environment.
env2.encryption.key	The encryption key to be used for the duplicate environment.
env2.product.path	The path to be used fro the duplicate environment.
env2.source.dbadmin.password	The database administrator password for the source database.
env2.source.dbadmin.userid	The database administrator username for the source database.
env2.hist.dbadmin.password	The database administrator password for the target database. (distributed instance only)
env2.hist.dbadmin.userid	The database administrator username for the target database.
env2.repository.password	The repository password for the duplicate environment.
env2.interface.password	The password for the interface user.
env2.transparency.layer.password	The password for the OBT_BUILD_AA user.
env2.transparency.layer.location	The location of the OBT_BUILD_AA user. Valid values are source or target.
env2.relocation.interface.password	The password for the OBT_RELOC user.
env2.source.password	The password for the interface user (for example, obt_if) on the source database.
env2.source.host	The host name for the source database.
env2.source.server	The source database name.
env2.source.port	The port number for the source database (for example, 1522).

Property	Description
env2.repo.dbadmin.userid	The database administrator username for the repository database.
env2.repo.dbadmin.password	The database administrator password for the repository database.
env2.repo.host	The host name for the repository database.
env2.repo.server	The repository database name.
env2.repo.port	The port number for the repository database (for example, 1522).
env2.environment	The new environment name of the test if it is different from the production environment.
env2.hist.host	The host name for the archive (target) database.
env2.hist.port	The port number for the archive database (for example, 1522).
env2.hist.server	The archive database name.
env2.hist.password	The password for the relocation user (for example, obt_reloc) on the archive database.
env2.hist.schema.password	The password for the history schema.
env2.aa.schema.password	The password for the archive access schema.
replace.dblink	Whether to replace database links (true or false).
env2.oltp.to.hist	If you set `replace.dblink` to true, the new name of the database link from the active database to the archive database, for example, OBT_OLTP_TO_HIST2.
env2.hist.to.oltp	If you set `replace.dblink` to true, the new name of the database link from the archive database to the active database, for example, OBT_HIST_TO_OLTP2.

Property	Description
env2.aa.to.remote	If you set `replace.dblink` to true, the new name of the database link from the archive access schema, for example, OBT_AA_TO_REMOTEDB2.
environment.id	The name of the old environment from which this procedure is refreshed.
configuration	The environment's configuration: ○ OA means the environment has only database to file. ○ LA means the environment has database to file and database to database.
configuration.la	If the environment includes database to database, then this property indicates: ○ SI for single instance, where source and target are in the same database ○ DI for distributed instance, where target is in a separate database)
recreate.views	If you set to true, it recreates the views in the archive access schemas, replacing the database link name with the new name in the replicated database environment. This is only relevant for a distributed instance configurations, and might be very time consuming. If set to false, it will not recreate these views.
replace.db_link	If you set to true, insert a new database link name in the following properties to drop and recreate the database links with the changed database domain: ○ env2.oltp.to.hist ○ env2.hist.to.oltp ○ env2.aa.to.remote **Note** The DBLINK name should be the same as the global name of the target if the Oracle GLOBAL_NAMES property is set to true. Furthermore, suffixes on the global name will be applied to DBLINK, which can cause issues on the target database if it has GLOBAL_NAMES set to true and a different suffix.
repository.username	The repository user name, for example, obt_rep.

Database to file environment example

```
 env1.repository.password=obt_rep2
env1.encryption.key=q2

env2.encryption.key=q2
env2.product.path=E:\\program files\\SDM760\\obt
env2.source.dbadmin.password=manager
env2.source.dbadmin.userid=system
env2.repository.password=obt_rep2
env2.interface.password=obt_if2
env2.source.password=obt_if2
env2.source.host=localhost
env2.source.server=orcl2
env2.source.port=1522
env2.repo.host=localhost
env2.repo.server=orcl2
env2.repo.port=1522

environment.id=OA
configuration=OA
configuration.la=DI
repository.username=obt_rep
```

Database to database environment example

```
env1.repository.password=obt_rep2
env1.encryption.key=q2

env2.encryption.key=q2
env2.product.path=E:\\program files\\SDM760\\obt
env2.source.dbadmin.password=manager
env2.source.dbadmin.userid=system
env2.hist.dbadmin.password=manager
env2.hist.dbadmin.userid=system
env2.repository.password=obt_rep2
env2.interface.password=obt_if2
env2.transparency.layer.password=obt_build_aa2
env2.transparency.layer.location=source
env2.relocation.interface.password=obt_reloc2
env2.source.password=obt_if2
env2.source.host=localhost
env2.source.server=orcl2
env2.source.port=1522
env2.repo.host=localhost
env2.repo.server=orcl2
env2.repo.port=1522
env2.hist.host=localhost
env2.hist.port=1522
env2.hist.server=orcl3
env2.hist.password=obt_reloc2
env2.hist.schema.password=target2
env2.aa.schema.password=aa2

replace.dblink=true
env2.oltp.to.hist=OBT_OLTP_TO_HIST2
env2.hist.to.oltp=OBT_HIST_TO_OLTP2
env2.aa.to.remote=OBT_AA_TO_REMOTEDB2
environment.id=LADI
configuration=LA
configuration.la=DI
repository.username=obt_rep
```

8. Run the appropriate script for your operating system:

Platform	Syntax
UNIX	`<install_dir>/obt/bin/replicate_environment.sh`
Windows	`<install_dir>\obt\bin\replicate_environment.bat`

> **Note**
>
> Remotely copying the production environment to another server on UNIX is not supported.

9. To replicate more than one environment from the same production instance, repeat Step 3 through Step 8 with the following modifications:

   - The repository password and encryption key of the first environment should be changed to that of the new environment because they will have already been updated in the first run of `replicate_environment.bat`.

   - If the database links have to be renamed, set `replace.dblink` to `true` and provide the new link names.

10. Update the `sqltuning.properties` file, which can be found changing references to the old environment name to the new environment name.

11. Restart the Web Console.

Rename environments

After you refresh the test environment, you may want to change the environment name.

To change the name of an existing environment

```
rename_environment.bat/sh -e [old_env_name]
-n [new_env_name]
```

To change the environment description

```
rename_environment.bat/sh -e [old_env_name]
-n [new_env_name] -d [new_env_description]
```

1.3.4. Manage business flows

Before you run business flows, you should perform some checks; for example, confirming parameter settings. Beyond that, certain types of business flows require that you perform preparatory tasks before running them.

After you run your business flows, you need to access the archived data or possibly recover from error conditions.

> **Tip**
>
> Step-by-step information for the Web Console user interface can be found in the Web Console online help system.

- Deploy business flows
- Prepare to run business flows
- Recover or cancel business flows and jobs
- Access archived data

1.3.4.1. Deploy business flows

Business flows contain the activities that actually perform the movement, copying, and reloading of data, as well as other business logic (Groovy scripts). In order to run a business flow, you must first deploy and configure it in an environment. You can deploy business flows either from Designer or from the Web Console.

To deploy business flows using the Web Console , they must first be generated by you or someone else in , and the resulting deployment files placed in a location accessible to the Web Console .

> **Note**
> - Business flows containing 2-tier configuration must be deployed separately from business flows containing 3-tier configuration.
> - Deployment of Database to Database business flow is a prerequisite for the deployment of a Database to File business flow in the 3-tier mode.
> - It is recommended not to select both the archive and undo business flows simultaneously for deployment. Always deploy the archive business flow before proceeding with the undo business flow.

Generated business flows are zip files with a `.busflow` extension. By default, business flows are stored in one of the following locations, but you can import `.busflow` files from other locations as well:

Location	Description
`businessflow` in home directory	Business flows stored in the main business flow directory can be used by all users.
`businessflow/<env_name>` in home directory	Business flows stored in an environment directory can only be accessed by users who are assigned to that environment.

For more information about the home directory, see [Manage the home directory](#).

The deployed business flows are stored in the following directory under the home directory:

`/artifacts/<environment_name>/businessflow/<busflow_name>/`

Parameter	Description
environment_name	The name of the environment the business flow was deployed into.
busflow_name	The name of the business flow.

If you need to modify your business flow and its components to account for changes to the database, or for business or legal requirements, then you must modify the business flow or its components in Designer and redeploy it.

As with a new deployment, if you plan to redeploy business flows using the Web Console , they must first be generated by you or someone else in Designer, and the resulting deployment files placed in a location accessible to the Web Console .

The redeployed business flows are stored in the same location as when they were originally deployed.

When you no longer need a particular business flow or data access cartridge, you can use the Web Console to remove (uninstall) it.

> ⚠ **Caution**
> You should never remove the deployment directory or its contents manually from the file system. It must be properly uninstalled from the Web Console . Otherwise, you will get unexpected results when you try to redeploy, run, or uninstall the business flow.

See the Web Console online help and [Deploy and run business flows](#) for more information about deploying business flows in the Web Console.

1.3.4.2. Prepare to run business flows

Some business flows, such as those that reload data, require some preparation before running.

Prepare to reload archived data

Before running a reload from database business flow, you need to ensure that the same indexes were created on the archive database that existed on the active database. Otherwise, the performance of the reload will be adversely affected.

If you ran your archive jobs with the default parameter settings for creating indexes on history tables, then the indexes were created at that time and you can proceed with your reload business flow without taking any further action. If the parameter was set otherwise and indexes were not created during the archive business flow, you can create the indexes manually yourself. You can also rerun the archive business flow with a parameter value that causes to create the indexes on the archive database.

If the archiving business flow was previously run with the Type of indexes to create on History parameter set to Operate on all indexes, the indexes should have been created at that time and you need not perform this procedure.

To create the indexes on the history tables using

1. Navigate to **Business Flow Management** > **Tasks** > **System Parameters** > **Database to Database** in the Web Console .

2. Set the value of the **Type of indexes to create on History** parameter to **Operate on all indexes**.

3. Rerun the business flow using parameter settings that will not archive any data. For example, you could select a data range that does not match your retention policies.

   The indexes are automatically created on the archive database, and you can now proceed to reload your archived data.

Prepare to upload archived data

Before you upload your archived data, you should consider the following:

- See Mapping schemas by cartridge if you want to map the archived data into a schema name that is different from the source.

- The upload user must have create table, drop table, create index, and drop index privileges on the target database where the data is being uploaded.

- If you are planning to upload from an archive file generated against Oracle to SQL Server, you should explicitly set the precision and scale on numeric columns, where appropriate, in the source database. Otherwise, if SQL Server has more restrictive precision and scale settings than Oracle, the upload may fail.

- The default batch size for upload is 1000 rows. will store that many rows per worker prior to inserting them into the database. For example, if the number of workers is set to 5, then 5 threads will be running with each holding 1000 rows in memory before inserting into the target database. If you need to adjust the batch size, change `upload.batch.size` in `outerbay.properties` in `config` under the home directory.

1.3.4.3. Recover or cancel business flows and jobs

In general, it is best practice to never leave a business flow or job in one of the following states:

- Failed

- Suspended

- Not Responding

Otherwise, you may get unexpected results when you attempt to run the same or other, related business flows, or re-deploy the business flow.

> **Tip**
>
> To check the status of a business flow or job, got to **Business Flow Management > Tasks > Monitoring > Overview** in the Web Console.

If a business flow does not complete successfully, you can use the to:

- **Recover it.** Recovering the job or business flow reruns it using the same parameters as the first run. If possible, recovering is the best option.

- **Cancel it.** Cancelling the job or business flow should be your option of last resort. If you cannot recover for some reason and you need to redeploy the business flow, you can cancel the business flow. Cancelling updates the status of the business flow or job, but does not clean up any of the selection tables nor does it remove partially copied, deleted, or masked data. You may end up with unnecessary information in your database, which can never be used after canceling the job, and it will continue to occupy space. Consequently, cancelling a job can lead to inconsistent results.

  > **Note**
  >
  > If you cancel an upload business flow, it can result in indexes not being created on the target database, which can lead to performance problems.

  > **Important**
  >
  > If a business flow containing an advanced selection database to file cartridge does not complete successfully, see the *Troubleshooting Guide* for further information before rerunning the business flow.

1.3.4.4. Access archived data

After you archive your inactive data to an archive database or file, you still need to access it. From the Web Console, you can perform a variety of operations to facilitate access to the archived data:

- For database to database archives, you may enable transparency to allow users to see the archived data as though it were still in the active database Web Console.

- For database to file archives, you may relocate, copy, or delete the archived data from the Web Console.

Create database to database archive access transparency

Data transparency refers to the ability of users to see archived data as though it were still in the active database. If you implement data transparency, both active and archive data can be accessed through the current user interface. All data relationships necessary to maintain data and application integrity are retained.

After archiving your data, you can run the Create Archive Access job to enable transparency, which allows your users to view your archived data directly from the application.

> **Note**
>
> Contact your OpenText sales representative for information about licensing requirements for data transparency.

The Create Archive Access job creates or recreates objects within the Archive Access schemas or databases. The running time of the script depends on the number of cartridges included in your business flows, the number of managed tables, and the number of views, stored procedures, and packages that are dependent on those objects. To customize this job, see the *OpenText™ Structured Data Manager API Reference Guide*.

You should run the Create Archive Access job whenever any of the following occur:

- Changes are made to tables, views, stored procedures, or packages in the active database.

- New cartridges are installed.

- Cartridges are upgraded.

- New patches are installed on the active database.

- Any of the ArchiveAccessConfiguration groovy scripts are run.

See the *OpenText™ Structured Data Manager API Reference Guide.*

> **Note**
>
> The Archive Access schema is unstable while the Create Archive Access job is running. To avoid errors, wait until after the job has completed successfully before accessing the archive.

1. Navigate to **Business Flow Management** in the .

2. Confirm that the configuration mode parameter is properly set by navigating to **Tasks > Monitoring > System Parameters > Database to Database**.

   See Choosing a data movement method for database to database.

1. Click **Launch > Business Flow**.

2. Click **Create Archive Access**. The job launch page for the Create Archive Access job displays.

3. Click **Run** to run the job immediately, or **Schedule** to schedule the job for a later time.

4. Click **Confirm** to confirm the job. When the job completes successfully, you should be able to access the access the archived data from your application.

5. Test accessing some of the archived data from your application to ensure that the archive access is working.

> **Note**
>
> If you have uploaded the same data to the same database, you can have duplicate data because the upload process only checks for duplicate data when uploading lookup and MTU (Multiple Table Use) data. (It checks the target database for duplicate data.) To prevent duplicate data, the reload process checks for duplicate data based on the data movement key of the table. If the table does not have a data movement key or the data movement key is ROWID, reload may result in duplicate rows.

Relocate database to file archive data

After you archive from database to file, you use **Business Flow Management > Tasks > Launch > Relocate** to relocate or copy the archived data to a variety of locations:

- source database from which it was archived

- another database

- another file location

You can also delete the archived data from a backend.

For more information about relocating files, see the online help.

Reload and upload

When uploading or reloading the archived data, the rows are inserted into the database in batches. The size of these batches of inserts is set by the `upload.batch.size` property in `outerbay.properties`, which is located in `config` under the home directory. See [Managing the home directory](#).

A database commit can be performed after each batch of inserts from the file. Performing the database commit at the batch level ensures that the size of the rollback segment remains smaller, but referential integrity cannot be guaranteed until reload completes and all of the data is committed.

For all cases of upload (XML and CSV), only batch commits are performed.

For reload from an XML file, only file group commits are performed. That is, after inserting from an entire group of files, a database commit is performed.

For reload from a CSV file, you can control when the database commit is done. Use the following setting in `<install_dir>\obt\config\outerbay.properties` to control database commits for reloading from a CSV file:

- `reload.batch.commit=true` indicates batch level database commits. This setting has the benefit of smaller rollback segments, but it can lead to referential integrity issues until reload completes and all of the data is committed for every table.

- `reload.batch.commit=false` is the default setting and indicates file group level database commits. It can lead to very large rollback segments and, if the reload should fail, it will restart for the entire file group.

Access the archive data file

To access contents of your archive data files, you use the Archive Query Server. See [Use AQS to access database to file output](#).

Import the archive file set

In case you have archived some data through another instance of or prior version of SDM and you would like to have the same as part of the current instance of SDM, then you can use import archive set feature of SDM.

> **Note**
>
> Before starting this procedure, you must place the files to be imported in a location accessible to this instance of SDM. For example, you might place the files in folder `C:\MyArchiveSets`, and create a location called `MyArchiveSets` as described in [Manage locations](#).

To import an archive file set from another Structured Data Manager to register the archive file set and optionally relocate it to a specific target location

1. Navigate to **Business Flow Management > Tasks > Launch > Import** in the Web Console.

2. Select the **Source Location** where you have placed the archive file set(s) to be imported, for example, `MyArchiveSets`. List of archive file set(s) is displayed.

3. Select the archive file set to be imported.

4. If you want to relocate the archive set during import process, then select a **Target Location** (for example OBT_IF).

5. If you want to register the archive set and relocate to a target location later then specify Target Location as [None].

6. Click **OK**.

The import job is executed. Once the execution is complete a status message is displayed.

If you have just registered the archive set as per Step 5 above, then follow the relocate process described in the section Relocate database to file archive data.

For more detailed information on the imported jobs already executed, navigate to **Monitoring > History**. Click the job to display the status, run ID, name, start time, end time, sequence, and more.

1.3.5. Run business flows and jobs from the command line

To provide additional flexibility, Structured Data Manager enables you to run business flows and jobs from the command line as well as from the Web Console. This capability is useful for environments where you cannot easily access the Web Console or need to script the running of business flows or jobs.

Before you begin

Before you run business flows, you should perform some checks; for example, confirming parameter settings. Beyond that, certain types of business flows require that you perform preparatory tasks before running them. In particular, business flows that involve reloading data or bulk archiving require some preparation before running:

- The preparatory tasks for running business flows from the Web Console also apply to running on the command line. See Manage business flows.

- To run business flows or jobs from the command line, you may need some or all of the following information, which you can obtain from reviewing the deployed business flows in the Web Console:

    - Environment name

    - Environment ID

    - Business flow name

    - Job parameters

    - Job name

    - Job run ID

    - Destination location

Run business flows

To run business flows or jobs from the command line

1. Make sure that you have performed the relevant tasks from Before you begin.

2. Open a command window.

3. Navigate to the following directory:

   `<install_directory>/obt/bin`

   where, <install_directory> is the location where you installed the software.

1. Run the business flow using the appropriate syntax:

Platform	Syntax
UNIX	./launch_businessflow.sh [-r] -e <environment_name> -j <business_flow_name> <parameters> -s cli
Windows	launch_businessflow.bat [-r] -e <environment_name> -j <business_flow_name> "<parameters>" -s cli

Parameter	Description
-r	The recover flag, -r, allows you to recover jobs and business flows. See Recovering business flows. Although the recover flag, -r, is optional, OpenText recommends that you use it whenever you run a job or business flow.
environment_name	The name of the environment. This is optional if you only have one environment.
business_flow_name	The name of the business flow.
parameters	The parameters as a space-separated list with each entry in the form <name>=<value>. For Windows, each individual entry must be enclosed by quotation marks. The parameter values for all cartridges included in the business flow are required. For a date or time policy parameter, use the following format: [YY]YY.MM.DD[HH24:MI:[SS[.SSS]]]
-s	Specifies the source from which the business flow is executed, that is `cli`.

> **Tip**
>
> To verify the usage of the job from the command line, enter `launch_businessflow.bat -h`.
> The usage is displayed.
> To verify the parameters for the job, enter `launch_businessflow.bat -j <business_flow_name> -e <environment_name> -s cli -h`.
> After you enter the encryption key, the parameters are displayed.

Windows example

```
launch_businessflow.bat -e Oracle_env -j orders_flow_d2d -r "Min_Months_to_Retain=24" -s cli
```

> **Tip**
>
> If the parameter value is left blank, and a default value was defined in Designer, the default value is used. If there is no default value, NULL is used.

> **Note**
>
> Running an undo business flow requires the Run ID for the business flow you want to undo. The Run ID for the business flow is displayed on the Run Summary page.

2. Enter the encryption key at the prompt. The encryption key is case sensitive.
   The job runs, and displays a success or failure message.
3. Optionally, launch the Web Console to monitor the progress of the job.

Recover and cancel business flows and jobs

Recover business flows

Using the recover flag, `-r`, allows you to rerun a business flow using the same parameters as the first time it was run. You can also use the -r flag to continue the business flow after a scheduled pause activity.

> **Note**
>
> To recover a job, the job parameters must be exactly the same as the last time you ran it. If there are any differences in the syntax, even an extra space, a new job is launched instead of the previous job being recovered.

1. Make sure that you have performed the relevant tasks from Before you begin.
2. Open a command window.
3. Navigate to the following directory:
   `<install_directory>/obt/bin`
   where, <install_directory> is the location where you installed the software.
4. Enter the same job name and parameters as the job you want to recover.
   Windows example
   If you ran the business flow example given in Run business flows, you would run:
5. `launch_businessflow.bat -e Oracle_env -j orders_flow_d2d -r "Min_Months_to_Retain=24" -s cli`

Recover jobs using the recovery script

Using the recovery script to recover jobs enables you to select the specific job you want to recover.

To recover a job using the recovery script

1. Make sure that you have performed the relevant tasks from Before you begin.
2. Locate the Group ID of the failed job you want to recover.
   The Group ID is displayed on the Web Console History page.
3. Open a command window.
4. Navigate to the following directory:
   `<install_directory>/obt/bin`
   where, <install_directory> is the location where you installed the software.
5. Run the appropriate recovery script for the job you want to recover:

Job	Platform	Syntax
Any business flow or database to file job	UNIX	./recover_job.sh -e <environment_name> -g <group_ID>
	Windows	recover_job.bat -e <environment_name> -g <group_ID>
The Create Archive Access job	UNIX	./recover_create_aa.sh -e <environment_name> -g <group_ID>
	Windows	recover_create_aa.bat -e <en

Parameter	Description
environment_name	The name of the environment. This is optional if you only have one environment.
group_ID	The Group ID of the failed job as it appears in the Web Console.

Windows example

```
recover_job.bat -e "Oracle_env" -g "3"
```

6. Enter the encryption key at the prompt. The encryption key is case sensitive.
   The job runs, and displays a success or failure message.

7. Optionally, launch the Web Console to monitor the progress of the job

Cancel jobs from the command line

Cancelling jobs from the command line requires the `cancelJob` API Groovy script. The script allows you to update the status of a job to Cancelled. However, you should only cancel jobs in extreme cases because using the cancelJob API does not restore the state of the database to beginning of a failed job. You will have to manually restore the state of the database.

To cancel a job

1. Locate the Group ID of the business flow you want to cancel.
   The Group ID is displayed on the Web Console History page.
2. Navigate to the following directory:
   `<install_directory>/obt/bin`
   where, <install_directory> is the location where you installed the software.
3. Run the `cancelJob` API using the appropriate syntax:

Platform	Syntax
UNIX	./launch_groovyscript.sh -e <environment_name> -f ../scripts/cancelJob.groovy <groupRunId>
Windows	launch_groovyscript.bat -e <environment_name> -f ..\scripts\cancelJob.groovy <groupRunId>

where
<groupRunID> is the Group ID as defined on the Web Console.
<environment_name> is the name of the environment.
4. Optionally, start the Web Console to monitor the progress of the job.

Access the archived data

- Create database to database archive transparency

- Relocate database to file archive data

Create database to database archive transparency

See Create database to database archive access transparency

To create data transparency from the command line:

1. Ensure that you have performed the relevant tasks from Before you begin.
2. If necessary, perform the following tasks to ensure your configuration mode is set correctly for the transparency layer:
   1. Navigate to **Business Flow Management > Tasks > System Parameters > Database to Database** in the Web Console.
   2. Check the Archive Access section to ensure that the configuration mode parameter is set to the configuration option you want to use.
      See Manage environments.
3. Run the Create Archive Access job using the following syntax:

Platform	Syntax
UNIX	./launch_create_aa.sh [-r] –e <environment_name> [<transparency_layer_name>]
Windows	launch_create_aa.bat [-r] –e <environment_name> [<transparency_layer_name>]

Parameter	Description
-r	The recover flag, -r, allows you to recover jobs and business flows. See Recover business flows. Although the recover flag, -r, is optional, OpenText recommends that you use it whenever you run a job or business flow.
environment_name	The name of the environment. This is optional if you only have one environment.

Windows example

```
launch_create_aa.bat -e Oracle_Environment -r
```

4. Enter the encryption key at the prompt. The encryption key is case sensitive.
   The job runs, and displays a success or failure message.

5. Optionally, launch the Web Console to monitor the progress of the job.
   When the job completes successfully, you should be able to access the archived data from your application.

6. Try accessing some of the archived data from your application to ensure that the archive access is working.

Relocate database to file archive data

See Relocate database to file archive data

To relocate database to file archive data from the command line:

1. Make sure that you have performed the relevant tasks from Before you begin.
2. Open a command window.
3. Navigate to the following directory:
   `<install_directory>/obt/bin`
   where, <install_directory> is the location where you installed the software.
4. Run the job from the command line using the following syntax:

Platform	Syntax
UNIX	./launch_ea_job.sh [-r] -e <environment_name> -j <jobName> [sourceJobRunID=<runID>] [sourceLocation= <source_loc>] [summaryFileName=<file1,file2,...>] [destLocation=<dest_loc>] [mappingFileName= <map_file>]
Windows	launch_ea_job.bat [-r] -e <environment_name> -j <jobName> [sourceJobRunID=<runID>] [sourceLocation= <source_loc>] [summaryFileName=<file1,file2,...>] [destLocation=<dest_loc>] [mappingFileName= <map_file>]

Parameter		Description
-r		The recover flag, -r, allows you to recover jobs and business flows.
environment_name		The name of the environment. This is optional if you only have one environment.

jobName	One of the following:	
	XML_ARCHIVE@COPY_BE_TO_DB	Reloads data from a backend to the source database.
	XML_ARCHIVE@COPY_BE_TO_BE	Copies data from a backend to a backend.
	XML_ARCHIVE@UPLOAD_BE_TO_DB	Uploads data from a backend to a different database than the original source, checks for missing schema objects, and creates any schema objects that are missing.
	XML_ARCHIVE@MOVE_BE_TO_DB	Reloads data from a backend to the source database, and then deletes it from the backend.
	XML_ARCHIVE@MOVE_BE_TO_BE	Copies data from one backend to another backend, and deletes it from the original.
	XML_ARCHIVE@DELETE_FROM_BE	Deletes the data from the backend.
	XML_ARCHIVE@DELETE_FROM_DB	Deletes the data from the database.
	XML_ARCHIVE@REGISTER_ARCHIVE	Registers a file archive with another Structured Data Manager instance and creates new lineage with the destination as a file system in the current Structured Data Manager instance.
runID	The run ID of the job that placed the archive at the current location.	
source_loc	The Web Console location where the file archive to import is stored.	
file1,file2,...	A list of summary files for the file archives you wish to import.	

dest_loc	The name of the destination location as defined in the Web Console. Required for COPY_BE_TO_BE, MOVE_BE_TO_BE, and UPLOAD_BE_TO_DB only.
map_file	The name of the mapping file to use during upload when disconnected from the source database. See Map schemas by cartridge for more information.

Windows examples

- Copy a file archive from one backend to another:
  ```
  launch_ea_job.bat -e "MyEnvironment" -j "XML_ARCHIVE@COPY_BE_TO_BE" -r "sourceJobRunId=3"
  "destLocation=LOCAL_ARCHIVE_FS"
  ```
- Import a file archive leaving the original intact:
  ```
  launch_ea_job.bat -e oracle_env -j XML_ARCHIVE@IMPORT -r "sourceLocation=Copy_BE_2_BE"
  "summaryFileName=Order_Table_0955wwx5r66em90yntcmf46jj3ucudsscua_summary.xml.gz"
  "destLocation=Import_From_FS"
  ```
- Import a file archive deleting the original:
  ```
  launch_ea_job.bat -e oracle_env -j XML_ARCHIVE@IMPORT_MOVE -
  r "sourceLocation=Copy_BE_To_BE"
  "summaryFileName=Order_Table_0hrqqo3q6cve036sebzj2kh1v3o8fr9o0zm_summary.xml.gz,Order_Tab
  "destLocation=Import_To_FS"
  ```

5. Enter the encryption key at the prompt. The encryption key is case sensitive.
6. The job runs, and displays a success or failure message.
7. Optionally, start the Web Console to monitor the progress of the job.

1.3.6. Map schemas for upload or AQS cache

When uploading from a file to a database, you may have reasons to perform some level of mapping. You may need to upload into a different schema name than the one from which you archived the data, which would require a simple mapping of one schema name to another. For example, if you archived from a SQL Server database and are now uploading to an Oracle database, you may prefer a different schema name.

In other cases, where you are uploading to or from non-native databases, you may need to perform a more elaborate mapping that includes data types. For example, you may have a `real` data type in your source that should be mapped to the `float` data type in an upload to SQL Server and the `number` data type in an upload to Oracle.

When using AQS cache, which uploads to PostgreSQL, you might also need schema mapping to account for differences between your source database and PostgreSQL. For example, your source database might allow longer name lengths than PostgreSQL.

> **Note**
>
> You also should be conscious of different length definitions. One database might measure length in characters while another uses bytes. For example, if you are uploading data that originated in a database that measured length in characters to one that measures length in bytes, you need schema mapping to reconcile those differences during upload.

> **Note**
>
> In case of Oracle, if you have 'CHAR' column types, you must check the addendum "Extracting binary zero in Oracle CHAR or VARCHAR" to avoid issues sending data to AQS having char(0) ASCII data in char type columns of any table.

Structured Data Manager provides two methods for performing such schema mappings.

- Map schemas by cartridge
- Map schemas by database

1.3.6.1. Map schemas by cartridge

One way to map schemas for upload is to create a mapping for your cartridge. This method is best for one-off mappings that you only need for a particular cartridge. If you are doing many, similar uploads, it may be more cost effective to create a database schema mapping. For more information, see Map schemas by database.

> **Tip**
>
> A schema mapping file can also be useful when uploading to a database where the maximum size of a column's data type is set smaller than the incoming data. The mapping file allows you to prevent errors or properly handle truncation, depending on the database you are using.

To create and use a schema mapping for your database to file cartridges

1. Create a schema mapping file in a location that is accessible to your Structured Data Manager instance. The contents of the mapping file should adhere to one of the following formats, depending on whether you want to map:

   - target catalog, schema, and table

   - target catalog and schema

   - target catalog only

   The mapping of the qualified target occurs by matching the above conditions in the order they are specified in the mapping file. If there is a match to a condition, no further mappings are checked.

   To map the target catalog, schema, and table, use the following format:

   ```
   <src_catalog>.<src_schema>.<src_table>=<target_catalog>.
   <target_schema>.<target_table>
   ```

   > **Note**
   >
   > For **Target schema name**, ensure you select **Use driving table schema** option which is default if the cartridge contains two tables with same name in two different schemas. Incase, if the table names are different then you can select any of the following options for the Target schema name:
   >
   >   - Use driving table schema
   >   - Use Mapping file
   >   - Use Cartridge name
   >
   >   - Public

   > **Note**
   >
   > In the case of Oracle, there is no catalog. Hence, you would omit the catalog in cases involving Oracle. For example, if you are going from Oracle to SQL Server, you would not need <src_catalog>.

   > **Note**
   >
   > In case of Oracle, if you have 'CHAR' column types, you must check the addendum XXX to avoid issues sending data to AQS having char(0) ASCII data in char type columns of any table.

   To map the target catalog and schema, use the following format:

   ```
   <src_catalog>.<src_schema>=<target_catalog>.
   <target_schema>
   ```

   If the databases are heterogeneous, you might use this format instead:

   ```
   <src_catalog>.<src_schema>=<target_schema>
   ```

   To map the target database only, us the following format:

   ```
   <src_catalog>=<target_catalog>
   ```

If any of the names are null, you can safely omit the name. For example, if the source database does not have a schema, to map the table <src_table> from database <src_catalog>, you can specify

```
<src_catalog>.<src_table>=<target_catalog>.
<target_schema>.<target_table>
```

> **Note**
>
> You do not need to specify the catalog or schema name if they are not supported in the source or target database. Do not insert null for these names.

2. In the Web Console, click **Business Flow Management > Tasks > System Parameters**.

3. Click **Cartridge**.

4. Scroll down to the parameter **Schema Mapping File for Upload**. If you are performing mapping for AQS Cache, go to the **AQS Cache Options** section.

5. Enter the path and name of the schema mapping file that you created in Step 1.

1.3.6.1.1. Cartridge mapping files

When using schema mapping to upload from a source to a target where the databases are heterogeneous, or you want to upload into a different schema name than the one from which you archived, you need a schema mapping file. A schema mapping file is useful when one database supports catalog and the other does not. You can also use the mapping file to perform explicit column type mapping.

Use the mapping file to map the following from source to target:

> **Note**
>
> These are case sensitive.

- catalog name
- schema name
- table name
- column name
- column type name
- map index name
- primary key index name

> **Note**
>
> If the table, schema, or column name contains a dot (.), the name must be enclosed in double quotes ("). If the name already contains double quotes, add one double quote for each double quote. For example, if a catalog is C, the schema name is my.schema, and the table name is new"customer", the fully qualified name is C."my.schema".new""customer"".

Syntax

The following is the syntax to use for mapping objects. You need to create a mapping for each object that you want.

```
[source_catalog_name].[source_schema_name].
[source_table_name].[source_column_name].
[source_column_type_name]=[target_catalog_name].
[target_schema_name].[target_table_name].
[target_column_name].[target_column_type_name]
```

Schema mapping supports the following:

- Column type properties on target:

- length

- precision

- scale

- Precision and scale for all numeric types.

- Only character length for all character types.

You can set these properties on any appropriate target column type. For example, you can set the length on VARCHAR2:

```
DEMARC60.dbo.ORDERHEADER.COL1.date=
DEMARC.ORDER_HDR.COL1_1.VARCHAR2[length:18]
DEMARC60.dbo.ORDERHEADER.COL1.money=
DEMARC.ORDER_HDR.COL1_1.NUMBER[percision:18,scale:3]
```

> **Note**
>
> The standard method of specifying these properties is not supported because some data types contain parenthesis. For example, varchar(max). Properties should be specified within square braces as name value pairs separated by a colon.

Each line the your schema map must map only source to target mapping. For example, the following maps the data column type on the source to a VARCHAR2 type on the target.

```
DEMARC60.dbo.ORDER_HEADER.COL1.date=DEMARC.ORAER_HDR.
COL1_1.VARCHAR2
```

The preceding example does not map any other objects, such as catalog, schema, table, or column name.

1.3.6.2. Map schemas by database

When uploading data between heterogeneous databases, it is often best to create mappings at the database level rather than the cartridge level. You must consider carefully the mapping of data types between databases and perhaps even apply some conditional logic to determine the correct mapping. This sort of data type mapping is handled through a database mapping file. If you are uploading from or to a database that is non-native to Structured Data Manager, you need to create a mapping file. For native databases, you should be able to upload data from one database to another without updating the mapping file.

Non-native to native database

When uploading archiving data from a non-native database to a native database, a mapping file already exists for the target database. During the upload process Structured Data Manager attempts to map the column type with the help of the existing mapping file. It also throws an exception if it finds any invalid mapping, such as the target column is incorrectly mapped, no mapping is found, or multiple mappings are found.

If the upload process fails an exception is thrown and recorded in the log file. To fix the mapping issues, edit the mapping file then rerun the upload process. For information about logging, see Use log files. For information about editing mapping files, see Database mapping files.

Native to non-native database

When uploading to a non-native database from a native database, a mapping file does not exist. In this case, Structured Data Manager creates a mapping file when you run the upload process. If invalid mappings are found, the upload process fails, and exceptions are thrown and logged. To fix the mapping issues, edit the mapping file then rerun the upload process. For information about editing mapping files, see Database mapping files.

Non-native to non-native database

When uploading from a non-native database to a non-native database, a mapping file does not exist. In this case, the upload process creates a mapping file. If invalid mappings are found the upload process fails, and exceptions are thrown and logged. To fix the mapping issues, edit the mapping file then rerun the upload process. For information about editing mapping files, see Database mapping files.

Database mapping files

When uploading from or to non-native databases, you may use the database mapping files to handle the transposition of data types between databases. Structured Data Manager supports and includes mapping files for the following databases:

- DB2
- Oracle
- SQL Server
- Sybase

The mapping files are XML files that specify how one data type should be mapped to another: The mapping can be:

- Simple— One data type is mapped to another without any considerations.

- Database specific— The data type mapping depends on which database is used as the source. For example, if the data type is real, it is mapped to float for SQL Server; otherwise, it is mapped to NUMBER.

- Auto-expansion— The data types being mapped have a precision, length, or scale values but have different maximums so the mapping depends on the actual value. For example, if length exceed 4001 when mapping from varchar to a VARCHAR2 data type, it is mapped to CLOB instead of a VARCHAR2.

For the elements and structure required in a mapping file, see the mapping file schema `dataTypeMapping.xsd`. It is located in the directory:

```
<install_directory>/obt/foundation/components/config
```

where, `<install_directory>` is the location where Structured Data Manager is installed.

Example

The following is a example of simple mapping:

```
<mapping>
   <columnInput>real</columnInput>
   <columnOutput>NUMBER</columnOutput>
<mapping>
```

In this example, the real data type in the source is mapped to the NUMBER data type in target.

Example

The following is an example of database specific mapping:

```
<mapping>
    <columnInput>real</columnInput>
    <exceptions>
      <exception>
        <sourceDBname>
          <![CDATA[.*(sql).*(server)*.*]]>
        </sourceDBname>
        <columnOutput>float</columnOutput>
      </exception>
    </exceptions>
    <columnOutput>NUMBER</columnOutput>
</mapping>
```

In this example, the real data type is mapped to NUMBER for all source databases. However, if the source is a SQL Server database, real is mapped to float.

> **Note**
>
> The upload process tries to match sourceDBname in the mapping element to the folder name generated in the /obt/config/dbms directory. For example, the regular expression ".*(sql).*(server)*.*" is matched with Microsoft_SQL_Server_9_0.

Example

The following is an example of auto-expansion mapping:

```
<mapping applyRule="VARCHAR2_RULE">
    <columnInput>varchar</columnInput>
    <columnOutput>VARCHAR2</columnOutput>
</mapping>
<propertiesExceptions>
    <propertiesException ruleName="VARCHAR2_RULE">
      <condition minLength="4001">
        <columnOutput>CLOB</columnOutput>
      </condition>
      <condition isLengthSet="false">
        <columnOutput length="20">VARCHAR2</columnOutput>
      </condition>
    </propertiesException>
</propertiesExceptions>
```

In this example, varchar is mapped to VARCHAR2. Also, the ruleName attribute is set to VARCHAR2_RULE and applies to the mapping. The rule specifies that if the length of varchar exceeds 4000, it is mapped to CLOB.

The propertiesException element is used to create rules. You can apply rules to mapping or exception elements. The condition elements are executed in order. When a condition is met, the data type specified in the columnOutput element is used. No subsequent conditions are applied. The condition element supports the following attributes:

- maxLength

- minLength

- maxPrecision

- minPrecision

- maxScale

- minScale

- isLengthSet

- isPrecisionSet

- isScaleSet

Create and edit database mapping files

If your source and target databases are databases supported by Structured Data Manager, you do not need to edit or create a mapping file because a mapping file is already provided. However, if a database other than those supported is involved, you will need to create and edit a mapping file.

To create a mapping file

1. Open a command window and navigate to the directory <install_directory>/obt/bin, where, <install_directory> is the directory where the software is installed.

2. Run the script `create_datatype_map`.

   On Windows:

   `create_datatype_map.bat`

   On UNIX:

   `create_datatype_map.sh`

   The script connects to target database to create mapping for the source to target column type.

   > **Note**
   >
   > Using the create_database_map is optional. However, running the script allows you to identify issues with data type mapping before running the upload process.

Validate the database mapping file

To validate your mapping file against the schema (dataTypeMapping.xsd), run the following command:

On Windows:

`create_datatype_map.bat <mapping_file_name> -v`

On UNIX:

`create_datatype_map.sh <mapping_file_name> -v`

where <mapping_file_name> is the name of your mapping file.

Create mapping by connecting to the source database

To create data type mapping in your mapping file by connecting to the source database, run the following command:

On Windows:

`create_datatype_map.bat -f`

On UNIX:

`create_datatype_map.sh -f`

The script connects to the source and target database to create mapping for the source to target column type.

Extracting binary zero in Oracle CHAR or VARCHAR

Some Oracle instance may have binary zero (CHR(0)) in the character strings. Columns containing CHR(0) characters can be extracted by SDM but fails refreshing the AQS cache.

In dbvisualizer the data would appear as:

1. In a text editor, open the mapping file created by the script and fill in the data types for the target column. Skeleton mapping elements are added to the end of the mapping file.

   When editing the mapping file, consider the following:

   - The schema dataTypeMapping.xsd provides the usage information for

     - auto-expanding a target data type based on certain properties.

     - mapping for a specific source database is supported.

     - setting certain target column data type properties.

   - The sourceDBname and columnInput elements can accept a regular expression. Regular expressions should be inside a CDATA section.

1. Create a SDM masking function

   In SDM home directory ${home}\extensions\runtime\masking

   Create a groovy file: cleanzerobin.groovy with the following content.

   ```
   Object mask(Object _in) {
   return _in.replaceAll("[\\x00-\\x00]"," ");
   }
   Object revert_mask(Object _in){
   return _in;
   }
   ```

2. Add cleanzerobin masking function.

3. Apply the cleanzerobin masking function to dirty columns.

4. Run the business flow with AQS cache enabled and you should have all the binary zeros replaced with space.

5. In dbvisualizer the data in AQS_CACHE now appear as:

1.3.7. Datatype mapping list for AQS Cache and Upload

AQS Cache Mapping

> **Note**
>
> For each source database, the corresponding Column Input data types are mapped to Column Output data types in the AQS cache database.

Source: DB2

Column Input	Column Output
VARGRAPHIC	VARCHAR
GRAPHIC	CHAR
INT	INTEGER
NVARCHAR	VARCHAR
NCHAR	CHAR
FLOAT	NUMERIC
TIME	VARCHAR
TIMESTAMP	VARCHAR
DATETIME	VARCHAR
DECFLOAT	VARCHAR
DOUBLE	DOUBLE PRECISION
CHAR() FOR BIT DATA	BYTEA
VARCHAR() FOR BIT DATA	BYTEA
BLOB	BYTEA
XML	TEXT

Source: MYSQL

Column Input	Column Output
BINARY	BYTEA
VARBINARY	BYTEA
BLOB	BYTEA
DOUBLE	DOUBLE PRECISION
FLOAT	DOUBLE PRECISION
DATETIME	TIMESTAMP
MEDIUMINT	DOUBLE PRECISION
TINYINT	SMALLINT
INT	INTEGER
BIGINT UNSIGNED	NUMERIC
MEDIUM UNSIGNED	INTEGER
SMALLINT UNSIGNED	INTEGER
TINYINT UNSIGNED	SMALLINT
INT UNSIGNED	BIGINT

Source: ORACLE

Column Input	Column Output
VARCHAR2	VARCHAR
NUMBER	NUMERIC
FLOAT	NUMERIC
BINARY FLOAT	REAL
BINARY DOUBLE	DOUBLE PRECISION
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE
TIMESTAMP WITH*	TIMESTAMP WITH TIME ZONE
INTERVAL YEAR TO MONTH	INTERVAL
INTERVAL DAY TO SECOND	INTERVAL
ROWID	CHAR
LONG	TEXT
CLOB	TEXT
RAW	BYTEA
LONG RAW	BYTEA
BLOB	BYTEA
XMLTYPE	XML

Source: SQLSERVER

Column Input	Column Output
NVARCHAR	VARCHAR
NVARCHAR MAX	INT
NCHAR	CHAR
NTEXT	TEXT
INT	INTEGER
FLOAT	DOUBLE PRECISION
DOUBLE	NUMERIC
XML	TEXT
TINYINT	SMALLINT
BIT	BOOLEAN
SMALL / MONEY	DECIMAL
UNIQUEIDENTIFIER	VVID
BINARY	BYTEA
TIMESTAMP	BYTEA
VARBINARY	BYTEA

Source: SYBASE

Column Input	Column Output
INT	INTEGER
NVARCHAR /UNIVARCHAR	VARCHAR
NCHAR /UNICHAR	CHAR
UNITEXT	TEXT
IMAGE	BYTEA
TINYINT	SMALLINT
BIT	CHAR
VARBINARY	BYTEA
FLOAT	DOUBLE PRECISION
DATETEIME	TIMESTAMP
UNSIGNED BIGINT	BIGINT
UNSIGNED SMALLINT	SMALLINT
UNSIGNED INT	INT
SMALL / MONEY	DECIMAL

Upload Mapping

> **Note**
>
> If the source database has a Column Input datatype, it will be mapped to the Column Output datatype in the destination database during upload.

Destination: DB2

Column Input	Column Output
NCHAR	CHAR
NVARCHAR	VARCHAR
VARCHAR2	VARCHAR
TIMESTAMP WITH*	VARCHAR
ROWID	CHAR
RAW	BLOB
LONG	CLOB
NUMBER	DECIMAL
FLOAT	DOUBLE
NUMERIC	DECIMAL
TINYINT	SMALLINT
MONEY	DECIMAL
BINARY	BLOB
TEXT	CLOB
UNITEXT	LONG VARCHAR
NTEXT	CLOB
XMLTYPE | XML	CLOB
NCLOB	DBCLOB
IMAGE	BLOB
BIT	CHAR
UNIQUEIDENTIFIER	VARCHAR

Destination: ORACLE

Column Input	Column Output
DATETIME	DATE
TIME	VARCHAR2
IMAGE	BLOB
TEXT	CLOB
REAL	NUMBER
DOUBLE	NUMBER
UNIQUEIDENTIFIER	VARCHAR2
BINARY	RAW
NUMERIC	NUMBER
INT	NUMBER
DECIMAL	NUMBER
MONEY	NUMBER
LONG VARCHAR	CLOB
XML	CLOB
BIT	NUMBER
CHAR() FOR BIT DATA	RAW
DECFLOAT	NUMBER
GRAPHIC	VARCHAR2
VARCHAR	VARCHAR2
NVARCHAR	NVARCHAR2
VARGRAPHIC	VARCHAR2

Destination: SQLSERVER

Column Input	Column Output
XMLTYPE	XML
NUMBER	NUMERIC
TIMESTAMP	DATETIME
TIMESTAMP WITH*	VARCHAR
ROWID	CHAR
TIME	VARCHAR
NVARCHAR2	NVARCHAR
BLOB	IMAGE
CLOB	TEXT
NCLOB	NTEXT
DBCLOB	NTEXT
DECFLOAT	NUMERIC
GRAPHIC	VARCHAR
VARGRAPHIC	TEXT
BINARY	IMAGE
CHAR() FOR BIT DATA	VARBINARY
DOUBLE	NUMERIC
DECIMAL	NUMERIC
UNSIGNED INT	INT
VARCHAR2	VARCHAR
LONG RAW	IMAGE
LONG	TEXT

Destiation: SYBASE

Column Input	Column Output
INTEGER	INT
TIMESTAMP WITH*	VARCHAR
ROWID	CHAR
NVARCHAR2	VARCHAR
NCHAR	CHAR
NUMBER	NUMERIC
LONGVARGRAPHIC	LONGVARCHAR	TEXT
BLOB	IMAGE
NCLOB	TEXT
DBCLOB	TEXT
DECFLOAT	DECIMAL
VARGRAPHIC	VARCHAR
TIMESTAMP	VARCHAR
DECIMAL	NUMERIC
BIGINT	NUMERIC
SMALLINT	SAMLLINT
UNSIGNED INT	INT
NTEXT	TEXT
XML	XMLTYPE	TEXT
UNIQUEINDENTIFIER	VARCHAR

Destination: VERTICA

Column Input	Column Output
REAL	FLOAT
BLOB	VARCHAR
TIMESTAMP WITH TIME ZONE	TIMESTAMP
DATETIMEOFFSET	TIMESTAMPTZ
TIMESTAMP WITH LOCAL TIME ZONE	TIMESTAMP
LONG RAW	BINARY
NUMBER	NUMERIC
DECIMAL	NUMERIC
CLOB	VARCHAR
LONG	VARCHAR
INT	INTEGER
DATETIME	TIMESTAMP
SMALLDATETIME	TIMESTAMP
TEXT	VARCHAR
SMALLINT	INTEGER
TINYINT	INTEGER
MONEY	NUMERIC
SMALLMONEY	NUMERIC
IMAGE	VARBINARY
NTEXT	VARCHAR
XML	VARCHAR
UNIQUEIDENTIFIER	VARCHAR
ROWID	CHAR

1.3.8. Perform history schema reconciliation

When using database to database archiving, you need to ensure that your target (history) schema remains synchronized with your source database schema. If the managed tables in the source database have been modified in some way (for example, columns added or deleted), you need to reconcile the history schema where data was archived to match those changes.

This chapter describes how to perform the history schema reconciliation to update your archived data.

You must run the history schema reconciliation only when you have changes in your source database that are not reflected in the history (target) database. For example, if some columns in the source tables have been added or deleted, you would then need to run history schema reconciliation.

- About the history schema reconciliation process
- Overview of the history schema reconciliation process
- Prepare to run the history schema reconciliation
- Discover differences between the source and target databases
- Generate the XML script
- Edit the XML script
- Deploy the XML script
- Verify results
- Check the log files

1.3.8.1. About the history schema reconciliation process

The history schema reconciliation process allows you to keep your source and target database schemas consistent. Over time, you inevitably make changes to your source database objects, such as adding new columns or upgrading versions. For example, if the US-based ACME company expanded to an overseas market, they would need to add a currency-type (dollars, euros, yen) column to their Orders table. They would need to add this column to all the tables and back-fill the old orders with USD (United States Dollars).

To reflect these types of changes in your target, the history schema reconciliation process discovers changes to the managed tables of your source schema and generates a corresponding XML script for your target schema.

You should perform a history schema reconciliation after you complete any of the following activities:

- Upgrade the source database.
- Apply an application patch to the source database that modifies a managed table.
- Customize the database resulting in changes affecting the managed tables of the cartridges.

When you perform the history schema reconciliation process, it performs the following tasks for you:

- Determines if there are differences in the managed tables of the source and target schema by comparing attributes:
- column datatype
- column datatype size
- column nullable status
- column default value
- missing individual columns
- dropped columns
- added columns
- sets of columns attributes
- Detects indexes that were modified in the source and indexes that exist in the target but do not exist in the source and drops them. New indexes created in the source will not be detected. However, if you want to add those indexes into the history schema, you can create them:
- In history reconciliation by modifying the `<prefix>_build.xml` file, which is located in the home directory under `/businessflow/schema_upgrade/`, before you deploy it. For more information about the home directory, see Manage the home directory.
- In an archive run.
- By adding the indexes manually directly to the database on the history side.

> **Note**
>
> Tables in the target database should not have more than 20 indexes due to the limitation with Oracle on using union views over database links.

If there are some indexes that you do not want to remove, modify the `<prefix>_build.xml` file, which is located in the home directory under `/businessflow/schema_upgrade/`, to delete the respective DROP INDEX statements before you deploy the script.

- Generates an XML script in the home directory under `/businessflow/schema_upgrade` to synchronize the source and target databases. The script can add, drop, or modify columns, and automatically populate new columns, if the new columns are not set to NULL. For example, suppose you change a LEDGER table to insert the AMOUNT_PLASTIC column, which tracks the amount of plastic in every bottle. The AMOUNT_PLASTIC column must have a numerical value because every bottle has some plastic in it. The XML script automatically fills this non-null column with a zero and flags it for editing.

The XML script contains searchable markers that you need to edit to make the current source and target managed tables match. After you have edited the XML script, you can deploy it in test, integration, and production environments. OpenText recommends carefully testing the XML script in each environment before progressing to the next.

By adding the history schema reconciliation process to the patch or upgrade process you can maintain consistency between the source and target as well as in the managed tables used by the cartridges in your business flows. For more information on cartridges and business flows, see the *OpenText™ Structured Data Manager Concepts Guide* or the *OpenText™ Structured Data Manager Developer's Guide*.

1.3.8.2. Overview of the history schema reconciliation process

The history schema reconciliation process consists of the following steps, which are described in greater detail in the following sections:

1. Run the history schema reconciliation script in checkonly mode.

2. Generate the XML script.

3. Edit the XML script.

    1. Search for TODOs.

    2. Add SQL statements where needed.

    3. Add data to any columns designated not-null.

    4. Remove or add INDEX statements where needed.

1. Deploy the XML script in the test environment. See Deploy the XML script.

2. Perform database tests to determine proper stability and function.

3. Diagnose any issues.

4. Edit the XML upgrade script again.

    1. Search for TODOs.

    2. Add or edit SQL statements as needed.

    3. Add or alter data to not-null columns as needed.

    4. Refresh the target database, if necessary.

    1. Repeat from step 3 as necessary.

    2. Deploy the XML script.

1.3.8.3. Prepare to run the history schema reconciliation

OpenText recommends running the history schema reconciliation process in a test environment or on a copy of the source database before deploying in the integration or production environments.

Before beginning the reconciliation, ensure below tasks are completed.

Verify current installation

You must have the most recent version of Structured Data Manager and the most recent patch installed.

Ensure that all the necessary requirements for Structured Data Manager are met before you install the software.

See *OpenText™ Structured Data Manager Installation Guide*

Verify current software version

Ensure that you are running the latest version of Structured Data Manager.

1. Log on to the source server as a user with the appropriate permissions.

2. Navigate to the following directory:

   `<install_directory>/obt/bin`

   where, <install_directory> is the location where you installed Structured Data Manager.

3. Type the appropriate command:

Operating System	Command syntax
UNIX	./version_info.sh
Windows	version_info.bat

The script displays the version of Structured Data Manager that is installed.

Verify completed jobs and the environment name

Use the Web Console to check for running and in-process database to database archiving business flows and jobs, and to determine the name of the environment.

To determine the status of the jobs

1. Start the Web Console and login to the Web Console.
2. From the Web Console home page, click **Business Flow Management**.
3. Click **Tasks > Monitoring > Overview** from the menu at the top of the page.
   The overview page displays the status of your business flows as well as any currently running commands.

To determine that no archiving jobs are scheduled to be run:

1. Start the Web Console and login to the Web Console.
2. From the Web Console home page, click **Business Flow Management > Tasks > Scheduler**.
   The Scheduler displays the list of scheduled jobs.

To determine the name of the environment

1. Start the Web Console and login to the Web Console.
2. From the Web Console home page, click **Environment Management > Environment > Manage**.
   The Manage Environments page displays all of the environments in your installation.

Verify necessary changes

Run the history schema reconciliation script in check only mode to verify if changes are required in the target.

> **Note**
>
> If the schemas for your production database and your archived historical data match, there is no need to run the history schema reconciliation script.

1. Open a command window.
2. Navigate to the following directory:

   `<install_directory>/obt/install`

   where, <install_directory> is the location where you installed OpenText™ Structured Data Manager .
3. Run the prepare_history_upgrade script using the appropriate syntax:

Operating System	Command syntax
UNIX	prepare_history_upgrade.sh -e <environment_name> [-key <EKEY>] -checkonly
Windows	prepare_history_upgrade.bat -e <environment_name> [-key <EKEY>] -checkonly

Parameter	Description
<environment_name>	The name of the environment.
<EKEY>	The case-sensitive encryption key you defined during installation. The <EKEY> is optional. If you place the encryption key into the command line, it is visible on the screen. If you do not use the <EKEY>, you are prompted to type the encryption key later.
-checkonly	The command to compare the source and target databases without generating an XML script. This option lets you know if a schema reconciliation is necessary.

UNIX example

`./prepare_history_upgrade.sh -e ENV1 -key obtkey -checkonly`

> **Tip**
>
> To see all the options for the history schema reconciliation script, use the -help option when you run it. For example, prepare_history_upgrade.bat -help.

If no changes are discovered, the following message is returned:

`OLTP and History match...`

If schema changes are discovered, the following message is returned:

`OLTP and History do not match. Need to create a History schema upgrade script.`

> **Note**
>
> If the schemas for your production database and your archived historical data match, there is no need to run the history schema reconciliation script.

1.3.8.4. Discover differences between the source and target databases

Your existing change management processes should track changes from patches, customizations updates, or DML (Data Manipulation Language). The history schema reconciliation helps you get the most from the system by ensuring that the source and target database table definitions match.

Differences between the source and target databases are typically the result of patches or customizations applied to the source database tables.

To identify the scope of changes

1. Use your company's database change management process to note, record, and communicate any changes to the source database tables managed by the cartridges in your business flows.

2. Log in to the repository, and run a query similar to the following to retrieve a list of all the managed tables.

```
SELECT a.appspack_name, mt.table_catalog, mt.table_schema, mt.table_name
   FROM obtpa_appspacks a,
        obtpa_appspack_table_uses atu,
        obtpa_managed_tables mt
WHERE a.id = atu.appspack_id
   AND mt.id = atu.managed_table_id
   ORDER BY 1,2,3,4;
```

1. Examine the patch upgrades for changes to the managed tables by looking at the log or the DML for the patch or upgrade.

2. Visually examine the source database to determine what updates have been applied to the managed tables.

1.3.8.5. Generate the XML script

The XML script allows you to update the target database to make it match the source database.

1. Open a command window.

2. Navigate to the directory that contains the `prepare_history_upgrade` script:

   `<install_directory>/obt/install`

1. Run the `prepare_history_upgrade` script using the appropriate syntax:

Operating System	Command syntax
UNIX	./prepare_history_upgrade.sh -e <environment_name> [-key <EKEY>] [-keepfiles] [-outputDir <output_dir>] [-prefix <prefix>] [-overwrite]
Windows	prepare_history_upgrade.bat -e <environment_name> [-key <EKEY>] [-keepfiles] [-outputDir <output_dir>] [-prefix <prefix>] [-overwrite]

Parameter	Description
<environment_name>	The name of the environment.
<EKEY>	The case-sensitive encryption key you defined during installation. The <EKEY> is optional. If you place the encryption key into the command line, it is visible on the screen. If you do not use the <EKEY> command here, you are prompted to type a hidden encryption key later.
-keepfiles	Retains all of the working files used by the reconciliation process. The files are placed in the same directory as the reconciliation script or the specified output directory.
<output_dir>	Use the -outputDir keyword to set a different output directory for the generated files. By default, the output is placed in the directory where the script is located.
<prefix>	The name chosen to specify a unique prefix to track various versions of the generated reconciliation script. Use sequential numbers, patch numbers, release numbers, or timestamps as unique names. The prefix is optional. By default, the output of the reconciliation script is named: HISTORY_UPGRADE_build.xml
-overwrite	An optional command you can use to overwrite a previously generated XML script of the same name. You cannot overwrite a script without using the -overwrite option. If you use the same name, the script fails with an error message.

Example

```
./prepare_history_upgrade.sh -e oracle_oltp_env -outputDir ../businessflow/schema_upgrade
```

> 💡 **Tip**
>
> To see all the options for the history schema reconciliation script, use the `-help` option when you run it. For example, `prepare_history_upgrade.bat -help`.

The generated XML script is saved to the current working directory, or the output directory you specify.

1.3.8.6. Edit the XML script

Edit the XML script as necessary to ensure that the source database matches the target database. You can add data definition language (DDL) operations, such as index or keys that do not exist in the generated XML script. You can add DML for deletes, updates, or inserts that do not exist in the generated XML script.

The generated XML script handles schema structure changes including:

- adding a column

- dropping a column

- revising the column data type

- revising the column width

If the changes in source database also involve data transformations, such as the following, then you must edit the reconciliation script to ensure the database schemas match:

- using application tt to transform data

- adding a default column

- adding or filling a not null column

Data changes that require editing are marked by one of the following TODO comments:

Type	Statement
default seeding	`<!-- TODO: Correct the following default SQL seeding statement, it is most likely to be incorrect.-->`
data transformation	`<!-- TODO: If needed, add data transformation SQL statement here by uncommenting the obtsql task and replacing the "SQL Statement" with a real SQL DML statement -->`

Not all of the TODO statements in the generated XML script require editing.

> **Note**
>
> If you want to use the greater than (>) or less than (<) expressions, you need to quote them according to XML 1.0 rules. The greater than (>) symbol should be written as > and the less than symbol (<) should be written as <.

To edit the XML script

1. Navigate to the directory containing the generated XML script.

   `<OBT_HOME>/businessflow/schema_upgrade/<prefix>_build.xml`

Parameter	Description
<install_directory>	The location where you installed OpenText™ Structured Data Manager
<prefix>	The name used to specify a unique prefix to the XML script. If you did not specify a prefix, the XML script is called HISTORY_UPGRADE_build.xml.

1. Use an XML editor or text editor to open the XML script.

2. Save the file with a unique name.

3. Search for comments containing the text TODO.

   Example

   ```
   <!-- TODO: If needed, add data transformation SQL statement here by uncommenting the obtsql task
   and replacing the "SQL Statement" with a real SQL DML statement -->
   <!--
   <obtsql connectionSource="${cs.HIST}" autocommit="true" echo="true"> SQL statement </obtsql>
   -->
   ```

   The TODO comments are accompanied by the obtsql element which is used to contain the optional SQL statement. The statements can appear before or after the specified update.

4. Uncomment the obtsql element that you want to edit.

   Example

   ```
   <!-- TODO: If needed, add data transformation SQL statement here by uncommenting the obtsql task
   and replacing the "SQL Statement" with a real SQL DML statement -->
   <obtsql connectionSource="${cs.HIST}" autocommit="true" echo="true"> SQL statement </obtsql>
   ```

5. Replace the placeholder text with a valid SQL statement.

   Example

   ```
   <!-- TODO: If needed, add data transformation SQL statement here by uncommenting the obtsql task
   and replacing the "SQL Statement" with a real SQL DML statement -->
   <obtsql connectionSource="${cs.HIST}" autocommit="true" echo="true">
   UPDATE TABLE_NAME SET CURRENCY_tt = ' '
   WHERE CURRENCY_tt IS NULL
   </obtsql>
   ```

   > **Note**
   >
   > Ensure that the SQL statements do not see tables that have been removed from the schema.

   You can edit the entire SQL statement or just a portion.

   For example, you can change the currency tt from an empty value to United States dollars (USD).

   Example

   ```
   <obtsql connectionSourse="${cs.HIST}"autocommit="true"
   echo="true">
   UPDATE TABLE_NAME SET CURRENCY_tt = 'USD'
   WHERE CURRENCY_tt IS NULL
   </obtsql>
   ```

   If adding non-null columns, the following placeholders are used:

Non-null column type	Placeholder value
date	1970
number	0
string	an empty space between single quotes ' '

6. Edit the TODO comment with a searchable string.

Example

```
<!-- Edited by John Doe on Dec. 8, 2008 -->
<obtsql connectionSource="${cs.HIST}" autocommit="true" echo="true"> UPDATE TABLE_NAME SET
CURRENCY_tt = ' '
WHERE CURRENCY_tt IS NULL </obtsql>
```

7. Save the edited XML script.

> **Note**
>
> The XML script elements that you define must be compatible with the environment where the XML script is prepared and used. For example, the test database can have columns the production database does not have.

> **Note**
>
> The history schema reconciliation uses Apache ANT to deploy the changes. If you want to add additional statements to the XML script, see the Apache ANT 1.7.0 user manual.

1.3.8.7. Deploy the XML script

After you have edited the XML script, you can deploy it against test, integration, and production environments. OpenText recommends carefully testing the XML script in each environment before progressing to the next. Deployment of the history reconciliation script creates interface and relocation tables and reinstalls the business flows. After you deploy the edited script, there is no undo. You should deploy and fully test the XML script before deploying it against a production database.

To deploy the XML script

1. Open a command window.

2. Log in as a user with the appropriate permissions.

3. Navigate to `/businessflow/schema_upgrade` under the home directory. See [Manage the home directory](#).

4. Run the deploy_history_upgrade script using the following syntax:

Operating System	Command syntax
UNIX	<install_directory>\obt\install\deploy_history_upgrade.sh <upgrade_script>
Windows	<install_directory>/obt/install/deploy_history_upgrade.bat <upgrade_script>

Parameter	Description
<install_directory>	The location where you installed OpenText™ Structured Data Manager
<upgrade_script>	The name you use specify a unique version of the XML script. Use this file name to track various versions of the generated XML script. For example, you could use sequential numbers, patch numbers, release numbers or timestamps as unique names.

Example

`/export/home/user/SDM/obt/install/deploy_history_upgrade.sh HISTORY_UPGRADE_build.xml`

1. Provide the encryption key when he script prompts you for it.

   After you provide the encryption key, the script runs against your database. If the deployment is successful, you receive a message stating `Deployment successful`. You can then confirm the results as per [Verify results](#).

   Otherwise, you receive a message stating:

   `Failed to deploy. Please see the log output for more information.`

   In that case, you should review the log files to determine what went wrong as per [Check the log files](#).

1.3.8.8. Verify results

After you have deployed the edited XML script in the test, integration or production environments, follow your company procedures for running test cases. Ensure that you run reports that would reveal empty non-null columns.

To verify results in the test environment:

- check the table definitions for expected changes.

- compare the managed tables for expected changes.

  If empty columns are discovered after running test cases then you can re-seed the non-null columns. The empty columns occur when either:

- the XML script did not update all records.

- not all values were applied as expected.

Search for the SQL statements associated with non-null columns and correct the default value or SQL statement.

1.3.8.9. Check the log files

All the schema changes applied to the target database are kept in a tracking log. The log file shows the succession of steps the history reconciliation schema completed. Each one automatically verifies that the previous step completed successfully.

Check the .out file

By default, the `.out` file is saved to the schema reconciliation directory. If you have not specified a unique prefix, the default name is `HISTORY_UPGRADE_build.out`. The output from each run of the history schema reconciliation is appended to the `.out` file with a timestamp.

To check the .out file:

1. Open a command window.

2. Navigate to the schema reconciliation directory:

   `<OBT_HOME>/businessflow/schema_upgrade`

3. Open the log file.

4. Use the log files to find information on the following:

   - incorrect encryption key
   - incorrect password
   - incomplete test environment
   - incorrect copying of the tt tree

5. Close the log file.

Check the obt.log file

By default, the logging level is set to INFO.

1. Open a command window.

2. Navigate to the log file:

   `<install_dir>/obt/log/`

3. Open the `obt.log` file.

4. Use the log files to find information on the following:

   - incorrect encryption key

   - incorrect password

   - unknown environment

   - incomplete test environment

   - out of resources, especially when applying updates

   - business flows copied to wrong location

   - incorrect copying of the tt tree

5. Close the log file.

Edit the logging properties

After you have installed the product, you can edit the `log4j2.properties` file to change the logging properties. For example, you can change:

- where the log files are kept.

- what information is logged.

- the maximum size of the log file.

- how much logging information is kept.

By default, the `obt.log` file is limited to 100 MB. When it exceeds the default limit, the obt.log file is renamed to a backup file, and a new obt.log file is created. By default, a total of nine backup files are kept, limiting the log files to a 1000 MB maximum size.

To edit the `log4j2.properties` file

1. Navigate to the directory that contains the `log4j2.properties` file.

   `<install_directory>/obt/config/`

   where, <install_directory> is the location where you installed OpenText™ Structured Data Manager.

1. Open the `log4j2.properties` file by using a text editor and edit appropriately.
2. Save the `log4j2.properties` file.

The changes are applied automatically.

1.3.9. Use AQS to access database to file output

After your data has been archived to XML, CSV, or JSON files, you can use the Archive Query Server (AQS) cache to directly access your data using SQL reporting and development tools on Windows and UNIX, and join the data with existing data in a database.

AQS uses PostgreSQL to cache your archived data for query access.

> ⚠️ **Important**
>
> Concurrent AQS cache transactions are not supported in Structured Data Manager.

Configure the AQS cache

To use the AQS cache, you must specify a location for AQS_CACHE and configure the cache parameters in your environment:

1. If the Web Console is not open in your browser, invoke it by accessing its URL, for example, `http://localhost:8080/WebConsole` .

2. Login as the admin user, whose password you set when installing the repository, or as another user with administrative privileges.

3. Click **Environment Management > Environment > Active Environment > Locations** on the Web Console home page.

> 💡 **Tip**
>
> If necessary, you can activate a different environment by clicking **Environment Management > Environment > Manage**, selecting the desired environment, and clicking **Set Active**.

1. Click **AQS_CACHE**.

2. Enter the connection information for a PostgreSQL database instance. If you installed an instance as part of Structured Data Manager installation, use that one. If not, you will need to use an existing instance or install one.

Field	Description
Hostname	Enter the name or address of the machine where the database in installed
Port	Enter the port number of your PostgreSQL database. For example: 5432
Database Name	Enter the name of the PostgreSQL database.
Additional Parameters	Click on the + button if you want to pass any additional parameters while connecting to the database. To remove the added parameter, click - button. The formed JDBC URL is displayed to review the same when you use this option
User ID	Enter the username for PostrgreSQL database user
Password	Enter the password of the PostrgreSQL database user

> **Note**
>
> Some PostgreSQL requirements differ from other databases. For example, PostgreSQL only allows schema, table, or column names to be a maximum of 63 characters whereas DB2 allows names up to 128 characters. Hence, if you are archiving from a source database of DB2, you need to implement schema mapping for your AQS cache to ensure that your schema, table, and column names are handled appropriately. See Mapping schemas for upload or AQS cache for more information about schema mapping.

3. Click **Update**.

> **Tip**
>
> Notice that this page also has buttons for refreshing and clearing the cache. When you need to perform those functions, remember to return here.

4. Choose **System Parameters > Database to File**.

5. Scroll to the **AQS Cache Options** section.

6. Set **Enable AQS Cache** to **true**.

7. Optionally, set the other AQS parameters. Click the information icons to get more information about the purpose of the parameter. For more information on setting these parameters, see *OpenText™ Structured Data Manager Runtime Guide*. Since you have already run your archive, you must refresh the cache to populate it with all of your archived records.

> **Note**
>
> You can specify the schema for the AQS cache in outerbay.properties, which is stored in the home directory under config. See Managing the home directory

8. Click **Locations** from the left navigation pane.

9. Click **AQS_CACHE**.

10. Click **Refresh All Cache**.

11. The AQS cache should now be populated and you can run queries against it as you would any other database.

1.4. Integration

This part provides information to integrate Structured Data Manager with the following products:

- Content Manager Integration
- Hadoop Integration
- Data Privacy and Protection Appliance (SecureData) Integration

1.4.1. Content Manager Integration

To more easily comply with government regulations and corporate policies, you can integrate Structured Data Manager with Content Manager. Content Manager helps secure information from inappropriate access and misuse. To integrate Structured Data Manager with Content Manager, you must meet the following prerequisites:

Prerequisites for Content Manager

For integration with Structured Data Manager, you must perform the steps that follow on your Content Manager installation. For detailed installation, configuration, and usage information on Content Manager, refer to your Content Manager documentation.

Internet Information Services (IIS) settings

1. In the **IIS Manager**, go to the **Default Web Site**.
2. Under Default Web Site, double-click on **CMServiceAPI**.
3. Click on **Authentication** tab.
4. In the **Authentication** pane, select **Basic Authentication**, and then, in the Actions pane, click **Enable**.
5. In the **Authentication** pane, select **Windows Authentication**, and then, in the Actions pane, click **Enable**.
6. Ensure you can access the **ServiceAPI** URL from the browser where the Structured Data Manager is installed.

CM 9.1 setup

Perform the following:

1. Create **New Custom Property** or **Additional Fields**.
   1. Open CM Admin Console.
   2. In the left panel, **Search for** drop down menu. Select **Custom Properties** or **Additional Fields**.
   3. In right-side pane, right-click and select **New Custom Property**.
   4. The **New Custom Property** dialog box appears, select **String** as **Data format for custom property** and click **Next**.

      On the **New Custom Property - String Field** page, create the following three Custom Properties using the above steps:

         1. Specify **Name** as `File Name` and **Maximum Length** as `200` and click **Next**.
            On the **New Custom Property - Applies To** page, click **Finish**.
         2. Specify **Name** as `RunID` and click **Next**.
            On the **New Custom Property - Applies To** page, click **Finish**.
         3. Specify **Name** as `DBA Environment ID` and click **Next**.
            On the **New Custom Property - Applies To** page, click **Finish**.
2. Add **Custom Properties** to **New Record Type**.
   1. Open CM Admin Console.
   2. In the left panel, **Search for** drop down menu. Select **Record Types**.
   3. On the right-side pane, right-click and select **New Record Type**.
   4. The **New Record Type** dialog box appears.
   5. Specify **Name**, and select **Folder** in the **Behaves Like** drop-down menu.
      Go to the **Custom Properties** tab and click on **RunID** and **DBA Environment ID** fields. On clicking the fields, you must see a check mark appearing and then click **OK**.
   6. Specify **Name** and select **Document** in the **Behaves Like** drop-down menu.
      Click on the **Custom Properties** tab and click on **File Name**, **RunID**, and **DBA Environment ID** fields. On clicking the fields, you must see a check mark appearing and then click **OK**.
3. Create a **New Category** or **Classification**.
   1. Open CM Admin Console.
   2. In the left panel, **Search for** drop down menu. Select **Categories**.
   3. In right-side pane, right-click and select **New** > **New Top Level Item**.
   4. The **New Category** dialog box appears.
   5. Specify **Name** for the category.
   6. Go to **Numbering** tab, choose the **Pattern** and specify a **Value** for the category.
   7. Click **OK**.

CM 9.2 and above setup

Perform the following:

1. Enable **Structured Data Management Integration** feature.
   1. Open **ADMINISTRATION** tab in the CM Admin Console.
   2. Click **System** and select **Features** tab in the **System Options** window.
   3. Click **Structured Data Management Integration** to enable the feature.
   4. Click **OK**.
2. Create two **New Record Types**.
   1. Open CM Admin Console.
   2. In the left panel, **Search for** drop down menu. Select **Record Types**.
   3. On the right-side pane, right-click and select **New Record Type**.
   4. Specify **Name** and select **SDM Folder** in the **Behaves Like** drop-down menu.
   5. Specify **Name** and select **SDM Document** in the **Behaves Like** drop-down menu.
   6. Click **Ok**.
3. Create a **New Category** or **Classification**.
   1. Open CM Admin Console.
   2. In the left panel, **Search for** drop down menu. Select **Categories**.
   3. In right-side pane, right-click and select **New** > **New Top Level Item**.
   4. The **New Category** dialog box appears.
   5. Create **Name** for the category.
   6. Go to **Numbering** tab, choose the **Pattern** and specify a **Value** for the category.
   7. Click **OK**.

Transferring larger files

For files larger than 4MB, you must add the following tt in the **web.config** file located at path `\Drive:\Program Files\Micro Focus\ContentManager\Service_API\web.config`:

1. Under `<system.webServer>` tag:

```
<security>
<requestFiltering>
<requestLimits maxAllowedContentLength="1073741824" />
</requestFiltering>
</security>
```

2. Under `<system.web>` tag:

   Add the `maxRequestLength` attribute in `<httpRuntime>` tag.

   For example, `<httpRuntime maxRequestLength="2147483647" />`

Using groovy script to set different additional fields values for each record

To configure customized additional field parameters for cartridges in each run, perform the following:

1. Add the custom fields you want to add in the **CMAdditionalFeilds.xm**l file.

   Or

   In the file available in **System Parameters** > **Database to File** > **Content Manager Options** > **Additional Fields Configuration File Name**.

2. Select **AdditionalFields** in the Content Manager and create the same fields that were added in the above step.

3. Add these properties to the record, which is defined in SDM.

4. For the business flow to choose the newly added additional fields on Content Manager, add the following tt in your custom groovy script attached to the business flow.

   `import com.outerbay.foundation.services.dao.configuration.ConfigurationSetting;`

   `ConfigurationSetting.clearConfigSettingMap();`

5. After the successful BF, you can see each record with different additional field value in CM.

   An example groovy script to choose the new additional field for each row present in the **obtdiscovery** schema:

```
 import com.outerbay.foundation.components.workflow.WorkflowLauncher
import com.outerbay.foundation.components.api.workflow.instance.JobInstance;
import
com.outerbay.foundation.components.api.workflow.instance.JobInstanceStatus;
import com.outerbay.foundation.services.dao.configuration.ConfigurationSetting;
import java.text.*;
import groovy.sql.*
String query = """SELECT table_name,column_name FROM obtdiscovery.fields where
project_name=? AND sensitive_flag<>0 AND reviewed_flag<>0""";
def rows = INTF_DB.rows(query,[p_projectname]);
WorkflowLauncher workflowLauncher = new WorkflowLauncher();
// default BF set of parameters (runs independent of OS)
def arglist = ["-s", "launch_businessflow", "-f", "erj", "-e", ENVIRONMENT_
NAME, "-j" ] as List
String bfPrefix = "BF-##ENVIRONMENT##@";
String bf = "exportTableColumn_BF"; // Change BF name
arglist.add(bfPrefix + bf);
// Add auto recover parameter if not specified, process would try launch a new
job not recover any failed jobs
arglist.add("-r");
// Add parameters
//SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd@hh:mm:ss");
//String date2 = sdf.format(new Date());
rows.each() { row →
def arglist2 = []
arglist.each() { arglist2.add(it) }
arglist2.add("p_columnname="+row.column_name);
arglist2.add("p_projectname="+p_projectname);
arglist2.add("p_tablename="+row.table_name);
arglist2.add("destLocation="+p_cmlocation);
println "-"*100
println(arglist2);
println "-"*100
// Launch BF and wait
JobInstance jobInstance = workflowLauncher.launchWithoutRegister(arglist2);
if ( jobInstance.getJobInstanceStatus() != JobInstanceStatus.COMPLETED ){
throw new IllegalStateException("${bf} Job did not complete successfully.
Please check the log for details: "+jobInstance.getJobInstanceStatus());}
ConfigurationSetting.clearConfigSettingMap();
}
```

Prerequisites for Structured Data Manager

For integration with CM, you must perform the steps that follow on your Structured Data Manager installation.

1. Define **Content Manager location**.

   1. Open your web browser and type the Web Console URL provided by the administrator in the address bar.

      The URL should be in the following format: **http://<HostName>:<Port>/webconsole**.

   2. Type your login credentials and click **Login** to access the Web Console.

      The home page displays all of the applications available to you.

   3. In the Web Console tree component, expand the **Environment Management > Environment** node.

   4. Under the expanded **Environment** node , click **Active Environment** > **Locations**.

   5. In the right panel, click **New** to add new location.

   6. On the **Add Location** page, to create new Content Manager location enter the following information:

1. In the **Type** drop down menu, select **Content Manager**.
2. Fill all the mandatory information (as per the tooltip).

> **Note**
>
> When **Rendering Format** is selected:
> - if you are using a `.hdp` file in the **Project File** field, the **DAC Name** field must include the project name along with the DAC name.
>   For example, <project_name\<DAC_name>
> - if you are using the Designer home path in the **Project File** field, only the DAC name is required in the **DAC Name** field.

3. Click **Create**.

2. Set **Content Manager Options**.

   1. In the Web Console tree component, expand the **Tasks** node.

   2. Expand the **System Parameters** node.

   3. Click on **Database to File** node of the System Parameters to be edited.

   4. On the **Parameters -Database to file** page, click on **Content Manager Options** tab.

   5. On the **Content Manager Options** page, enter the following information:

      1. **Additional Fields Configuration File Name**:Sets this property to its default value.
      2. **Additional Fields Profile Name**:Sets this property to its default value.
      3. **Classification**: Set this property with the same value as specified in CM 9.1 or 9.2 and above.
      4. **Container Record Type**: Set this property with the same value as specified in CM 9.1 or 9.2 and above.
      5. **Document Record Type**:Set this property with the same value as created in CM 9.1 or 9.2 and above.
      6. (Optional): **External Reference**, **Notes**, and **Retention Schedule** fields are optional.

> **Note**
>
> - The above values set are global level configurations that can be overridden at the cartridge level.
> - The **AdditionalFields.XML** file can be used to send any user defined Additional fields or Custom Properties. To use this file on the SDM side, you must ensure to create and assign the **Custom Property** to the respective **Record Types** on the Content Manager side.

Accessing Index on Content Manager

The indexes are available on the admin console and web client of the Content Manager.

To access the indexes do the following:

Admin console

1. Open the **Content Manager Admin Console**.

2. In the left-side panel, click **Search for** drop down menu, and select **Records**.

3. Search and select the required data files (.csv, .xml, or .json) or IDX files (for indexing cartridge) from the list.

4. Click the **Rendition** tab in the bottom right corner. The following message is displayed.

   ```
   Viewing this document will trigger an update to the Record, click to confirm displaying the
   document.
   ```

5. Click the above message to view the indexes.

6. Clicking the index link will redirect you to the SDM Webconsole where the DAC contents are displayed.

Webclient

1. Browse to the **Content Manager Webclient**. For example,

   `https://<Host>:<Port>/ContentManager`

2. Search and select the required data files (.csv, .xml, or .json) or IDX files (for indexing cartridge) from the list.

3. Click the ⬜ icon to view the indexes.

4. Clicking the index link will redirect you to the SDM Webconsole where the DAC contents are displayed.

1.4.2. Hadoop Integration

For integration of Structured Data Manager with Hadoop you must perform the following steps.

1. Ensure IP address of server(Hadoop) is added to the host file of the client(SDM) and vice versa.

2. Create the hadoop folder structure on the client site as follows:

   `<hadoop-Installation-dir>/etc/hadoop`

3. Copy `core-site.xml` file from server into client directory

   `<hadoop-Installation-dir>/etc/hadoop`

4. In `outerbay.properties`, which is located in your application `<OBT_HOME>`, you must specify the path to the Hadoop configuration files in the below two Hadoop properties:

   - `##Specifies the hadoop home for client operations`

     - `hadoop.home.dir=<hadoop-Installation-dir>`

   - `##Specifies the location of the hadoop configuration files used by the java client.`

     - `hadoop.config.dir=<hadoop-Installation-dir>/etc/hadoop`

> **Note**
>
> You must use forward slash (/) as the directory separator even on Windows. If you use a backslash (\) in the `hadoop.config.dir` path, Structured Data Manager throws an exception.

1.4.3. Data Privacy and Protection Appliance (SecureData) Integration

Structured Data Manager provides automated information lifecycle management and structured data optimization by relocating inactive data from expensive tier 1 production systems and legacy databases while preserving data integrity and access.

As a company with both the above products in our portfolio, we offer a solution which helps you to comply with GDPR or such a regulatory requirements in an easy manner.

This chapter provides a detailed description to integrate DPP Appliance (SDA) with SDM.

Overview

The SDM is a product which deals with large volume of data and for the intent to secure the data the following features are supported:

1. Identification (unmask/decryption) / de-identification (mask/encryption) of bulk of data

   - This feature is available to SDM administrators only.

   - With reference to SDM, undo business flow uses identification of the data to get the original data.

2. Allow users to access the secured data (for example, reporting) based on the user's access rights, such as to view the masked data (partial data are available in web services only, not for SimpleAPI), full access or no access to the data.

In the given context, the following features of DPP Appliance (SDA) are important:

1. Authentication methods (credentials)

   - Shared secret

   - Username/password (LDAP Server is required)

   - Certificate (LDAP server is required if LDAP group based matching criteria is used for identity authorization rules)

2. Identity

   - Apart from the credentials, user identity plays an important role in generation of a secure key.

   - Consider Two-factor authentication, which allows user to access multiple data based on the security questions presented to the user. If the answers to the security questions matches the credentials selected by the user during the enrollment, then the user's identity is verified which eventually permits the user to access or protect the relevant data.

   - In DPP Appliance (SDA) , an identity is set based on a format where the authorization is made through pattern matching on identity. But to get a unique secure key for encryption/decryption of any given data, the identity must be the same. Also, note that multiple formats can have the same identity.

Visualization of Secured (De-identified) Data

In this section let us visualize the data access to various users based on the access rights on the system.

Let us assume the following users in the system:

User	Access Rights
Admin	Can see all the data
SSNUser	Can see SSN data
CCUser	Can see Credit Card data
SSNCCUser	Can see SSN and Credit Card data

> ⚠️ **Important**
> In the below tables, the data in green is accessible and the data in red is not accessible.

Customer Data in Production Database

CustID	Name	SSN	Credit Card Number
1	Customer1	489-36-8350	4929-3813-3266-4295
2	Customer2	514-14-8905	5370-4638-8881-3020
3	Customer3	690-05-5315	4916-4811-5814-8111
4	Customer4	421-37-1396	4916-4034-9269-8783
5	Customer5	458-02-6124	5299-1561-5689-1938

Customer Data in Archive Database

CustID	Name	SSN	Credit Card Number
1	Customer1	612-20-6832	5293-8502-0071-3058
2	Customer2	300-62-3266	5548-0246-6336-5664
3	Customer3	660-03-8360	4539-5385-7425-5825
4	Customer4	213-46-8915	4916-9766-5240-6147
5	Customer5	449-48-3135	4556-0072-1294-7415

Customer Data from Archive Database for the SSNUser

CustID	Name	SSN	Credit Card Number
1	Customer1	489-36-8350	5293-8502-0071-3058
2	Customer2	514-14-8905	5548-0246-6336-5664
3	Customer3	690-05-5315	4539-5385-7425-5825
4	Customer4	421-37-1396	4916-9766-5240-6147
5	Customer5	458-02-6124	4556-0072-1294-7415

Customer Data in Archive Database for the CCUser

CustID	Name	SSN	Credit Card Number
1	Customer1	612-20-6832	4929-3813-3266-4295
2	Customer2	300-62-3266	5370-4638-8881-3020
3	Customer3	660-03-8360	4916-4811-5814-8111
4	Customer4	213-46-8915	4916-4034-9269-8783
5	Customer5	449-48-3135	5299-1561-5689-1938

Customer Data in Archive Database for the SSNCCUser/Admin

CustID	Name	SSN	Credit Card Number
1	Customer1	489-36-8350	4929-3813-3266-4295
2	Customer2	514-14-8905	5370-4638-8881-3020
3	Customer3	690-05-5315	4916-4811-5814-8111
4	Customer4	421-37-1396	4916-4034-9269-8783
5	Customer5	458-02-6124	5299-1561-5689-1938

Now, it is easy to understand that the key to encrypt the data plays an important role. Therefore, you must have this key based on the data format. In the above example, if the key used for SSN and Credit Card Number are different then you will be able to

achieve data access to the users based on their rights. In case of DPP Appliance (SDA) , the key is generated based on the identity provided by the user. With this, consider the identities per format as shown below:

Format	Identity
Social Security Number	idSSN
Credit Card	idCC
Alpha Numeric	idAlphaNum

Now, create a table with identity and user access rights mapping as shown below:

Identity	Admin	SSNUser	CCUser	SSNCCUser
idSSN	Full	Full	-	Full
idCC	Full	-	Full	Full
idAlphaNum	Full	-	-	-

From the above table, there should be some rules which are to be provided to restrict the identification (of the data) functionality based on the identity and user. This can be done by using identity rules on DPP Appliance (SDA) part.Note that this example will be used in the following sections.

Authentication Methods

DPP Appliance (SDA) uses shared secret, username/password, and certificates as authentication mechanism. With SDM, only the shared secret (in a limited manner) and username/password methods are used along with "LDAP + Shared Secret" mechanism.

DPP Appliance (SDA) Settings

Create the following authentication methods as described on OpenText™ Data Privacy and Protection Appliance *Administrator Guide*. Below are the screenshots for your quick reference:

Authentication Methods

SharedSecret method setting details

Home	Key Management	SSL	Data Protection Settings
District	Key Rotation	▪ Authentication	Policy	Hadoop TDE

Authentication

Edit existing Authentication Method

Method Name *   Shared Secret

Identity Patterns *   id* ❓

IP Addresses *   * * * * ❓

Enabled ☑

Comment   voltage123

* denotes mandatory field.

Authentication Settings

Method Type   Shared Secret ▼

Secret String *   •••••••• ❓

Note: Secrets must be at least 8 characters and contain both letters and numbers.

Confirm Secret *   •••••••• ❓

[Save] [Cancel]

LDAP + SharecSecret method setting details

Home	Key Management	SSL	Data Protection Settings
District	Key Rotation	▪ Authentication	Policy	Hadoop TDE

Authentication

Edit existing Authentication Method

Method Name *   LDAP + Shared Secret

Identity Patterns *   id* ❓

IP Addresses *   * * * * ❓

Enabled ☑

Comment

* denotes mandatory field.

Authentication Settings

Method Type   LDAP ▼

LDAP Resource *   LDAP ▼ ❓

You can manage the list of available LDAP resources from the LDAP Resources page.

Search by Group Name   ☐ ❓

Use LDAP plus shared secret   ☑ ❓

Secret String *   •••••••• ❓

Note: Secrets must be at least 8 characters and contain both letters and numbers.

Confirm Secret *   •••••••• ❓

Authentication Type   Direct shared secret in key request ▼ ❓

[Save] [Cancel]

SDM Settings

The SDM side settings must be done as shown below. Also, refer Masking Server settings for more information on SDM settings.

Default Identity

Format Specific Identity

Username/Password

This method requires LDAP instance. As the identity rules on the DPP Appliance (SDA) side takes LDAP groups into account, therefore make sure to setup the LDAP group such that the access right requirements are met.

LDAP Configuration

Assure to configure the groups such that they can be referred correctly in identity authorization rules of DPP Appliance (SDA). Also, make sure that the identity groups are also formed for DPP Appliance (SDA) to validate the identity for a given format. The DPP Appliance (SDA) needs the identity information for a group in "proxyAddresses" attribute and these identity groups can be made parent group of user groups. The following table provides the details on users under a specific user group for the example data:

Group	Group Name	Users
Admin	G-Admin	Admin
Social Security Number	G-SSN	SSNUser, SSNCCUser
Credit Card	G-CC	CCUser, SSNCCUser
OnlySDM	G-SDM	UserSDM

The following table specifies the identity groups with the user group as child groups:

Group	Group Name	User Groups	proxyAddresses
Social Security Number	G-idSSN	G-Admin, G-SSN	idSSN
Credit Card	G-idCC	G-Admin, G-CC	idCC

Date-Time Formats in SDM

In SDM the DateTime or Timestamp database data types are treated in the same format as Timestamp type. The internal string representation of the Timestamp is of the format YYYY-MM-DD HH24:MI:SS.nnnnnnnnn (where n representation nanoseconds part). However, while using the same with DPP Appliance (SDA) for encryption/decryption the nanosecond part is ignored.

DPP Appliance (SDA) has ORA-DATE format which is different from SDM internal representation and it does not work. Therefore, you need to create another DATE data type format in DPP Appliance (SDA) to make sure that the Date-Time formats works correctly. You can create the same with the format string as YYYY-MM-DD HH24:MI:SS as shown below:

Format Settings

Edit Date format

Format Name *	SDM-ISO-DATE-TIME
Format String *	YYYY-MM-DD HH24:MI:SS
Minimum Year *	1900
Maximum Year *	2023
Data Protection Type *	FPE: Format-Preserving Encryption v2
Comment	Date between years 1900 and 2023

* denotes mandatory field.

Navigate to the Format Mask Settings page under Web Service to specify the mask settings when using the masked parameter in the Web Service APIs for this format.

Save   Cancel

> **⚠ Important**
>
> - In SDM, except for Oracle Database, the internal representation of the DATE data type is YYYY-MM-DD. Therefore, you need to create the same DATE data type format in DPP Appliance (SDA) to ensure that the Date format works correctly. As shown in the below figure, you can create a YYYY-MM-DD format string.

Format Settings

Edit Date format

Format Name *   SDM-ISO-DATE

Format String *   YYYY-MM-DD

Minimum Year *   1900

Maximum Year *   2023

Data Protection Type *   FPE: Format-Preserving Encryption v2

Comment   years between 1900 and 2023

* denotes mandatory field.

Navigate to the Format Mask Settings page under Web Service to specify the mask settings when using the masked parameter in the Web Service APIs for this format.

Save   Cancel

> **Note**
>
> Anytime you add any new format to DPP Appliance (SDA) you need to make sure that you restart the SDM Web Console.

Data Access Cartridge

The de-identified data columns require pseudo column if the data is to be identified (decrypted). This identified data will be available to the user based on the privileges set in DPP Appliance (SDA) . In the pseudo column using unmask method with the syntax given below identify the data:

```
unmask(value).byApplying('function').usingProperty(secureData)
```

You need to specify static import as below before using the method:

```
import static com.outerbay.foundation.components.datamasking.DataMasking.unmask
```

Here are some examples:

- ```
  unmask(CUSTOMER.EMAIL.toString())
  .byApplying('SecureData.AUTO')
  .usingProperty(secureData)
  ```
- ```
  unmask(CUSTOMER.DATEOFBIRTH.toTimestamp())
  .byApplying('SecureData.ORA-DATE')
  .usingProperty(secureData)
  ```
- ```
  unmask(CUSTOMER.CREDITCARD.toString())
  .byApplying('SecureData.CC')
  .usingProperty(secureData)
  ```

An example screen shot is given below:
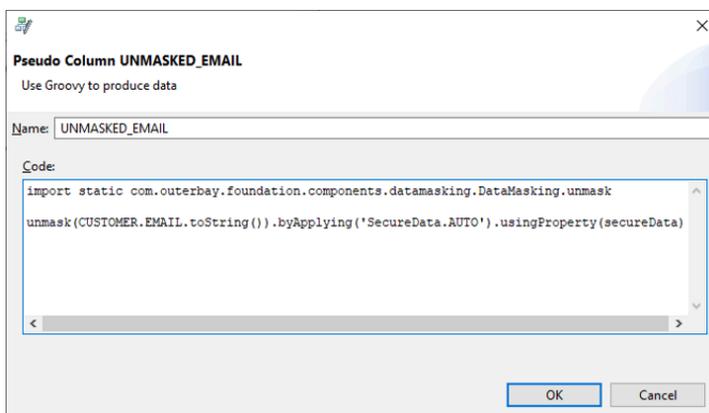
> **Note**
>
> Special characters and spaces in the Pseudo column name are not supported.

Similarly, you can de-identify the unprotected data columns using pseudo columns using mask method with the syntax given below:

```
mask(value).byApplying('function').usingProperty(secureData)
```

You need to specify static import as below before using the method:

```
import static com.outerbay.foundation.components.datamasking.DataMasking.mask
```

Here are some examples:

- ```
  mask(CUSTOMER.EMAIL.toString())
    .byApplying('SecureData.AUTO')
    .usingProperty(secureData)
  ```
- ```
  mask(CUSTOMER.DATEOFBIRTH.toTimestamp())
    .byApplying('SecureData.ORA-DATE')
    .usingProperty(secureData)
  ```
- ```
  mask(CUSTOMER.CREDITCARD.toString())
    .byApplying('SecureData.CC')
    .usingProperty(secureData)
  ```

> **Note**
>
> If any of the authentication, authorization, or format related changes done in the DPP Appliance (SDA) are not shown in Web Console, then you must ensure to restart the Web Console. This occurs due to the in-memory cache used by the underlying SimpleAPI.

1.5. Utility

This section provides information about the utilities added to SDM installation folder:

- Copy Jar utility
- Replace Jar utility
- Remove Jar utility
- Enable SSL for WebConsole Utility

1.5.1. Copy Jar utility

This utility copies the jars to the SDM installation folder.

The utility script `copy_jar.bat/sh` accepts the following parameters :

- *<jarName>*.jar or *<jarName>**, the name of the jar to copy.

- *<path>* of the *<CopyJar>*.jar to copy from,

  where *<path>* is the path of the jar to copy and *<CopyJar>* is the name given for jar.

- Optional, copy utility also accepts location options(-all, -jdbc) as parameters:

  - When using **-all** option, it copies the corresponding jars to the following locations:

    - `<install_dir>\obt\lib`

    - `<install_dir>\obt\ui\plugins\noneclipsedependencies`

    - `<install_dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

      > **Example**
      > copy_jar.bat "C:\Users\utility\CopyJar*" -all
      > copy_jar.bat "C:\Users\utility\CopyJar.jar" -all

      > **Tip**
      > When no option is used, it copies the corresponding jars to the same folders as option -all.
      > **Example**:
      > copy_jar.bat "C:\Users\utility\CopyJar.jar"
      > copy_jar.bat "C:\Users\utility\CopyJar*"

  - When using **-jdbc** option, it copies the corresponding jars to the following locations:

    - `<install_dir>\obt\lib`

    - `<install_dir>\obt\lib\jdbc`

    - `<install_dir>\obt\ui\plugins\noneclipsedependencies`

    - `<install_dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

      > **Example**
      > copy_jar.bat "C:\Users\utility\CopyJar*" -jdbc
      > copy_jar.bat "C:\Users\utility\CopyJar.jar" -jdbc

1.5.2. Replace Jar utility

This utility replaces the existing jars with a new jar within the SDM installation folder.

The utility script `replace_jar.bat/sh` expects the following parameters :

- *<jarName>*.jar or *<jarName>**, the name of the existing jar.

  > **Note**
  >
  > **wildcard (*)** can only be used with -all option.

- *<path>* of the *<NewJar>*.jar to replace with,

  where *<path>* is the path of the new jar and *<NewJar>* is the name given for new jar.

- Optional, replace utility also accepts location options(-all, -jdbc) as parameters:
  - When using **-all** option, it replaces the corresponding jars from the following locations:
    - `<SDM Install Dir>\obt\lib`
    - `<SDM Install Dir>\obt\lib\jdbc`
    - `<SDM Install Dir>\obt\ui\plugins\noneclipsedependencies`
    - `<SDM Install Dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

      > **Example**
      > replace_jar.bat sample* "C:\Users\utility\NewJar.jar" -all

  - When using **-jdbc** option, it replaces the corresponding jars from the following locations:
    - `<SDM Install Dir>\obt\lib\jdbc`
    - `<SDM Install Dir>\obt\ui\plugins\noneclipsedependencies`
    - `obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

      > **Example**
      > replace_jar.bat sample.jar "C:\Users\utility\NewJar.jar" -jdbc

  - When no option is used, it replaces the corresponding jars from the following locations:
    - `<SDM Install Dir>\obt\ui\plugins\noneclipsedependencies`
    - `<SDM Install Dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

      > **Example**
      > replace_jar.bat sample.jar "C:\Users\utility\NewJar.jar"

1.5.3. Remove Jar utility

This utility removes the existing jars from the SDM installation folder.

The utility script `remove_jar.bat/sh` accepts the following parameters :

- *<jarName>*.jar or *<jarName>**, the name of the existing jar

  > **Note**
  >
  > **wildcard (*)** can only be used with -all option.

- *<path>* of the *<ExistingJar>*.jar to remove from,

  where *<path>* is the path in the SDM installation folder to remove the jar and *<ExistingJar>* is the name given for jar.

- When using **-all** option, it removes the corresponding jars from the following locations:

  - `<SDM Install Dir>\obt\ui\plugins\noneclipsedependencies`
  - `<SDM Install Dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib`

    > **Example**
    > remove_jar.bat ExistingJar* -all
    > remove_jar.bat ExistingJar.jar -all

- When using **-jdbc** option, it removes the corresponding jars from the following locations:
  - `<SDM Install Dir>\obt\lib\jdbc`
  - `<SDM Install Dir>\obt\ui\plugins\noneclipsedependencies`
  - <SDM Install Dir>\obt\webconsole\apache-tomcat\webapps\WebConsole\WEB-INF\lib

    > **Example**
    > remove_jar.bat ExistingJar.jar -jdbc

> **Note**
> You must manually remove the jars from obt\lib.

1.5.4. Enable SSL for WebConsole Utility

This utility is used to enable SSL for WebConsole using self signed certificate.

> **Tip**
>
> It is recommended, do not use this utility in production environment.

1. Go to `<install_dir>\obt\bin`.

2. Run the utility `enable_ssl_for_wc.bat/sh`.

3. Enter the location of the self-signed certificate in the **Host** field.

4. Update `server.xml` file under `<install_dir>\obt\webconsole\apache-tomcat\conf` with the SSL configuration details when the commands are completed.

1.6. Performance

This section provides information to enhance the performance of OpenText™ Structured Data Manager.

- Connection Pool

1.6.1. Connection Pool

To enable Connection Pooling in SDM

1. Stop the WebConsole and Designer.

2. Go to *<OBTHOME>|config* folder and open the `outerbay.properties` file with a text editor.

3. Search for the property `jdbc.connection.pool` and uncomment that line.

> **Example**
>
> ```
> .
> .
> ## This property is used to enable the connection pooling
> #jdbc.connection.pool=hikariCP
> .
> .
> ```
>
> Remove the # from the property to uncomment.

4. Optional, you can change the default or add new configurations for connection pooling in `hikariCP.properties` file under *<OBTHOME>|config|connectionpool*.

Configuration	Description	Default Value
minimumIdle	To set the minimum number of idle connections pool. It controls how many connections are added to the pool when it runs out of available connections. It sets the minimum number of idle connections that connection pooling should maintain in the pool.	1
maximumPoolSize	To determine the maximum number of connections pool. **Note** It should be less than the Database maximum connections.	100
poolName	It is the unique name to identify each pool. Each pool determines the number of separate connections in it.	hikariPool
prepStmtCacheSize	To configure the size of the prepared statement cache. This cache stores SQL statements prepared by the database connection and can be reused for future executions. This cache aims to improve performance by avoiding the need to recreate prepared statements every time they are executed.	250
cachePrepStmts	To enable or disable the caching of prepared statements and set them to true, prepared statements are cached, which leads to improved performance in certain scenarios.	true

5. Restart the WebConsole and Designer.

1.7. Appendixes

The following is the list of appendixes:

- Configuration and runtime parameters
- Database archiving advanced tasks
- Change encryption keys and passwords
- Scripted product setup and business flow deployment
- Data Privacy and Protection Appliance (SecureData) Integration

1.7.1. Configuration and runtime parameters

The following parameters govern the running of business flows and jobs:

- Database to file configuration parameters
- Database to database configuration parameters
- Indexing configuration parameters
- File naming parameters

1.7.1.1. Database to file configuration parameters

Selecting Database to file on the Parameters page displays the configuration parameters applied to all business flows and jobs that employ database to file archiving.

The configuration parameters are divided into the following sections:

- Core parameters
- Performance parameters
- Validation parameters
- Amazon S3
- Upload options
- AQS cache options
- Content Manager options

1.7.1.1.1. Core parameters

The following parameters are located in the Core section:

Parameter name	Description
Compression algorithm	Specifies the compression algorithm to apply to the XML and CSV files created after running the database to file archive. Valid values are No Compression and GZIP compression. • **No compression** —places the files in the specified directory without compression. • **GZIP compression** —compresses the files into GZIP format and places them in the specified directory. GZIP compression is the default.
Extract file format	Specifies the file format for extracted data. Valid values are: • **XML denormalized** extracts data as XML. • **CSV normalized** extracts data as comma separated values. • **JSON denormalized** extracts data as JavaScript Object Notation. The default format is **CSV normalized**.
Job engine SQL tracing enabled	Turns on SQL tracing for the job engine repository connection.
Preserve temporary files	Saves the temporary files generated when running the job.
Primary key index location	Displays the location where the primary key indices are stored.
Source database location	Specifies the name of the active database.
Storage Retention	Is the retention policy to submit to the storage subsystem when files are stored to the storage system. Enter the number of days to retain the extracted data. If left blank, the retention behavior defaults to that of the storage system.
Unmask data on undo and reload	By default, prevents any data that has been masked from being reloaded into the source database. Set to false if you want to reload masked data instead of the original values.
Unmask data on upload	Set to true to have data unmasked during upload from file to database. The data is unmasked if the mask used during archiving is reversible. If the mask is not reversible, no attempt is made to revert values, and masked values are uploaded as is from the file archive.
Use Data Movement Key to move data	Set to true to use a data movement key when required to identify rows. For Oracle databases, you can set this parameter to false to use ROWID rather than the data movement key. For non-Oracle databases, this parameter has no effect. Because database reorganization may change ROWID values, you should only consider using ROWID when you are certain that it will not change during archive. For more information about ROWID, see your Oracle documentation.

Parameter name	Description
User index location	Displays the location where the user indexes are stored.
Verify Row Counts	Set to true to perform verification of row counts between the current job and its corresponding selection job. Set to false to bypass verification. True is the default value. This parameter applies only to database to database.
Write XSD and Summary even when there are no data files	Set to true to write XSD and summary files even when there are no data files.The default is true.

1.7.1.1.1.1. Core parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the core parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Compression algorithm	COPY_DB_ TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD, WRITE_SUMMARY	Job / Task	
Extract file format	COPY_DB_ TO_BE (D2F)	SEED_COPY_DB_TO_CSV, SEED_CSV2XML	Job / Task	
Job engine SQL tracing enabled	Starts trace for JobEngine Module only. All jobs		Job	BF
Preserve temporary files	COPY_DB_TO_BE (D2F), UPLOAD_BE_TO_DB	SEED_COPY_DB_TO_CSV, SEED_CSV2XML, CSV2XML_CLEANUP, SEED_XML_TO_CSV	Job / Task	
Primary key index location	COPY_DB_TO_BE (D2F), DELETE_FROM_DB (D2F)		Job / Task	
Source database location	All jobs requiring source database			
Storage Retention	COPY_DB_TO_BE (D2F), COPY_BE_TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD, WRITE_SUMMARY	Job / Task	
Unmask data on undo and reload	ARCHIVE_UNDO_SELECTION RELOAD_SELECTION	ARCHIVE_UNDO_SELECTION - Snapshot qualifying HIST-Table rows into Undo-Selection Tables, RELOAD_SELECTION - Snapshot qualifying HISTTable rows into Reload-Selection Tables	Task	BF/Tasks
Unmask data on upload	UPLOAD_BE_TO_DB	UPLOAD_BE_TO_DB	Job / Task	

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Use Data Movement Key to move data	Selection and Archive Jobs	During prepare it uses this parameter to generate selection views. For D2D it uses to decide if copy should populate obt_saved_rowid on history table.	Job	BF
User index location	COPY_DB_TO_BE (D2F)	EXTRACT_USER_INDEXES		
Verify Row Counts	Selections Jobs only	UPDATE_ROWCOUNT _SUMMARY- Update rowcount summary for the run		
Write XSD and Summary even when there are no data files	COPY_DB_TO_BE (D2F)	WRITE_ATT_GROUP_XSD, WRITE_SUMMARY	Job / Task	

The table below lists the Legend Key and the Legend Value.

Table legend

Legend key: Short name of the task.

Legend value: Description of the task.

Legend Key	Legend Value
CREATE_TABLE_INDEXES	Create Indexes on destination tables
CSV2XML	Legend Value: File format conversion
CSV2XML_CLEANUP	File format conversion clean up
DELETE_FROM_DB	Delete archived data from database
EXTRACT_DELETE_INDEXES	Extract delete-index data matching selection tables to delete-index tables
EXTRACT_USER_INDEXES	Extract user-index data matching selection tables to user-index tables
SEED_COPY_DB_TO_CSV	Seed JE threads to copy Indexes and to CSV
SEED_CREATE_INDEXES	Seed JE Threads for creating indexes on destination tables
SEED_CSV2XML	Seed threads for file format conversion
SEED_XML_TO_CSV	Seed JE Threads for Converting XML Data to CSV format.

1.7.1.1.2. Performance parameters

The following parameters are located in the performance section:

Parameter name	Description
Combined delete count	Defines the maximum number of delete statements that will be executed in a single request to the database.
Combined statement count	Defines the maximum number of database statements which can be combined in a single query.
Data movement Batch size	Controls the number of driving table rows per transaction. This parameter is used for data movement operations on related parent and child rows in the same transaction, which includes database to XML file operations. The total number of rows operated on can be much larger that the value entered, and depends on the characteristics of the data. A value of zero disables intermediate commits.
Eligibility Analytics Configuration	By default, eligibility analytics is disabled. If you want to enable eligibility analytics for a business flow, you must enable it before you run the business flow. Enabling eligibility analytics allows querying of the analytics tables for information on record eligibility. • Select **Disable the eligibility analytics** to improve performance • Select **Enable the eligibility analytics** to enable eligibility analytics. **Note** If you select this option, you must include an interrupt step in your business flow to view the data.
Maximum number of parallel workers	Defines the default maximum number of job workers for tasks that can take advantage of parallelism.
Populate Index Tables	Specifies whether the index tables should be populated. If you plan to delete your data rather than copy it, the value must be set to true.
Populate User-Index tables	Specifies whether the user index tables should be populated. If you plan to delete your data rather than copy it, the value must be set to true.
Selection batch size	Defines the number of driving table rows per transaction. This is used for selection operations that select related parent and child row IDs from the source database into the selection tables.
Units of work	Defines the number of units amongst which the total amount of work will be divided. Each worker picks up a whole unit at a time to ensure clear progress indication and manage the total work in units for the job engine.

1.7.1.1.2.1. Performance parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the performance parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job Pending	Task	Recovery	Jobs/Tasks
Combined delete count		DeleteByArchivedData FromDatabaseTask	Job	Tasks
Combined statement count	COPY_BE_TO_DB, UPLOAD_BE_TO_DB	COPY_BE_TO_DB, UPLOAD_BE_TO_DB	Job / Task	
Data movement Batch size	COPY_DB_TO_BE (D2F)	SEED_COPY_DB_TO_C SV		
Eligibility Analytics Co nfiguration	Selections Jobs only			
Maximum number of parallel workers	COPY_DB_TO_BE (D2F), DELETE_FROM_DB, UPLOAD_BE_TO_DB	COPY_DB_TO_CSV, CSV2XML, DELETE_FROM_DB, COPY_CSV_TO_DB	Job / Task	
Populate Index Tables	COPY_DB_TO_BE (D2F)	EXTRACT_DELETE_IN DEXES	Job / Task	
Populate User-Index tables	COPY_DB_TO_BE (D2F)	EXTRACT_USER_INDE XES	Job / Task	
Selection batch size	Selections Jobs only			
Units of work				

The table below lists the Legend Key and the Legend Value.

Table legend

Legend key:: Short name of the task.

Legend value: Description of the task.

Legend Key	Legend Value
VALIDATE_DATA_UNCHANGED	Validate data unchanged since archival
WRITE_ATT_GROUP_XSD	Write XSD file for archived files
WRITE_SUMMARY	Write summary XML and XSD files for archive

1.7.1.1.3. Validation parameters

The following parameters are located in the validation section:

Parameter name	Description
Check archive consistency	Indicates whether to validate that the archive as a whole is consistent.
Check cardinality constraints	Indicates whether to validate that the extracted data does not violate cardinality constraints in the model instance definition.
Checksum algorithm	Indicates whether to run the checksum algorithm on created files.The supported algorithms are MD5 and SHA-256.
Data unchanged validation	Indicates whether to validate that the data to be deleted has not changed on the source database.
Delete Count Validation	Indicates whether to verify the number of rows to be deleted against the expected number of rows.
Match rowcounts	Indicates whether to verify that rowcounts in the XML files match those in the database. All database to file jobs use this parameter to match row counts for the current one with that of the previous per table use.
Verify Cartridge version for DB deletion	Indicates whether to verify that the AppsPack version used during database deletion is the same as the one used during database extraction.
Verify source data on undo and reload	If set to true (default), the job will fail in the following conditions: • The tables are opted for copy only and the records do not exist in the source • The tables are opted for purge and the records exist in the source DB If you do not want the job failure in the above conditions, change this to false
XML Schema validation	Indicates whether to validate that the XML files do not violate their XML schema. **Tip** For database to file archiving, if you have large BLOB, CLOB, LONG, or LONG RAW fields, OpenText recommends leaving the value of the XML schema validation parameter set to false.
XML file checksum validation	Indicates whether to validate that the XML file checksums have not changed.

1.7.1.1.3.1. Validation parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the validation parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Parameter name	Job	Task	Recovery
Check archive consistency	All D2F and D2FLite Jobs	VALIDATE_ARCHIVE_CONSISTENCY- Verify XML archive is self-consistent	Job / Task
Check cardinality constraints	COPY_DB_TO_BE (D2F)	CSV2XML	Job/Task
Checksum algorithm	All D2F and D2FLite Jobs (Except DELETE-Jobs)	VALIDATE_CHECKSUMS - Verify checksums for files	Job / Task
Data unchanged validation	DELETE_FROM_DB (D2F)	SEED_VALIDATE_DATA_UNCHANGED, VALIDATE_DATA_UNCHANGED	Job / Task
Delete Count Validation	DELETE_FROM_DB (D2F)	DELETE_FROM_DB	Job / Task
Match rowcounts	DELETE_FROM_DB (D2F)	UPD_ROWCOUNT_SUMMARY	Job / Task
Verify Cartridge version for DB deletion			
XML Schema validation	All D2F and D2FLite Jobs (Except DELETE-Jobs)	VALIDATE_FILE_CONTENTS - Validate contents of XML files	Job / Task
XML file checksum validation	All D2F and D2FLite Jobs (Except DELETE-Jobs)	VALIDATE_CHECKSUMS - Verify checksums for files	Job / Task

1.7.1.1.4. Amazon S3

The following parameters are located in the Amazon S3 section:

Parameter name	Description
Maximum number of read attempts	Is the number of times you want to try to read a file from Amazon S3 if an exception occurs during the read operation.

1.7.1.1.5. Upload options

The following parameters are located in the Upload Options section:

Parameter name	Description
Enable Add Column	Set to true to add a target column during upload for any source column that does not already have a corresponding target column. Set to false if you do not want to add such columns during upload.
Enable Drop Column	Set to true to drop a target column during upload if it does not exist in the source table. Set to false if you do not want to drop such columns during upload.
Enable Modify Column	Set to true to modify the target column during upload if it is different from the source column. Set to false if you do not want to change the target column during upload.

1.7.1.1.6. AQS cache options

The following parameters are located in the AQS Cache Options section:

Parameter name	Description
Drop AQS cache schema and tables after clearing the cache	Set to true to drop the AQS cache tables after clearing the cache, if no data is found. If no tables are found, drop the schema as well.
Enable AQS cache	Set to true to create the AQS cache at the end of a database to file archive. Set to false if you do not want an AQS cache.
Enable add column to cache tables	Set to true to add a cache column during archive for any source column that does not already have a corresponding cache column. Set to false if you do not want to add such columns during archive.
Enable de duplication check against cache	Set to true to check for duplicate entries in the cache. Set to false to prevent de duplication from running.
Object mapping file for AQS cache	Specifies a mapping file to use when creating the cache.
Target schema name	Choose from the list of values what you want to use as the schema name.

1.7.1.1.7. Content Manager options

> **Note**
>
> - `CMAdditionlField.xml` file can be used to set the Parameter value from OpenText™ Structured Data Manager.
> - `CMAdditionlField.xml` is used to set the Parameter value if the parameter is alreadydefined.
> - Data type Text in RM is mapped to data type `String` in OpenText™ Structured Data Manager.
> - Property **LEVEL** can be added in `CMAdditionlField.xml` file which allows the metadata to go to either container or document or both.
> - The values for **LEVEL** can be 'container', 'document' or 'both'. If **LEVEL** is not defined then we assume the value is 'both'.

The following parameters are located in the Content Manager Options section:

Parameter name	Description
Additional fields configuration file name	Specifies the name of the file containing the additional Content Manager field definitions.
Additional fields profile name	Specifies the name of the Content Manager profile.
Classification	Specifies the Content Manager classification name or UID. > **Note** > For example, if the UID is 26 then the format is ID:26.
Container record type	Specifies the Content Manager container record type name or UID.
Document Record type	Specifies the Content Manager document record type name or UID.
External Reference	Specifies the Content Manager external reference.
Notes	Specifies the Content Manager note.
Retention schedule	Specifies the Content Manager retention schedule. > **Note** > Enter the parameter as Schedule Number instead of Schedule Title.

1.7.1.2. Database to database configuration parameters

Selecting database to database on the Parameters page displays the configuration parameters applied to all business flows and jobs that employ database to database archiving.

The configuration parameters are divided into the following sections:

- Core parameters
- Performance parameters
- Archive access parameters
- Advanced selection parameters
- Distributed instance parameters
- Partition based delete

1.7.1.2.1. Core parameters

The following parameters are located in the core section:

Parameter name	Description
Job Engine SQL tracing enabled	Turns on SQL tracing for the job engine repository connection.
Unmask data on undo and reload	By default, prevents any data that has been masked from being reloaded into the source database. Set to false if you want to reload masked data instead of the original values.
Use Data Movement Key to move data	Set to true to use a data movement key when required to identify rows. For Oracle databases, you can set this parameter to false to use ROWID rather than the data movement key. For non-Oracle databases, this parameter has no effect. Because database reorganization may change ROWID values, you should only consider using ROWID when you are certain that it will not change during archive. For more information about ROWID, see your Oracle documentation.
Verify Row Counts	Set to True to perform verification of row counts between the current job and its corresponding selection job. Set to False to bypass verification. True is the default value.

1.7.1.2.1.1. Core parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the Core parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Job Engine SQL tracing enabled	Starts trace for JobEngine Module only. All jobs	ARCHIVE_UNDO_SELECTION - Snapshot qualifying HIST-Table rows into Undo-Selection Tables, RELOAD_SELECTION - Snapshot qualifying HIST-Table rows into Reload-Selection Tables	Job	BF
Unmask data on undo and reload	ARCHIVE_UNDO_SELECTION RELOAD_SELECTION	ARCHIVE_UNDO_SELECTION - Snapshot qualifying HIST-Table rows into Undo-Selection Tables, RELOAD_SELECTION - Snapshot qualifying HIST-Table rows into Reload-Selection Tables	Task	Tasks
Use Data Movement Key to move data	Selection and Archive Jobs	During prepare it uses this parameter to generate selection views. For D2D it uses to decide if copy should populate obt_saved_ rowid on history table.	Job	BF
Verify Row Counts	Selection and Archive Jobs	During prepare it uses this parameter to generate selection views. For D2D it uses to decide if copy should populate obt_saved_rowid on history table.		

1.7.1.2.2. Performance parameters

The following parameters are located in the performance section:

Parameter name	Description
Analyze Tables	Indicates whether tables must be analyzed and table-statistics gathered. This parameter does not apply to advanced selection.
Compress tables on History	Compresses the history tables when set to True and is only used during first time deployment. This parameter is applicable only to Oracle.
Data movement Batch size	Controls the number of driving table rows per transaction. This parameter is used for data movement operations on related parent and child rows in the same transaction, which includes database to XML file operations. The total number of rows operated on can be much larger than the value entered, and depends on the characteristics of the data. A value of zero disables intermediate commits.
Data movement method	Defines the method to use to move the data. • Fully transactional—Related parent and child rows are copied and deleted in the same transaction. Referential integrity on history is maintained at all times. • Table-Parallel—Related parent and child rows are copied in separate transactions per table. Deletes happen in a separate transaction, but related parent and child rows are deleted in the same transaction. This option allows faster performance, but archive access will temporarily have inconsistent data. This parameter does not apply to advanced selection.
Eligibility Analytics Configuration	By default, eligibility analytics is disabled. If you want to enable eligibility analytics for a business flow, you must enable it before you run the business flow. Enabling eligibility analytics allows querying of the analytics tables for information on record eligibility. • Select **Disable the eligibility analytics** to improve performance • Select **Enable the eligibility analytics** to enable eligibility analytics. **Note** If you select this option, you must include an interrupt step in your business flow to view the data.
Estimate Percentage	Defines the estimate percentage for gathering statistics. This value is passed to DBMS_STATS.GATHER_TABLE_STATS to compute statistics. Enter a number between 0 and 100. This parameter does not apply to advanced selection.
History Index Tablespace Name	(Oracle only) Specifies a separate tablespace for storing indexes.

Parameter name	Description
Maximum number of parallel workers	Defines the number of tasks that can be executed concurrently in an action. This parameter only affects jobs for the product and does not alter the database vendor's parallelism. If the server is exclusively used, the suggestion is 2 times the number of server CPUs. This parameter does not apply to advanced selection.
Minimum driving ATT chain size to process in a single commit for Fully Transactional operation	The driving ATT chain size greater than or equal to this value will be processed in a single commit for Fully Transactional (FT) operation. Only set this parameter when Advance Selection is executed and the driving ATT may be highly skewed because of ATT chaining. Entering the value 0 disables the feature.
Number of driving table rows per commit for Fully Transactional operation	Defines the maximum number of driving table rows per commit for fully transactional operation. A driving table row includes all related child rows. The total number of rows operated on can be much larger than the value entered, and will depend on the characteristics of the data. A value of zero disables intermediate commits. Also controls the commit size for table parallel delete operation, driven by driving table rows.
Number of rows per commit for Table Parallel copy operation	Controls the transaction size for the copy operation for table parallel data movement's copy operation. That is, it will not commit rows greater than the value set for this parameter. A value of zero disables intermediate commits.
Number of tasks to be created for Data Movement	Defines the maximum number of tasks (chunks of work) that will be created for data movement. More than one task allows parallel execution if number of workers is set greater than 1. For table parallel copy task, if the value is less than the total number of tables in a cartridge, then one task will be created per table.
Parallel degree for creating History indexes	Specifies the parallel degree to use for index creation on history.
Selection batch size	Defines the number of driving table rows per transaction. This is used for selection operations that select related parent and child row IDs from the source database into the selection tables.
Type of Snapshots for Lookup tables	This parameter controls the type of snapshot to be created on history for lookup tables. The possible types are: • SNAPSHOT_WITH_HISTORY—Retain the row history for snapshots from previous runs. Eligibility analytics is not available for lookup tables if you select this option. • SNAPSHOT_WITHOUT_HISTORY—Do not retain any row history for snapshots from previous runs. > **Note** > • Snapshots for lookup tables is not supported for bulk archiving. > • When setting this parameter, make sure that you have unique keys on the lookup tables.

Parameter name	Description
Type of indexes to create on History	This parameter controls the type of indexes to re-create on History after archiving the data into History. Select Operate on All to improve the performance of access to your archived data. This parameter does not apply to advanced selection.
Type of indexes to drop on History	This parameter controls the type of indexes to drop on History before archiving the data into History. This parameter does not apply to advanced selection. Dropping indexes can improve the performance for copying data, but the indexes will need to be recreated after the archive is completed. If the quantity of data to be copied is small compared to the quantity of already archived data, this could impact the overall archive performance. Discuss with your database administrator (DBA) to determine whether it is better to drop and recreate the indexes or leave them there for the copy.

1.7.1.2.2.1. Performance parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the performance parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Analyze tables	Archive Job	Determining whether to gather statistics on History tables	Task	Tasks
Compress tables on History (only used during first time deployment).				
Data movement Batch size				
Data movement method	Archive Job	PARALLELIZE_MOVE_ O2H/H2O - Parallelize moving data from source to history and vice versa	Task	BF / Tasks
Eligibility Analytics Configuration	Standard and PDM Selection Job	ARCHIVE_ANALYZE_E LIGIBILITY - Perform eligibility analytics on snapshot rows	Task	BF / Tasks
Estimate Percentage	Archive Job	Determining whether to gather statistics on History tables	Task	BF / Tasks
Maximum number of parallel workers	All Jobs		Job	BF
Number of driving table rows per commit for Fully Transactional operation	Archive and Reload Job	NATIVE_MOVE_O2H/H 2O - Move data from source tables to history tables and vice versa	Task	BF / Tasks
Minimum driving ATT chain size to process in a single commit for Fully Transactional operation	Archive and Reload Job	NATIVE_MOVE_O2H/H 2O - Move data from source tables to history tables and vice versa	Task	BF / Tasks
Number of rows per commit for Table Parallel copy operation	Archive and Reload Job	NATIVE_MOVE_O2H/H 2O - Move data from source tables to history tables and vice versa DELETE_TASK - Delete data from source	Task	BF / Tasks

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Number of tasks to be created for Data Movement	Archive and Reload Job	PARALLELIZE_MOVE_ O2H/H2O - Parallelize moving data from source to history and vice versa	Task	BF / Tasks
Selection batch size	Standard and PDM Selection Job	ARCHIVE_SELECTION -Snapshot qualifying OLTP-Table rows into Archive Selection Tables		
Type of Snapshots for Lookup tables	Archive Job	PARALLELIZE_MOVE_ O2H/H2O - Parallelize moving data from source to history and vice versa	Task	BF / Tasks
Type of indexes to create on History	Archive Job	SEED_CREATE_HISTO RY_INDEX - Determining whether to create indexes on History tables	Task	BF / Tasks
Type of indexes to drop on History	Archive Job	SEED_CREATE_HISTO RY_INDEX - Determining whether to create indexes on History tables	Task	BF / Tasks

1.7.1.2.3. Archive access parameters

The following parameters are located in the archive access section:

Parameter	Description
Add the rowid column to the archive access union view	Indicates whether the rowid column should be added to the archive access union view.
Configuration mode	Specifies the configuration mode for archive access. Valid values are UNION and ARCHIVE_ONLY.
Create synonyms to dependent packages defined with invokers' rights	Specifies whether dependent packages, which are defined with invokers rights should be cloned, or have synonyms created for them. Creating synonyms can result in a faster runtime for the Create Archive Access job and can result in fewer errors. However, it can also cause problems on some applications.
Default optimizer hint for archive access union view	Specifies the default optimizer hint for the archive access union view.
Recreate all managed table dependent objects	Indicates whether to recreate objects in reruns of the create archive access schema job. If set to false, reruns will only recreate objects in the archive access schema that need to be recreated. If set to true, then all objects will be recreated.
Remote Access Database Link	Specifies the name of the database link in the archive access schema that points to the remote database.
Revoke privileges from the archive access schemas	Indicates whether privileges should be revoked from archive access schemas after the archive access run.

1.7.1.2.3.1. Archive Access parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the Archive access parameters.

Archive Access parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the Archive access parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Add the rowid column to the archive access union view	CreateAA	CloneObjects0 - Clone Objects, Level 0	Task	BF / Tasks
Configuration mode	CreateAA	CloneObjects0 - Clone Objects, Level 0	Task	BF / Tasks
Create synonyms to dependent packages defined with invokers' rights	CreateAA	CloneObjects0 - Clone Objects, Level 0	Task	BF / Tasks
Default optimizer hint for archive access union view	CreateAA	CloneObjects0 - Clone Objects, Level 0	Task	BF / Tasks
Recreate all managed table dependent objects	CreateAA	SeedDBCloneTask - Initialize	Task	BF / Tasks
Remote Access Database Link	CreateAA	CloneObjects0 - Clone Objects, Level 0	Task	BF / Tasks
Revoke privileges from the archive access schemas.	CreateAA	RevokePrivileges	Task	BF / Tasks

1.7.1.2.4. Advanced selection parameters

The following parameters are located in the PDM section:

Parameter	Description
Materialize Union Views	Indicates whether to materialize union views that are used for data location purposes. For example, during reload, a union view can be used during the selection process to pull data from both the HIST table and the OLTP table. If set to false, union views are not be materialized. If set to true, union views are materialized.
Run Output Option	This parameter controls how much PDM produces diagnostic output in the PDM server side log.

Selection Parallel Degree	Defines the selection degree of parallelism. Enables parallel processing only if the value is greater than 1. This parallel degree can be overridden at the statement level by putting a hint in `sql-tuning.properties`. For example:
	
	`testenv.ADVANCED_SELECTION...STMT_1.HINT_2=parallel(pc,8)`
	`testenv.ADVANCED_SELECTION...STMT_2.HINT_1=parallel(ins,8)`
	
	It is not possible to handle parallel degree for each table as the same table can appear in different statements and the same statement can work with different tables, too.

Unify MTU Selections	Unify selections in multiple table uses (MTU) into one selection table, and remove duplicate rows. For database to database archiving, the default value of this parameter is true to ensure that rows selected from multiple table uses of the same table are unified and any duplicate rows are removed. If set to false, duplicated rows may be copied into the history table.

1.7.1.2.4.1. Advanced selection parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the advanced selection parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Materialize Union Views	PDM_* and RELOAD_PDM jobs	PDM_MERGE_TO_HIST	Job / Task	
Run Output Option	PDM_* and RELOAD_PDM jobs	Every PDM Task		
Selection Parallel Degree	PDM_* and RELOAD_PDM jobs	PDM_DTAB_SELECTION		BF / Tasks
Unify MTU Selections	PDM_* and RELOAD_PDM jobs	PDM_PROP_SELECTION	Job / Task	BF / Tasks

The table below lists the Legend Key and the Legend Value.

Table legend

Legend key: Short name of the task.

Legend value: Description of the task.

Legend Key	Legend Value
DP_PARALLEL_DEGREE	DP Parallel Degree
MATERIALIZE_UNION_VIEWS	Materialize Union Views
RUN_OUTPUT_OPTION	Run Output Option
UNIFY_MTU_SELECTIONS	Unify MTU Selections

1.7.1.2.5. Distributed instance parameters

The following parameters are located in the distributed instance section:

Parameter name	Description
History DB: SID name	The SID name of the history database.
History DB: TNS name	The TNS name of the history database.
History to OLTP link name	The name of the database link from History to OLTP.
OLTP DB: SID name	The SID name of the OLTP database.
OLTP DB: TNS name	The TNS name of the OLTP database.
OLTP to History link name	The name of the database link from OLTP to History

Distributed instance parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the distributed instance parameters.

Table legend

Usage: Indicates if it is used by a job, task, or deployment process.

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Usage	Job	Task	Recovery	Pending Jobs/Tasks
History DB: SID name	Job / Task	Bulk - Interface to Relocation	COPY_DATAFILES - Copy Datafiles from OLTP Server to HIST Server, IMPORT_TTS_RELOC - Run Oracle import-utility to import data from TTS into RELOC Schema	Task	BF / Tasks
History DB: TNS name	Job / Task / Deployment	Bulk - Interface to Relocation job, Selection, Archive and Reload jobs	**VALIDATE_LA_ENVIRONMENT** - Validate Live archive Environment, **COPY_DATAFILES** - Copy Datafiles from OLTP Server to HIST Server, **IMPORT_TTS_RELOC** - Run Oracle import-utility to import data from TTS into RELOC Schema	Task	BF / Tasks
History to OLTP link name	Deployment				
OLTP DB: SID name	Job / Task	Bulk -Interface to Relocation	COPY_DATAFILES - Copy Datafiles from OLTP Server to HIST Server, EXPORT_TTS_INTF - Run Oracle export-utility to TTS from INTF Schema	Task	BF / Tasks

Parameter name	Usage	Job	Task	Recovery	Pending Jobs/Tasks
OLTP DB: TNS name	Job / Task / Deployment	Bulk - Interface to Relocation job, Selection, Archive and Reload jobs	**VALIDATE_LA_EN VIRONMENT** - Validate Live archive Environment, **COPY_DATAFILE S** - Copy Datafiles from OLTP Server to HIST Server, **EXPORT_TTS_IN TF** - Run Oracle export-utility to TTS from INTF Schema	Task	BF / Tasks
OLTP to History link name	Deployment				

1.7.1.2.6. Partition based delete

The following parameters are located in the partition based delete section:

Parameter name	Description
Enable Foreign Key Management	Set to true if you want to disable all foreign keys during data movement. Set to false if you do not want to disable the foreign keys during data movement.
Partition Based Delete Method	Indicates the method to use for Partition Based Delete, **Exchange Partition with an empty table** or **Truncate partition**.

1.7.1.3. Indexing configuration parameters

Selecting Indexing on the Parameters page displays the configuration parameters applied to all business flows and jobs that employ Indexing.

The configuration parameters contain following section:

Content Manager options

1.7.1.4. File naming parameters

The following file naming parameters are used only with database to file archiving and can be set for each individual cartridge:

Parameter	Description
Group File Prefix	The prefix for the group XML or CSV files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Group File Suffix	The suffix you want to use for the group XML or CSV files.
Group XML Schema File Prefix	The prefix for the group XML schema files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Group XML Schema File Suffix	The suffix you want to use for the group XML schema files.
Schema Mapping File for Upload	The name (with extension) of a schema mapping file you have created to define the mapping to use when uploading from a file to a destination database. See Mapping schemas by cartridge.
Summary File Prefix	The prefix for the summary files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Summary File Suffix	The suffix you want to use for the summary files.
Summary XML Schema File Prefix	The prefix for the summary XML schema files. The default value is <cartridge_name>_, where cartridge_name is the name of the cartridge.
Summary XML Schema File Suffix	The suffix you want to use for the summary XML schema files.

File naming parameter usage

Each parameter may be used differently by the system. For example, different parameters may be used in different jobs and tasks. In addition, some parameters are used in recovery operations while others are not. The table below lists the usage of the file naming parameters.

Table legend

Job: Name of the job it applies to.

Task: Name of the task it applies to.

Recovery: Indicates if it is picked up by a recovery job or a recovery task.

Pending Jobs/Tasks: Indicates if it is picked up by a pending job or a pending task.

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Pending Jobs/TasksGroup File Prefix	COPY_DB_TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD. On WebConsole: **DB_TO_BE_DIRECT** - Extract data from database to backend directly.	Job	BF
Group File Suffix	COPY_DB_TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD. On WebConsole: **DB_TO_BE_DIRECT** - Extract data from database to backend directly	Job	BF
Group XML Schema File Prefix	COPY_DB_TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD On WebConsole: **WRITE_ATT_GROUP_ XSD** - Write XSD file for ATT instance group files	Job	BF
Group XML Schema File Suffix	COPY_DB_TO_BE (D2F)	COPY_DB_TO_CSV, CSV2XML, WRITE_ATT_GROUP_XSD On WebConsole: **WRITE_ATT_GROUP_ XSD** - Write XSD file for ATT instance group files	Job	BF
Summary File Prefix	COPY_DB_TO_BE (D2F)	**WRITE_SUMMARY** - Write summary XML and XSD files for XML archive	Job	BF
Summary File Suffix	COPY_DB_TO_BE (D2F)	**WRITE_SUMMARY** - Write summary XML and XSD files for XML archive	Job	BF

Parameter name	Job	Task	Recovery	Pending Jobs/Tasks
Summary XML Schema File Prefix	COPY_DB_TO_BE (D2F)	**WRITE_SUMMARY** - Write summary XML and XSD files for XML archive	Job	BF
Summary XML Schema File Suffix	COPY_DB_TO_BE (D2F)	**WRITE_SUMMARY** - Write summary XML and XSD files for XML archive	Job	BF

> **Note**
> All these tasks pickup the new value of the parameter, unless they already have the old value. There is no scenario where you may want to change the value of the parameter during recovery.

1.7.2. Database archiving advanced tasks

This appendix covers advanced tasks that are less frequently performed.

- SQL tuning
- Gather eligibility analytics for advanced selection
- Reclaim space in the active database
- Use views for customized reporting
- Tune the archive access layer
- Implement partition-based deletion for Oracle
- Enable SQL trace for Oracle
- Add minimal privilege users for Oracle
- Lock and unlock the repository

1.7.2.1. SQL tuning

OpenText™ Structured Data Manager uses SQL hints, session variables, and custom pre-statement execution tt to allow you to configure the SQL statements in your Oracle, SQL Server, or Sybase installation, and improve the performance of your selection and data movement statements.

About SQL tuning

You can create hints, set session variables, or create custom pre-statement execution tt that affect the selection and data movement statements for your installation.

Each modification is applied to a specific SQL statement for an individual cartridge in a business flow. For example, if the ORDERS_BF business flow contains two cartridges, ORDERS_DOM and ORDERS_INTL, you would have to write separate hints for each cartridge, or use wildcards.

Each cartridge contains the following seed statement files in the cartridge directory:

Seed statement files	Description
seed-selection-statements.xml	Contains SQL statements that apply to the cartridge selection process. Data selection statements are supported for database to database cartridges and database to file cartridges.
seed-dm-sql-statements.xml	Contains SQL statements that apply to the cartridge data movement process. Data movement statements are supported for database to database cartridges.

With wildcards, each modification can affect one or more statements in each seed statement file. The modifications are defined in the `sql_tuning.properties` file, which is stored in the home directory under `config`. See Manage the home directory.

If you are using Advanced Selection, selection statements are created at runtime. If you want to use SQL tuning with Advanced Selection, you should run Advance Selection in diagnostic mode by setting the configuration parameter Run Output Option to SHOW_PLAN to get a list of all selection statements that Advanced Selection will execute along with the associated execution plans for each statement. You can use the diagnostic information to create specific SQL tuning hints.

About statements

When you deploy database to file or database to database archiving, the SQL statements in the seed-selection-statements.xml and seed-dm-sql-statements.xml files are generated. Each statement contains the following information:

- source environment
- context
- appspack_name
- table_identifier
- statement_name

Example

```
<SRC_ENV>Oracle_OLTP</SRC_ENV>
<CONTEXT>OLTP_SELECTION</CONTEXT>
<APPSPACK_NAME>d2d_trans</APPSPACK_NAME>
<TABLE_IDENTIFIER>ORDER_HEADER</TABLE_IDENTIFIER>
<STATEMENT_NAME>INSERT_SELECTION_NO_ANALYTICS
</STATEMENT_NAME>
```

About context values

The following context values are valid:

Context	Description
OLTP_SELECTION	The archive selection step.
ADVANCED_SELECTION	The advanced selection step. (Use this context instead of OLTP_SELECTION when using Advanced Selection.)
ELIGIBILITY_ ANALYTICS	Eligibility analytics step.
NATIVE_MOVE_O2H	Archive copy step. Data is copied from the source (OLTP) database to the target (history) database, and then deleted. For database to database archiving only.
HISTORY_SELECTION	History selection step for reload. For database to database archiving only.
NATIVE_MOVE_H2O	History copy step for reload. Data is copied from the target (history) database to the source (OLTP) database and then deleted. For database to database archiving only.
ARCHIVE_UNDO_ SELECTION	Selection step for archive undo. For database to database archiving only.
NATIVE_COPY_O2I	Copy OLTP to interface step. Data is copied from the OLTP to the interface tables. For database to database archiving bulk archive only.
NATIVE_COPY_R2H	Copy relocation to history step. Data is copied from the relocation tables to the history tables. For database to database archiving bulk archive only.
NATIVE_COPY_O2RT	Copy OTLP to retention step. For database to database archiving bulk archive only.
NATIVE_COPY_RT2O	Copy retention to OLTP step. For database to database archiving bulk archive only.

About statement_name values

Each context has one or more of the following statement names.

Selection statement names

The following statement values are used when inserting into the selection tables:

Statement	Description
INSERT_SELECTION_ANALYTICS	Used when the Eligibility Analytics Configuration configuration parameter in the Web Console is set to "Enable the eligibility analytics". Eligibility Analytics must be enabled in both Designer and the Web Console before eligibility analytics occurs.
INSERT_SELECTION_NO_ANALYTICS	Used when the Eligibility Analytics Configuration configuration parameter in the Web Console is set to "Disable the eligibility analytics".
STMT_x	The format of statement names for Advanced Selection where x is an integer relating to the target statement number. You can obtain the list of statements by running Advanced Selection in diagnostic mode and setting the configuration parameter RUN_OUTPUT_OPTION to SHOW_PLAN.

Copy and move statement names

The following statement values are used when moving data between the source database and the target. Each move operation consists of an INSERT statement and a DELETE statement.

Statement	Description
COPY_TRX_RANGE	Used for fully transactional data moment when RANGE is applicable.
COPY_TRX_NO_RANGE	Used for fully transactional data moment when NO_RANGE is applicable.
COPY_TP_RANGE	Used for table parallel data moment when RANGE is applicable.
COPY_TP_NO_RANGE	Used for table parallel data moment when NO_RANGE is applicable.
DELETE_TP_RANGE	Used for deletion with table parallel when RANGE is applicable.
DELETE_TRX_RANGE	Used for deletion with fully transactional when RANGE is applicable.
DELETE_TP_NO_RANGE	Used for deletion with table parallel when NO_RANGE is applicable.
DELETE_TRX_NO_RANGE	Used for deletion with fully transactional when NO_RANGE is applicable.

NO_RANGE will be executed when it is a single task per driving table (fully transactional) or table (table parallel), and one of the following configuration combinations is set:

Configuration parameter	Required value
Use database parallelism to archive or reload data	TABLE_PARALLEL
Number of rows per commit for Table Parallel copy operation	0

or

Configuration parameter	Required value
Use database parallelism to archive or reload data	FULLY_TRANSACTIONAL
Number of driving table rows per commit for Fully Transactional operation	0

If neither of these two combinations is set, RANGE will be used.

Eligibility analytics statement name

The following statement value is used for eligibility analytics:

Statement	Description
ELIGIBILITY_ANALYTICS	Used for eligibility analytics.

Archive undo statement names

The following statement values are used for the archive undo job:

Statement	Description
CHECK_SELECTION_FOR_ARCHIVE_UNDO	Used for checking if the archive undo selection process is required for a specific table. If the archive undo job failed the last time it was run, some of the tables may have been successfully processed. This statement skips any tables that do not need to be added to the selection tables.
INSERT_SELECTION_FOR_ARCHIVE_UNDO	Used for loading selection tables with information about the rows that need to be processed by the archive undo job.

About appspack_name and table_identifier values

The appspack_name and table_identifier values depend on your cartridge and the tables included in that cartridge. The appspack_name corresponds to the cartridge name designated in Designer, and the table_identifier corresponds to the table alias designated in the model in Designer.

See the seed-selection-statements.xml and seed-dm-sql-statements.xml files for the valid values.

In Advanced Selection, the table_identifier values must be set using the wildcard operation because specific table name hints are not supported in advanced selection.

About using session variable statements

Session variable statements require different commands depending on the database you are using.

Database	Command
Oracle	Uses the ALTER SESSION command arguments.
SQL Server	Uses the SET command arguments.

Because OpenText™ Structured Data Manager automatically wraps it with the appropriate command (ALTER SESSION or SET), the variable value is just the arguments for the database command.

SQL Server example

For SQL Server, if you want to set the deadlock priority to low, and you would normally change the settings with:

```
SET deadlock_priority low
```

set the value to:

```
deadlock_priority_low
```

Oracle example

For Oracle, if you want to set optimizer index caching to 10, and you would normally change the optimizer settings with:

```
ALTER SESSION set optimizer_index_caching=10
```

set the value to:

```
set optimizer_index_caching=10
```

See Edit the sql_tuning.properties file

About using custom pre-statement execution tt

The PreExec tt enables you to specify custom tt to be run before the selection or data movement statements.

Pre-statement execution tt is run in different locations depending on the type of statement.

Statement	Location
Selection statement	The PreExec tt is run on the source local database.
Copy statement	The PreExec tt is run on the target local database.
Delete statement	For Oracle, the PreExec tt is run on the remote source database. For other databases, the tt is run on the remote target database.

> **Note**
>
> If the statement you are modifying is for a Reload job, the locations are reversed.

Example

```
Oracle_OLTP.OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTION_ANALYTICS.PREEXEC_NAME=apps_initialization
```

where `apps_initialization` is the name of a SQL block that you want to execute.

For advanced selection, the syntax for the session variable is:

```
<src_env>.ADVANCED_SELECTION.<appspack_name>.*.*.SESSION_VAR1=set optimizer_index_cashing = 1
```

where TABLE_IDENTIFIER and STATMENT_NAME are set to wildcards.

Edit the sql_tuning.properties file

Use the `sql_tuning.properties` file to add SQL hints or set session variables.

1. Navigate to the directory under the home directory that contains the seed statements for the cartridge you want to modify. See Manage the home directory.

   For example:

   ```
   /artifacts/businessflow/<busflow_name>/cartridge/<product>/<cartridge_name>
   ```

Parameter	Description
<install_dir>	The location where you installed the software
<business_flow_name>	The name of the business flow that contains the cartridge you want to modify
<product>	The archiving method the cartridge uses: ○ oa—database to file archiving ○ la—database to database archiving
<cartridge_name>	The individual cartridge you want to query

> **Note**
>
> Database to file cartridges only support selection statement hints.

1. Open the seed statement file you want to view.

    ○ seed-selection-statements.xml—contains selection statements

    ○ seed-dm-sql-statements.xml—contains data movement statements

    > **Note**
    >
    > Do not edit the seed statement files.

2. Search the seed statement file for the statement you want to modify.

    You need to note the following information:

    ○ source environment

    ○ context

    ○ appspack_name

    ○ table_identifier

    ○ statement_name

    Example

    ```
    <SRC_ENV>Oracle_OLTP</SRC_ENV>
    <CONTEXT>OLTP_SELECTION</CONTEXT>
    <APPSPACK_NAME>d2d_trans</APPSPACK_NAME>
    <TABLE_IDENTIFIER>ORDER_HEADER</TABLE_IDENTIFIER>
    <STATEMENT_NAME>INSERT_SELECTION_NO_ANALYTICS
    </STATEMENT_NAME>
    ```

4. In a text editor, open the `sql_tuning.properties` file, which is stored in the home directory under `config`. See Manage the home directory.

    > **Note**
    >
    > For Oracle, the `sql_tuning.properties` file has default hints. The SQL Server sql_tuning.properties file has no defaults.

    To add a hint

1. Add a new hint using the following format:

`<SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>. <STATEMENT_ NAME>.<hintType>=<hint>`

Parameter	Description
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<hintType>	INSERT_HINT, SELECT_HINT or DELETE_HINT.
<hint>	The SQL hint. In Advanced Selection, the hint name is defined using the following format: HINT_x where x is an integer. You can obtain the hint number from the list of statements produced by running Advanced Selection in diagnostic mode, setting the configuration parameter "RUN_OUTPUT_OPTION" to "SHOW_PLAN".

Example hint

`OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.<hintType>=<hint>`

> 💡 **Tip**
>
> You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<hintType>=<hint>` applies the hint to all statements in all installed cartridges.

Example SQL in seed file

A SQL statement in a seed file looks something like the following sample. The items between the ## symbols are the available hintTypes for that statement.

```
INSERT ##INSERT_HINT## INTO "CUSTOLTP_HIST"."ORD" ("ORDERDATE", "ORDID", "OBT_WF_RUN_ID",
"OBT_ROW_SEQ", "OBT_TIMESTAMP", "OBT_SAVED_ROWID" ) SELECT ##SELECT_HINT## "S"."ORDERDATE",
    "S"."ORDID",
    "S"."OBT_WF_RUN_ID",
    "S"."OBT_ROW_SEQ",
    sysdate,
    "S"."OBT_SAVED_ROWID"
```

```
FROM "OBT_IF"."ORD_SVO"@ORCL3 "S"
WHERE ("S"."OBT_WF_RUN_ID"=?)
AND ("S"."OBT_ROW_SEQ" BETWEEN ? AND ?)
```

2. Fill in the appropriate SQL hint.

   Oracle example

   ```
   Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.SELECT_HINT=use_nl
   ```

   SQL Server example

   ```
   MSOLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.SELECT_HINT=FAST 5
   ```

3. Save the sql_tuning.properties file.

   To add a session variable

4. Add a new session variable using the following format:

   ```
   <SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>.<STATEMENT_ NAME>.<SESSION_VARn>=<var>
   ```

Parameter	Description
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<SESSION_VARn>	The identifier for the session variable.
<var>	The session variable.

   Example

   ```
   Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.<SESSION_VARn>=
   <var>
   ```

   > 💡 **Tip**
   >
   > You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<SESSION_VARn>=<var>`
   > applies the session variable to all statements in all installed cartridges.
   > For example, if the default statement is as follows: `DefEnv.OLTP_SELECTION.*.*.*.INSERT_HINT=APPEND`
   > changing the statement to: `DefEnv.OLTP_SELECTION.INVENTORY.ITEM.*.INSERT_HINT=PARALLEL`
   > will override the APPEND hint for only the ITEM table in INVENTORY.

5. Fill in the appropriate session variable.

Oracle example

```
Oracle_OLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.SESSION_VAR3=set
optimizer_index_caching=10
```

SQL Server example

```
MSOLTP.OLTP_SELECTION.d2d_trans.ORDER_HEADER.INSERT_SELECTION_NO_ANALYTICS.SESSION_VAR3=deadlock_p
low
```

6. Save the sql_tuning.properties file.

To add a preexec statement

7. Add a new pre-statement execution tt using the following format:

```
<SRC_ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>. <STATEMENT_NAME>.PREEXECNAME=
<preexecname>
```

Parameter	Description
<SRC_ENV>	The contents of the <SRC_ENV> tag in the seed statement file.
<CONTEXT>	The contents of the <CONTEXT> tag in the seed statement file.
<APPSPACK_NAME>	The contents of the <APPSPACK_NAME> tag in the seed statement file.
<TABLE_IDENTIFIER>	The contents of the <TABLE_IDENTIFIER> tag in the seed statement file.
<STATEMENT_NAME>	The contents of the <STATEMENT_NAME> tag in the seed statement file.
<preexecname>	The name of the custom tt you want to execute.

Example

```
Oracle_OLTP.OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTION_ANALYTICS.PREEXEC_NAME=apps_initializat
```

where `apps_initialization` is the name of the custom tt you want to execute.

> 💡 **Tip**
>
> You can use wildcards to apply the changes to more than one statement. `*.*.*.*.<PREEXEC_NAME>=`
> `<preexecname>` applies the modification to all statements in all installed cartridges.

8. Save the `sql_tuning.properties` file.

Specify SQL hints for non-intrusive environments

In the sql_tuning.properties file, you can add hints for optimizing particular SQL statements. You can specify hints for driving statements and cursor loops.

Hints for driving statements have the following syntax:

```
<env_name>.OLTP_LITE_SELECTION.<cartridge>.<table_alias>.
DRIVING_STATEMENT.<hint_type>=<hint>
```

Hints for cursor loops have the following syntax:

```
<env_name>.OLTP_LITE_SELECTION.<cartridge>.<table_alias>.
CURSOR_LOOP.<hint_type>=<hint>
```

> **Note**
>
> You should understand how your particular database handles hints before implementing them in your sql_tunning.properties file.

Parameter	Description
<env_name>	The database that the hints is being applied to. For example, SQL Server or Oracle.
<cartridge>	The name of the cartridge. You can specify * to apply the hint to all cartridges.
<table_alias>	The table to apply the hint to. You can specify * to apply the hint to all tables.
<hint_type>	The hint type specifies where in the SQL statement to insert the hint. The possible values are: • PREFIX-HINT—The hint is at the beginning the SQL statement before the SELECT clause. • SELECT_HINT—The hint is placed after SELECT in the SQL statement. • TABLE_ALIAS_HINT—The hint is placed after each table alias in the FROM clause. • SUFFIX_HINT—The hint is placed at the end of the SQL statement.
<hint>	In Advanced Selection, the hint name is defined using the following format: HINT_x where x is an integer. You can obtain the hint number from the list of statements produced by running Advanced Selection in diagnostic mode, setting the configuration parameter "RUN_OUTPUT_OPTION" to "SHOW_PLAN".

Specify SQL hints for SQL Server D2D environments

In the sql_tuning.properties file, you can add hints for optimizing particular SQL statements.

Hints can be applied with the following syntax:

```
<SRC ENV>.<CONTEXT>.<APPSPACK_NAME>.<TABLE_IDENTIFIER>.<STATEMENT_NAME>.<hintType>=<hint>
```

> **Note**
>
> - SQL hints on SQL Server D2D environment is supported only when the system parameter `Data movement method` is set to default, `Move data using table parallelism`
>
> - You should understand how your particular database handles hints before implementing them in your sql_tunning.properties file.

Parameter	Description
<SRC ENV>	The name of the environment where hints are to be applied.
<CONTEXT>	Data movement identifier. Example: `D2F_COPY_DB_TO_BE`. You can specify * to apply the hint to all data movements.
<APPSPACK_NAME>	The name of the cartridge. You can specify * to apply the hint to all cartridges.
<TABLE_IDENTIFIER>	The name of the table to apply the hints to. You can specify * to apply the hint to all tables.
<STATEMENT_NAME>	Name of a particular statement on which hints must be applied. you can specify * to apply the hint to all statement under the specified context.
<hintType>	The hint type specifies where in the SQL statement to insert the hint. The possible values are: • INSERT_HINT - A query hint, which is placed after INSERT in the SQL statement. • SELECT_HINT - A query hint, which is placed after SELECT query in the SQL statement. • DELETE_HINT - A query hint, which is placed after DELETE query in the SQL statement. • TABLE_HINT - A table hint which is placed after the table to which hints applied to. Based on the <STATEMENT_NAME>, this parameter decides on which query the hints are to be applied. Example: `<SRC ENV>.*.*.*.DELETE_TP_RANGE.TABLE_HINT=ROWLOCK`. Here, the ROWLOCK table hint is applied to the DELETE query and hint is placed after the table name.
<hint>	The particular hint which the user wants to apply.

> **Tip**
>
> Values of the above tags can be inferred from `seed-dm-sql-statements.xml` file which is specific for a specific cartridge. You can find the file in the following location *<OBT_HOME>*`\artifacts|`*<ENV_DIR>*`|businessflow|` *<BUSINESS_FLOW>*`|cartridge|`*<CARTRIGDE_TYPE>*`|`*<CARTRIDGE_NAME>*

Specify SQL hints for advanced selection

In the `sql_tuning.properties` file, if you have not customized the default hints for advanced selection, you should comment them out with the # character as shown. You need to make this change because the previous default might disallow the Web Console configuration parameter for parallel degree for statements 9, 10, and 11.

> **Note**
>
> Advanced selection does not support the use of wildcards for statement names and hint types. Specific statement names and specific hint types should be defined in order for the hint to be used.

```
#=========================================================
# Default Advanced Selection Hints for Oracle
# Note: Advanced selection does not support the use of
# wildcards for statement names and hint types.
# Specific statement names and specific hint types should be
# defined in order for the hint to be used.
#DefEnv.ADVANCED_SELECTION...STMT_9.HINT_2=parallel(fr_s)
#  parallel(c) parallel(pia)
#DefEnv.ADVANCED_SELECTION...STMT_10.HINT_1=parallel(fr_s)
#  parallel(src) parallel(tgt) parallel(to_s)
#DefEnv.ADVANCED_SELECTION...STMT_11.HINT_2=parallel(c1)
#  parallel(npia) parallel(pia)
```

The parallel hint for advanced selection needs the table alias and the degree, unlike above default statements:

```
#DefEnv.ADVANCED_SELECTION...STMT_11.HINT_2=parallel(c1,8)
#  parallel(npia,8) parallel(pia,8)
```

> **Tip**
>
> You need not redeploy for the hints to take effect.

1.7.2.2. Gather eligibility analytics for advanced selection

When performing eligibility analytics for standard selection, you can follow the steps in Viewing eligibility analytics. For advanced selection, you must perform some additional queries against the advanced selection tables to determine which rows are ineligible for archiving.

To ascertain ineligible rows for advanced selection

1. Find the controller id:
   ```
   select * from pdm_controllers where controller_name =<pdm_cartridge_name>
   ```
2. Find the alias for the driving table:
   ```
   select * from pdm_nodes where controller_id= <controller_id>
   ```
3. Get the node and cartridge exception table:
   ```
   select 'node'||node_id||'_s_'||c.controller_id NODE, 'PDM_CARTRIDGE_EXC_' || c.controller_id
   EXCEPTION
     from pdm_controllers c,
           pdm_nodes n
    where c.controller_id = n.controller_id
      and c.controller_name =<pdm_cartridge_name>
      and n.alias = <alias_name>
   ```
   Assume for the purposes of the example that the driving table for your model is OE_ORDER_HEADERS_ALL.
4. Obtain the total driving table rows:
   ```
   select count(*) from pdm_chains
   ```
5. Obtain the total ineligible driving table rows:
   ```
   select count(*) from pdm_chains where at_id in (select at_id from pdm_invalid_ats)
   ```
6. Find the at_id, if you have a specific header_id, for example, the primary key for OE_ORDER_HEADERS_ALL:
   ```
   select at_id from node<n>_s_<controller_id> a
   where a.header_id = &header_id
   and exists (select 'x' from pdm_invalid_ats b
                 where a.at_id = b.at_id)
   ```
   where node<n>_s_<controller_id> is replaced with the result from step 3.
7. Using &at_id from the results of the previous step and PDM_CARTRIDGE_EXC_<controller_id> from the results of Step 3, find out why certain driving table rows are ineligible:
   ```
   select * from PDM_CARTRIDGE_EXC_<controller_id>
   where at_id = &at_id
   ```
8. Advanced selection populates a table named pdm_table_exc_<controller_id>_<table_id>. Using controller_id from Step 1 and table_id from Step 7, you can query this table to get the actual rows:
   1. Run this query to retrieve obt_saved_rowid
      ```
      select * from pdm_table_exc_<controller_id>_<table_id>
      where row_id in (select row_id
      from PDM_CARTRIDGE_EXC_<controller_id>
      where at_id = &at_id)
      ```
      where &at_id comes from the results of Step 6.
   2. You can now further join to table_name from Step 6 to get the rows:
      ```
      select b.*
      from pdm_table_exc_<controller_id>_<table_id> a, <table_name> b
      where row_id in (select row_id from PDM_CARTRIDGE_EXC_<controller_id>
      where at_id = &at_id) and a.obt_saved_rowid = b.rowid;
      ```
      where &at_id comes from the results of Step 6.

1.7.2.3. Reclaim space in the active database

The amount of disk space necessary for the active database is reduced after the inactive archived data has been segregated and relocated.

After running a business flow, you should perform the following types of tasks using your company's procedures to reclaim space in the active database.

To reclaim space after running the Transactional Archive

Oracle

1. Perform a database reorganization.

2. Perform a datafile reorganization.

SQL Server

1. Use the DBCC SHRINKDATABASE command or DBCC CLEANTABLE command to reclaim space in the database.

   For more information, see the Oracle or SQL Server documentation.

To reclaim space after running Bulk Archive

1. Perform a datafile reorganization.

   The tables in the active database are automatically reorganized as part of the Bulk Archive process.

   For more information, see your Oracle documentation.

1.7.2.4. Use views for customized reporting

OpenText™ Structured Data Manager supports the following views of the product metadata:

View	Definition
OBTWC_BF_RUNS_V	Displays business flow runs, but does not display the detailed steps.
OBTWC_RUN_DETAILS_V	Displays business flow runs and the detailed steps of the business flow.
OBTWC_RUN_PARAMETERS_V	Displays the name and value of each parameter and maps them to an individual job run_id.
OBTWC_ROWCOUNT_DETAILS_V	Display the row count and row count operation for each table.

You can use these views to create customized reports for your installation.

1.7.2.5. Tune the archive access layer

To improve the performance of the archive access layer for data transparency (database to database), OpenText™ Structured Data Manager provides a number of parameters. By adding a logon trigger to the archive access schema, you can set these parameters to tune performance in your environment.

Note

You need to add the logon trigger to any archive access schema used to access the combined data. If you only have one archive access schema, you would add it just to that schema. If you used multiple archive access schemas to combine the data, then you would need to add it to each of those schemas.

Tip

The Oracle E-Business Suite integration kit automatically installs a logon trigger in the archive access schema for you. For PeopleSoft integration kits, you need create the trigger yourself.

Example

You could add a logon trigger similar to the following to all of the archive access schemas from which you are combining data:

```
 CREATE or replace TRIGGER obtva_cust_logon AFTER logon ON apps_aa.schema
begin
declare
procedure execute_sql (p_sql_stmt in varchar2) is
   l_count    number;
   l_c        number;
   l_rc       number;
   begin
      l_c := dbms_sql.open_cursor;
      dbms_sql.parse(l_c, p_sql_stmt, dbms_sql.native);
      l_rc := dbms_sql.execute(l_c);
      dbms_sql.close_cursor(l_c);
   end execute_sql;
BEGIN
execute_sql('ALTER SESSION SET "_PUSH_JOIN_PREDICATE"=TRUE');
execute_sql('ALTER SESSION SET "_COMPLEX_VIEW_MERGING"=true');
execute_sql('ALTER SESSION SET "_UNNEST_SUBQUERIES"=FALSE');
execute_sql('ALTER SESSION SET optimizer_index_cost_adj=100');
execute_sql('ALTER SESSION SET optimizer_index_caching=0');
execute_sql('ALTER SESSION SET "_index_join_enabled"=false');
execute_sql('ALTER SESSION SET "_optimizer_push_pred_cost_based"=false');
execute_sql('ALTER SESSION SET "_optimizer_cost_based_transformation"=on');
execute_sql('ALTER SESSION SET "_push_join_union_view"=true');
end;
END;
/
```

1.7.2.6. Implement partition-based deletion for Oracle

When you create your cartridge in the Designer, you can choose to use partition-based deletion, if the source is Oracle 11c or 12g, and all rows are selected.

- Partition-based deletion has some system parameters that you can set. See Partition Based Delete.

  > **Note**
  >
  > These values must be set prior to running the job.

- The system creates the exchange table, performs the partition exchange, then drops the exchange table. However, if you create the exchange table yourself manually, then you must register it using the API `addXChangeTableMap.groovy`.

  Once the table is registered it will not be dropped after the partition exchange, but it will be truncated.

  To un-register the table, use the API `removeXChangeTableMap.groovy`.

- Oracle provides several options for partition exchange statements, like INCLUDING INDEXES and WITHOUT VALIDATIONS. You can apply such options in `sql_tuning.properties`. See SQL tuning for more information about `sql_tuning.properties`.

  For example, adding the line below, controls the options that should apply during partition exchange, and hence the system will execute:

  ```
  DefaultEnvironment.PBD_H2O.INVENTORY.ORDER.PARTITION_EXCHANGE.STMT_EXTN=INCLUDING INDEXES WITHOUT
  VALIDATION UPDATE GLOBAL INDEXES
  ALTER TABLE <PARTITIONED_TABLE> EXCHANGE PARTITION WITH TABLE <XCHANGETABLE> INCLUDING INDEXES
  WITHOUT VALIDATION UPDATE GLOBAL INDEXES
  ```

- During the deployment, the history table is created with the same partitioning as the corresponding source table. However, you can recreate the history table and indexes manually, if any extra storage attribute needs customization after the deployment or, before the deployment, when the history schema mapping is set using the API `addHistoryOwnerMapping.groovy` prior to deployment.

- If rows are selected from more than one partition, a partition-based delete is performed on all the qualifying partitions.

- After the partition exchange, any unusable global indexes are rebuilt and all local indexes are rebuilt on the empty partition to ensure the indexes are in valid state.

The only valid data movement for partition-based delete is Table Parallelism.

If you set the **Data Movement Method** parameter to **Move Data in a Fully Transactional manner**, that setting is ignored by the job. For Reload, the run uses the fully transactional method for all tables except those marked for partition-based delete, where the management of foreign keys comes into the play.

1.7.2.7. Enable SQL trace for Oracle

You can enable or disable Oracle SQL trace using properties files. Because SQL trace can impact performance, OpenText recommends that you only enable it for performance tuning.

- To enable SQL tracing for the database to database archiving and database to file selection, use the sql_tuning.properties file.

- To enable SQL tracing for database to file data movement, use the outerbay.properties file.

The trace files are saved to the user_dump_dest directory. For more information, see your Oracle documentation.

Enable SQL trace using the sql_tuning.properties file

Use the `sql_tuning.properties` file to enable SQL trace for database to database selection and data movement, and database to file selection.

1. Navigate to `config` under the home directory. See Manage the home directory.

2. Open the `sql_tuning.properties` file. See SQL tuning.

1. Add a session variable to enable SQL trace.

   Example

   SESSION_VAR1=set events '10053 trace name context forever, level 1'

   The trace begins before the statement that the session variable is attached to.

   For example, if you attach it to the following statement:

   OLTP_SELECTION.INVENTORY.ITEM.INSERT_SELECTION_ANALYTICS.SESSION_VAR1=set events '10053 trace name context forever, level 1'

   the trace command is issued before the OLTP_SELECTION.INVENTORY.ITEM. INSERT_SELECTION_ANALYTICS SQL statement is run.

   Enabling SQL trace using the outerbay.properties file

   Use the outerbay.properties file to enable SQL trace for database to file data movement.

2. Navigate to the `outerbay.properties` file, which is stored in the home directory under `config`. See Manage the home directory.

3. Open the `outerbay.properties` file with a text editor.

4. Locate the SQL trace section.

5. Update the properties file. By default, the following properties are set:

   DASL.sqlTrace.level=1
   DASL.sqlTrace.components=ALL

6. Save the outerbay.properties file.

1.7.2.8. Add minimal privilege users for Oracle

An ARCHADMIN user is a user that has the minimum set of privileges for performing archive tasks. You create an ARCHADMIN user when you want to avoid having an archive administrator that has a database administrator role, making sure that data security is not compromised. The ARCHADMIN user can only deploy business flows. This user cannot be used for:

- Installing repositories

- Installing or uninstalling environments

- JDBC connections

- Databases other than Oracle

- Packaged applications (such as PeopleSoft)

The ARCHADMIN user is created as either a source ARCHADMIN user or target ARCHADMIN user. Source ARCHADMIN users are created in the source database (OLTP). Target ARCHADMIN users are created in the target database (History). The target database ARCHADMIN user is required for distributed instance configurations.

The script or scripts that you run to create an ARCHADMIN user depend on the configuration for cartridge deployment. The scripts support the following configurations:

- Database to file

- Standard selection

- Advanced selection

- Database to database

- Standard SI and DI

- Advanced selection DI

- Advanced PDM SI and DI

- Advanced bulk archive

> **Note**
>
> An ARCHADMIN user does not support deployment of cartridges for database to file in non-intrusive environments.

To create an ARCHADMIN user

1. Open a command window by selecting **Start>Run**, typing `cmd`, then clicking **OK**.

2. Change to the directory to *<install_dir>*`/obt/foundation/util/sql/oracle`.

3. Connect to SQL*Plus with the database administrator user having the DBA role.

4. From the SQL*Plus command line, run the appropriate script or scripts based on your configuration for cartridge deployment.

Standard Selection Configurations	Script
Database to file archive	obt_source_admin_user_d2f_bf.sql
Transactional data movement archive	obt_source_admin_user_d2d_bf.sql obt_target_admin_user_d2d_bf.sql
Transactional data movement reloads	obt_source_admin_user_d2d_bf.sql obt_target_admin_user_d2d_bf.sql

Advanced Selection Configurations	Script
Transactional data movement archive	obt_source_admin_user_d2d_bf.sql obt_target_admin_user_d2d_bf.sql
Transactional data movement reload	obt_source_admin_user_d2d_bf.sql obt_target_admin_user_d2d_bf.sql

Upload Configuration	Script
File to Database upload	obt_source_admin_user_f2d_upload.sql

1. At the prompts enter the user name for the ARCHADMIN user, source or target administrator user password, and tablespace name.

> **Note**
>
> You can have more than one ARCHIVEADMIN user in the source and target databases. However, each user must have a different name. For example, ARCHADMIN_1, ARCHADMIN_2, and so on.

2. Exit SQL*Plus.

3. Deploy your business flow using the ARCHADMIN user name and password for the source database administrator. For distributed instances, use the ARCHADMIN user name and password for the target database administrator. See the Web Console online help and Deploying and running business flows for more information about deploying business flows in the Web Console.

1.7.2.9. Lock and unlock the repository

In some situations, you may need to lock the repository. When you lock the repository, you can:

- prevent business flows and cartridges from being deployed in all environments.

- prevent business flows and cartridges from being deployed in a particular environment.

- prevent any new environments from being created.

- prevent any existing environments from being deleted.

For example, you may want to stop all deployments while you perform some system maintenance or upgrade activities. After you are ready, you can then unlock the repository to allow business flows and cartridges to again be deployed.

> **Note**
>
> All new environments are unlocked by default.

Lock the repository

To lock the repository

> **Note**
>
> If a business flow or cartridge deployment is already in process, locking the repository has no effect on that deployment.

1. Navigate to the *<install_dir>*/obt/bin directory.

   where, <install_dir> is the location where you installed the software.

2. Run the lock_repository script using the appropriate syntax:

Platform	Syntax
UNIX	./lock_repository.sh [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]
Windows	lock_repository.bat [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]

Parameter	Description
lock_mode	Optional. If you want to specify a lock mode, enter NO_CREATE_REPOS. This lock mode prevents any environments from being created or deleted.
environment_name	Optional. The name of the environment.
yourName	The name of the person locking the repository. Used for auditing purposes only. If name contains spaces, you need to use double-quotes around the name.
comment	Optional. The reason why the repository is being locked.

3. Enter the encryption key at the prompt.

Windows examples

- To prevent business flows and cartridges from being deployed in all environments:

  ```
  lock_repository.bat -u Joe
  ```

- To prevent business flows and cartridges from being deployed in the environment Oracle_Env:

  ```
  lock_repository.bat -e Oracle_Env -u "Joe Smith"
  ```

- To prevent new environments from being created:

  ```
  lock_repository.bat -m NO_CREATE_ENV -u Paul -c "No new environment creation"
  ```

Unlock the repository

To unlock the repository

1. Navigate to the *<install_dir>*/obt/bin directory.
   where, *<install_dir>* is the location where you installed the software.
2. Run the unlock_repository script using the appropriate syntax:

Platform	Syntax
UNIX	./unlock_repository.sh [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]
Windows	unlock_repository.bat [-m <lock_mode>] [-e <environment_name>] -u <yourName> [-c <comment>]

Parameter	Description
lock_mode	Optional. If you want to specify a lock mode, enter NO_CREATE_REPOS. This lock mode enables environment creation and deletion in the repository.
environment_name	Optional. The name of the environment.
yourName	The name of the person unlocking the repository. Used for auditing purposes only. If name contains spaces, you need to use double-quotes around the name.
comment	Optional. The reason why the repository is being unlocked.

The script prompts for the encryption key.

3. Enter the encryption key at the prompt.
   Windows examples

- To unlock all environments:
  ```
  unlock_repository.bat -u Paul -c "Unlock to allow check-ins"
  ```

- To unlock just the environment Oracle_Env, without affecting any other locked environments:

  ```
  unlock_repository.bat -e Oracle_Env -u Mike -c "Unlock Oracle_Env"
  ```

- To allow new environments to be created:

```
unlock_repository.bat -m NO_CREATE_ENV -u Paul -c "Allow environment creation"
```

1.7.3. Change encryption keys and passwords

The password manager utility is used to change the encryption key and repository password.

- Change the encryption key

- Reset the encryption key

- Change the repository password

- Update database location passwords

Change the encryption key

Use password manager to change the encryption key for the environment.

To change the encryption key

1. Open a command window.

2. Run the utility using the following syntax where, <install_dir> is the location where you installed the software:

Platform	Syntax
UNIX	<install_dir>/obt/bin/runPwManager.sh -m
Windows	<install_dir>/obt/bin/runPwManager.bat -m

3. The following message is displayed:

    Change the master encryption key for the stored passwords: If you change the password in the
    database for OBTREP_DB, then you must use this utility with the -p flag to properly reset the
    password before using the -m flag.
    - Enter CURRENT encryption key:
    - Re-enter CURRENT encryption key:
    - Enter NEW encryption key:
    - Re-enter NEW encryption key:

    All encrypted passwords in the connection-sources.xml file are re-encrypted using the new key. The encryption key
    values are not displayed on the window.

Reset the encryption key

Use password manager to reset a lost encryption key for the environment.

To reset the encryption key

1. Open a command window.

2. Run the utility using the following syntax where, <install_dir> is the location where you installed the software.:

Platform	Syntax
UNIX	<install_dir>/obt/bin/runPwManager.sh -s (environment name)
Windows	<install_dir>/obt/bin/runPwManager.bat -s (environment name)

3. The following message is displayed:

   ```
   Reset the encryption key for an environment.
   You will be prompted to provide passwords for the locations created in this environment.
   Do you want to proceed to reset the encryption key of environment sql_int? (yes/no)
   ```
   If you enter **yes**, then provide the following details:
   - Enter the new encryption key:
   - Enter the repository database admin password:

4. You will then be prompted for passwords for all of the database locations for all of the environments associated with the repository. If a location cannot be reached, you are prompted to skip the location, re-enter the password, or quit the process.

All encrypted passwords in the `connection-sources.xml` file are re-encrypted using the new key. The encryption key values are not displayed on the window.

Change the repository password

If you need to change the password for your repository, use the password manager utility to update the password in the `connection_sources.xml` file. recommends using the Web Console to modify location passwords for all other databases. For more information, see Change locations and update passwords.

> **Note**
> You can use the password manager utility when required for automation purposes.

To change the password for the repository owner on the database

1. Ensure the encryption key has not been changed.

   The master encryption key is required for verification purposes. You can change the encryption key after completing the steps in this section.

2. Open a command window.

3. Run the password manager utility using the following syntax:

Platform	Syntax
UNIX	<install_dir>/obt/bin/runPwManager.sh -p
Windows	<install_dir>/obt/bin/runPwManager.bat -p

where, <install_dir> is the location where you installed the software.

4. Enter values for the prompts. Enter a question mark (?) to see a list of available values.

   - Enter environment to modify:

   - Enter master encryption key:

   - Re-enter master encryption key:

   - Enter location name to modify:

     > **Tip**
     > You should enter OBTREP_DB for location.

   - Enter NEW password:

   - Re-enter NEW password:

The password is changed, and the values saved to `connection-sources.xml` in `config` under the home directory. See Manage the home directory.

Update database location passwords

When you create a standard environment, creates the necessary schema or database depending upon your underlying database (Oracle, SQL Server, or DB2). At the same time, it assigns passwords to those newly created objects. These objects and their passwords are required for to perform its various operations on your source and target databases.

> **Note**
>
> For non-intrusive environments, creates no schema or databases, and therefore requires no passwords for them.

When you create a new, standard environment in Oracle or SQL Server, creates the necessary schema or database, and assigns the passwords specified by you. When you create a new environment in DB2, you must provide the necessary operating system user password required to perform operations against the database.

stores the current passwords for the schema or databases at the time of standard environment creation because it needs them to perform its operations.

Other than the initial setting of the schema or database passwords for Oracle or SQL Server database location passwords are managed and changed through the facilities of the underlying database, not . Hence, when a schema or database password is changed on the database side, you must update the password stored by . For example, suppose that your company requires all schema passwords be changed every other month. When your OBT_IF schema password changes in your Oracle database, you must update the OBT_IF password that stores. Otherwise, it will not be able to perform operations against that location.

You can update database location passwords in one of two ways:

- From the Web Console. For more information, see Change locations and update passwords.

- From the command line using the password manager utility:

  `runPwManager.bat|sh -p <newpasswd> -l <location> -e <env> -k <encryption>`

  where:

  <location> is one of the following:

- OLTP_DB is the source database.

- OBTINTF_DB is the interface schema (OBT_IF).

- ARCHIVE_ACCESS_DB is the archive access schema (OBT_BUILD_AA).

- HISTORY_DB is the target database.

- HISTRELOC_DB is the relocation schema (OBT_RELOC).

<env> is the environment for which you are changing passwords.

<encryption> is the encryption key.

1.7.4. Scripted product setup and business flow deployment

If you plan to set up and deploy the OpenText™ Structured Data Manager multiple times, you can use scripted deployment. See the *OpenText™ Structured Data Manager Installation Guide* for information about scripted installation of the product.

Understanding properties files

Scripted deployment requires that you create the following properties files to define the installation.

Property file	Description
repository.install.properties	Contains properties to create your repository.
product.install.properties	Contains properties to create environments with database to file and optionally database to database.
businessflow.install.properties	Contains properties to deploy your business flows.

You can create these files in one of two ways:

- Use the Web Console to generate the properties files.
- Use a text editor to manually create the files.

The instructions in this chapter assume that you have used the Web Console at least once to create the properties files in the install directory.

> **Note**
>
> Properties files should contain property values with ISO 8859-1 character encoding only. For more information on using characters that cannot be expressed using ISO 8859-1 encoding, see Using special characters in properties files.

Use special characters in properties files

Properties files should contain property values with ISO 8859-1 character encoding only. Unitt escapes can be used for characters that cannot be directly represented in this encoding. One u character can be used per escape sequence.

To convert properties files that contain non-ISO 8859-1 character encoding

1. Run the Native-to-ASCII Converter, native2ascii, with the following syntax:

   ```
   native2ascii -encoding <encoding_name> <input_file> <output_file>
   ```

Parameter	Description
<encoding_name>	A valid Java encoding. For more information, see the Sun Java website at http://java.sun.com.
<input_file>	The original properties file with non-ISO 8859-1 character encoding.
<output_file>	The new properties file with only ISO 8859-1 character encoding.

2. Use the new properties file for your scripted installation.

Script the repository setup

To script your repository installation, you need to create a file with the properties you want to use, and then run the script. The following sections contain the scripting procedure, the properties in the properties file, and examples of the properties file.

> **Note**
>
> If you are archiving data from Sybase, you must install your repository on either an Oracle or PostgreSQL database.

Scripted repository installation procedure

If you choose to include passwords or encryption.key properties in the properties file, they are saved in the text file and can be accessed. To ensure that the passwords are not saved, omit the entire password property line from the properties file. You will then be prompted for the passwords.

1. Make sure that the OpenText™ Structured Data Manager has been installed. See the *OpenText™ Structured Data Manager Installation Guide*

2. Navigate to the directory that contains the installation scripts:
   Example
   <install_dir>/obt/install
   where, <install_dir> is the location you installed the software.

3. Open the `repository.install.properties` file in a text editor, or create the file if it does not exist.

4. Edit the properties file as necessary for your installation. See Scripted environment creation properties.

5. Open a command window.

6. Install the repository using the appropriate syntax:

Platform	Syntax
UNIX	`./deploy_repository.sh <properties_file_name>`
Windows	`deploy_repository.bat <properties_file_name>`

   where `properties_file_name` is the name of the properties file you are using for the repository.

Scripted repository installation properties

The following properties are used for installing the repository:

Property name	Expected value
encryption.key	The encryption key to be created. If you omit the line, you will be prompted for the encryption key.
repository.conn.rdbmstype	oracle or postgresql.
repository.conn.dbserver	Database SID or server name for the repository.
repository.conn.host	Host name for the repository.
repository.conn.port	Port number for the repository.
repository.dbadmin.username	Repository database or database server administrator username.
repository.dbadmin.password	Repository database or database server administrator password. If you omit the line, you will be prompted for the password.
repository.owner.name	Repository user to be created by the installer.
repository.owner.password	Repository user password to be created. If you omit the line, you will be prompted for the password.
repository.default.storagelocation	Default tablespace to use when creating repository users. (Oracle only)
repository.temp.storagelocation	Temporary tablespace to use when creating repository users. (Oracle only)
repository.dbname	Repository database name. (PostgreSQL)
repository.default.storagelocation.size	Data device size to be allowed to the repository database in MB. (PostgreSQL)
repository.lot.storagelocation.size	Log device size to be allocated to the repository database in MB. (PostgreSQL)
console.admin.username	The username for the Web Console administrator.

console.admin.passwd	The password for the Web Console administrator.
console.admin.email	Optional. The email address of the Web Console administrator.
console.admin.userRealName	Optional. The real name of the Web Console administrator.
console.admin.phone	Optional. The phone number of the Web Console administrator.
console.admin.description	Optional. The description of the Web Console administrator.

Scripted repository installation properties file example

The following examples demonstrate the use of the properties file.

Oracle

```
encryption.key=myEncryptionKey
repository.conn.rdbmstype=oracle
repository.conn.dbserver=orcl1
repository.conn.host=myHost
repository.conn.port=1521
repository.dbadmin.username=system
repository.dbadmin.password=myPassword
repository.owner.name=obt_rep
repository.owner.password=myPassword
repository.default.storagelocation=USERS
repository.temp.storagelocation=TEMP
console.admin.username=admin
console.admin.passwd=admin
```

Script the environment creation

To script your environment creation, you need to create a file with the properties you want to use, then run the script. The following sections contain the scripting procedure, the properties in the properties file, and examples of the properties file.

Each environment requires a separate product.install.properties file.

Scripted environment creation procedure

The sample product.install.properties file is installed with the OpenText™ Structured Data Manager.

If you choose to include passwords or encryption.key properties in the properties file, they are saved in the text file and can be accessed. To ensure that the passwords are not saved, omit the entire password property line from the properties file. You will then be prompted for the passwords.

> **Note**
>
> Database to file archiving is installed with all environments. Database to database archiving is optional.

1. Make sure that the OpenText™ Structured Data Manager has been installed. See the *OpenText™ Structured Data Manager Installation Guide.*

2. For non-intrusive environments, place the generic JDBC driver in the directory `<install_dir>`/obt/lib/jdbc, where, <install_dir> is the location where you installed the software.

3. Navigate to the directory that contains the installation scripts:

Example

<install_dir>/obt/install

where, <install_dir> is the location you installed the software.

4. Open the product.configure.properties file in a text editor.

5. Edit the properties file as necessary for your installation. See Scripted repository installation properties file example

6. Open a command window.

7. Create the environment and deploy the product using the appropriate syntax:

Platform	Syntax
UNIX	./deploy_product.sh <properties_file_name>
Windows	deploy_product.bat <properties_file_name>

where `properties_file_name` is the name of the properties file you are using.

Scripted environment creation properties

The following properties are used for creating the environment and deploying database to file:

Property name	Expected value
encryption.key	Encryption key.
	If you omit the line, you will be prompted for the encryption key.
environment.id	The name you want to use for the environment.
environment.description	The optional description of the environment.
ea.product.configuration	Specifies whether the environment is standard or non-intrusive. For a non-intrusive environment, set the value to `lite`. Otherwise, set it to default. If you set this property to `lite`, the following properties are ignored because they are not applicable to a non-intrusive environment:
	• user.index.location • source.dbadmin.username • source.dbadmin.password • source.dbadmin.osauth • source.interface.default.storagelocation • source.interface.temp.storagelocation • source.interface.default.storagelocation.size • source.interface.log.storagelocation.size
source.rdbms.name	Acceptable values are oracle, sqlserver, or sybase.
source.conn.dbserver	The name of the Oracle database service, SQL Server or Sybase server name, or DB2 data source for the source database.
source.conn.host	The host for the source database.
source.conn.port	The port number for the source database.
source.dbadmin.username	The source database or database server administrator username.
source.dbadmin.password	The source database or database server administrator password.
	If you omit the line, you will be prompted for the password.
source.interface.dbname	The interface database name. (SQL Server only)
source.interface.default.storagelocation.size	The data device size to be allocated to the interface database in MB. (SQL Server only)

source.interface.log.storagelocation.size	The log device size to be allocated to the interface database in MB. (SQL Server only)
source.interface.owner.name	Interface user to be created by the installer.
source.interface.owner.password	Interface user password. If you omit the line, you will be prompted for the password.
source.interface.default.storagelocation	The default tablespace to use when creating the interface user for the source database. (Oracle only)
source.interface.temp.storagelocation	The temporary tablespace to use when creating the interface user for the source database. (Oracle only)
user.index.location	The location for the user index. Possible values are source or repository.

The following optional properties are used for deploying database to database:

Archive option	Property name	Expected value
	product.configuration	SI or DI
	archiveaccess.owner.name	The archive access user to be created.
	archiveaccess.owner.password	The password for the archive access user. If you omit the line, you will be prompted for the password.
	archiveaccess.location	The location of the archive access user. Acceptable values are OLTP_DB or HISTORY_DB.
	archiveaccess.default.storagelocation	The default tablespace to use when creating the archive access user. (Oracle only)
	archiveaccess.temp.storagelocation	The temporary tablespace to use when creating the archive access user. (Oracle only)
DI	hist.conn.dbserver	The target database server name or SID.
DI	hist.conn.host	The host for the target database.
DI	hist.conn.port	The port number for the target database.
DI	hist.dbadmin.username	The target database server administrator username.
DI	hist.dbadmin.password	The target database server administrator password. If you omit the line, you will be prompted for the password.
DI	hist.reloc.owner.name	Target relocation user (Oracle)
DI	hist.reloc.owner.password	Target relocation password (Oracle). If you omit the line, you will be prompted for the password.

DI	hist.reloc.default.storagelocation	The default tablespace for creating the target relocation user. (Oracle only)
DI	hist.reloc.temp.storagelocation	The temporary tablespace for creating the target relocation user. (Oracle)
DI	hist.reloc.dbname	The target database name. (SQL Server only)
DI	hist.reloc.default.storagelocation.size	The data device size to be allocated to the target relocation user in MB. (SQL Server only)
DI	hist.reloc.log.storagelocation.size	Log device size to be allocated to the target relocation user in MB. (SQL Server only)
DI	histreloc.sourceinterface.dblink.name	The database link name to be created from the target relocation schema to the interface schema. (Oracle only) If GLOBAL_NAMES is set to true, then the database link name must be the same as the source database.
DI	sourceinterface.histreloc.dblink.name	The database link name to be created from the interface schema to the target relocation schema. (Oracle only) If GLOBAL_NAMES is set to true, then the database link name must be the same as the target database name.
DI	source.tnsname	The source TNS name used to create the database link. (Oracle)
DI	hist.tnsname	The target TNS name used to create the database link. (Oracle)

Scripted environment creation properties file example

The following examples demonstrate the use of the properties file.

SQL Server example

SQL Server with database to file only:

```
#Product Information
encryption.key=myEncryptionKey
environment.id=myEnvironment
environment.description=My Default Environment

#Source Database Information
source.rdbms.name=sqlserver
source.conn.dbserver=MSOLTP
source.conn.host=myHost
source.conn.port=5001
source.dbadmin.username=sa
source.dbadmin.password=myPassword

#Interface User Information
source.interface.dbname=obt_if
source.interface.owner.name=obt_if
source.interface.owner.password=myPassword
source.interface.default.storagelocation.size=50
source.interface.log.storagelocation.size=25
```

Oracle example

Oracle with database to database enabled for distributed archive:

```
#Product Information
encryption.key=myEncryptionKey
environment.id=myOracleEnv
environment.description=Oracle Environment

#Source Database Information
source.rdbms.name=oraOLTP
source.conn.dbserver=myServer
source.conn.host=myHost
source.conn.port=1521
source.dbadmin.username=system
source.dbadmin.password=myPassword

#Interface User Information
source.interface.owner.name=OBT_IF
source.interface.owner.password=myPassword
source.interface.default.storagelocation=ADM_DATA
source.interface.temp.storagelocation=TEMP

product.configuration=DI

#Archive Access Information
archiveaccess.owner.name=OBT_BUILD_AA
archiveaccess.owner.password=myPassword
archiveaccess.default.storagelocation=ADM_DATA
archiveaccess.temp.storagelocation=TEMP
archiveaccess.location=OLTP_DB

#Relocation User Information
hist.reloc.owner.name=OBT_RELOC
hist.reloc.owner.password=myPassword
hist.reloc.default.storagelocation=ADM_DATA
hist.reloc.temp.storagelocation=TEMP

#Target Database Information
hist.conn.dbserver=oraHIST
hist.conn.host=myHost
hist.conn.port=1521
hist.dbadmin.username=system
hist.dbadmin.password=myPassword

histreloc.sourceinterface.dblink.name=OBT_HIST_TO_OLTP
sourceinterface.histreloc.dblink.name=OBT_OLTP_TO_HIST

#TNS Names
source.tnsname=oraOLTP
hist.tnsname=oraHIST
```

Sybase example

Sybase with database to file only:

```
#Product Information
encryption.key=myEncryptionKey
environment.id=MySybaseEnv
environment.description=Sybase environment

#Repository Information
rdbms.name=postgresql
repository.dbname=obt_rep
repository.conn.rdbmstype=postgresql
repository.conn.dbserver=xxx
repository.conn.host=localhost
repository.conn.port=1527
repository.dbadmin.username=obt_rep
repository.owner.name=obt_rep

#Source Database Information
source.rdbms.name=sybase
source.conn.dbserver=SYBASE_OLTP125
source.conn.host=myHost
source.conn.port=5000
source.dbadmin.username=sa
source.interface.owner.name=obt_if
source.interface.dbname=obt_if
source.interface.default.storagelocation=my_device
source.interface.log.storagelocation=my_log_device
```

DB2 examples

DB2 with database to database:

```
#Product Information
encryption.key=all4one
environment.id=DefEnvD2DDB2
environment.description=DefEnvDB2
product.configuration=DI
#user.index.location=Source

#Source Database Information
source.rdbms.name=db2
source.conn.dbserver=DB2OLTP
source.conn.host=oltp1
source.conn.port=50000
source.dbadmin.username=DB2ADMIN
source.dbadmin.password=

#Interface User Information
source.interface.owner.name=OBTINTF
source.interface.owner.password=
source.interface.default.schema=OBT_IF
source.interface.default.storagelocation=INTF_DATA
#source.interface.temp.storagelocation=TEMP

#Build Archive Access User Information
archiveaccess.owner.name=OBTAA
archiveaccess.owner.password=
archiveaccess.default.schema=OBT_AA
archiveaccess.default.storagelocation=INTF_DATA
#archiveaccess.temp.storagelocation=TEMP
archiveaccess.location=OLTP_DB
aa.to.remotedb.link=HIST_DB

#Target Database Information
hist.conn.dbserver=DB2HIST
hist.conn.host=oltp1
hist.conn.port=50000
hist.dbadmin.osauth=false
hist.dbadmin.username=DB2ADMIN
hist.dbadmin.password=

#Relocation User Information
hist.reloc.owner.name=OBTRELOC
hist.reloc.owner.password=
hist.reloc.default.schema=OBT_RL
hist.reloc.default.storagelocation=RELOC_DATA
histreloc.sourceinterface.dblink.name=OLTP_DB
sourceinterface.histreloc.dblink.name=HIST_DB

ea.product.configuration=default
```

DB2 with database to file only:

```
#Product Information
encryption.key=all4one
environment.id=DefEnvDB2D2F
environment.description=DefEnvDB2D2F
ea.product.configuration=default
user.index.location=Source

#Source Database Information
source.rdbms.name=db2
source.conn.dbserver=DB2OLTP
source.conn.host=oltp1
source.conn.port=50000
source.dbadmin.username=DB2ADMIN
source.dbadmin.password=

#Interface User Information
source.interface.owner.name=OBTINTF
source.interface.owner.password=
source.interface.default.schema=OBT_IF_D2F
source.interface.default.storagelocation=INTF_DATA
```

Script business flow deployment

To script your business flow deployment, you need to create a file with the properties you want to use, and then run the script. The following sections contain the scripting procedure, the properties that can be used, and an example of the properties file.

Scripted business flow deployment procedure

1. Make sure the repository and environment has been created.

   > **Note**
   >
   > If you plan to use eligibility analytics, you must enable eligibility analytics in the model for your business flow before deploying, and the model must contain a pause after the selection step. For more information see the *OpenText™ Structured Data Manager DevelopersGuide*.

2. Navigate to the directory that contains the generated business flows and the businessflow properties files.

   Business flows located in the home directory under the main businessflow directory can be accessed by all environments.

   Example

   <OBT_HOME>/businessflow

   See Manage the home directory.

   Business flows located in an environment directory can only be accessed by that environment.

   Example

   <OBT_HOME>/businessflow/<environment_name>

3. Make sure that the business flow properties file has the same name as the generated business flow that you want to deploy.

   Example

   If the full name of the .businessflow file in the business flow directory is ORDERS_D2F_BF.1_0_0_0.busflow, the properties file should be called ORDERS_D2F_BF.1_0_0_0.properties.

4. Edit the properties file with the appropriate property values from .

5. Save the file.

6. Open a command window.

7. Navigate to the following directory:

   <install_dir>/obt/install

   where, <install_directory> is the directory where you installed the software.

8. Deploy the business flow using the appropriate command:

Platform	Syntax
UNIX	`./deploy_businessflow.sh <environment_name> <businessflow_full_name>`
Windows	`deploy_businessflow.bat <environment_name> <businessflow_full_name>`

where <businessflow_full_name> is the full name of the business flow you want to deploy, and <environment_name> is the name of the environment.

Example

`deploy_businessflow.bat MyEnvironment ORDERS_D2F_BF.1_0_0_0`

> 💡 **Tip**
>
> If you want to remove the job run history of any existing business flows, add the following command: `-Dforce.job.history.drop=true`

Scripted business flow deployment properties

Name	Values
environment.id	Environment ID
encryption.key	Encryption key. If you omit the line, you will be prompted for the encryption key.
businessflow.install.configuration	Possible values are: • PrepareOnly • DeployOnly • PrepareAndDeploy
rdbms.name	The rdbms.name of the source database. Acceptable values are sqlserver, oracle, or DB2.
source.dbadmin.username	The source database administrative username.
source.dbadmin.password	The password for the source database administrative user.
<cartridge_name>.hist.default.storagelocation	The default target tablespace to use for the specified cartridge. (Oracle only)
<cartridge_name>.hist.temp.storagelocation	Temporary target tablespace to use for the specified cartridge. (Oracle only)
<cartridge_name>.archiveaccess.default.storagelocation	Default archive access tablespace to use for the specified cartridge. (Oracle only)
<cartridge_name>.archiveaccess.temp.storagelocation	Default archive access tablespace to use for the specified cartridge. (Oracle only)
<cartridge_name>.hist.default.storagelocation.size	Data device size to be allowed for the specified cartridge on the target database. (SQL Server)
<cartridge_name>.hist.log.storagelocation.size	Log device size to be allowed for the specified cartridge on the target database. (SQL Server)
<cartridge_name>.archiveaccess.default.storagelocation.size	Data device size to be allowed for the specified cartridge on the archive access database. (SQL Server)
<cartridge_name>.archiveaccess.log.storagelocation.size	Log device size to be allowed for the specified cartridge on the archive access database. (SQL Server)

archiveaccess.owner.username	Required if the business flow contains a database to database cartridge run on Oracle database. (Oracle only)
archiveaccess.owner.password	Required if the business flow contains a database to database cartridge run on Oracle database. (Oracle only)
hist.dbadmin.username	Required if the business flow contains a distributed archive database to database cartridge.
hist.dbadmin.password	Required if the business flow contains a distributed archive database to database cartridge.
histreloc.sourceinterface.dblink.name	Name of the history to source interface dblink. (Oracle only)
sourceinterface.histreloc.dblink.name	Name of the source interface to history dblink (Oracle only)
gen.override	Set to true for 3-tier configuration, or false for 2-tier configuration. If omitted or left blank, defaults to false. Required for business flows containing database to file cartridges.
source.location.name	The location of the source database. Required for business flows containing database to file cartridges.
oa.deploy.config	The database to file deployment configuration. Accepted values are 2tier or 3tier. If omitted or left blank, defaults to 2tier. Required for business flows containing database to file cartridges.

Scripted business flow deployment properties file example

The following examples demonstrate the use of the properties file.

Oracle example

Oracle database to database business flow

```
#Business Flow Install Properties
businessflow.install.configuration=PrepareAndDeploy
environment.id=MyEnv
rdbms.name=oracle
source.dbadmin.username=system
product.configuration=SI
ORD.hist.default.storagelocation=USERS
ORD.hist.temp.storagelocation=TEMP
ORD.archiveaccess.default.storagelocation=USERS
ORD.archiveaccess.temp.storagelocation=TEMP
```

Oracle example

Oracle database to file business flow

```
#Business Flow Install Properties
businessflow.install.configuration=PrepareAndDeploy
environment.id=MyEnv
rdbms.name=oracle
source.dbadmin.username=system
source.location.name=OBTINTF_DB
oa.deploy.config=2tier
gen.override=false
```

1.7.5. Data Privacy and Protection Appliance (SecureData) Integration

SSL Certificate

> **Note**
>
> For an easy import of certificates SDM provides a wrapper script `importCert.bat` (Windows) / `importCert.sh` (Linux) to import the certificate using the command as below:
>
> `importCert.bat/.sh <Path to Certificate File>`

We need to make sure that we trust the DPP Appliance (SDA) certificate. We need to add this certificate in our trust store. For Windows, please refer to *OpenText™ Data Privacy and Protection SimpleAPI_6.22_Install.pdf (Trusted Root Certificates section)*. Also, we need to make sure that Java trust store also has the DPP Appliance (SDA) root certificate. A sample command is below:

```
"<SDM Install Directory>\jre\bin\keytool" -import -trustcacerts -alias VoltageInternal -file "<Path to Cerfiticate File>" -keystore "<SDM Install Directory>\jre\lib\security\cacerts"
```

Default password: changeit

For example:

```
"C:\SDM\SDM760\jre\bin\keytool" -import -trustcacerts -alias VoltageInternal -file "C:\Temp\Acne Root.cer" -keystore "C:\SDM\SDM760\jre\lib\security\cacerts"
```

Please note that the keystore to be used is the cacerts file under jre/lib/security directory.

> **Note**
>
> If you are using Windows 7, it may so happen that the public DPP Appliance (SDA) instance (voltage-pp-0000.dataprotection.voltage.com) provided for testing purpose may give you an error (in the sdm.log file you will see an error VE_ERROR_CANNOT_VERIFY_CERT). This is due to one of the update is missing on your OS. Please refer to https://support.microsoft.com/en-us/help/3140245/update-to-enable-tls-1-1-and-tls-1-2-as-a-default-secure-protocols-in for required Windows 7 update. If the update is done then you need to make sure that you apply "Easy fix" provided in the same article.

DPP Appliance (SDA) Setup with Active Directory

Proxy Settings

On the Windows SimpleAPI, use WinHTTP library to communicate with DPP Appliance (SDA) . This library is different from WinINET, and therefore, the proxy settings are different from the settings that you use in Internet Options.

For setting proxy server (and similarly to bypass proxy server in case of intranet), `netsh winttp` command must be used. Below is the list of various options available for the same:

- To view current WinHTTP proxy settings

  `netsh.exe winhttp show proxy`

- To set WinHTTP proxy server

  `netsh.exe winhttp set proxy <proxyserver:port>`

- To import the IE proxy settings of the current user

  `netsh.exe winhttp import proxy source=ie`

- To reset proxy settings

  `netsh winhttp reset proxy`

- To bypass proxy server

  `netsh winhttp set proxy proxy-server=<server:port> bypass-list=<address1;address2>`

opentext™

© Copyright 2026 Open Text

For more info, visit https://docs.microfocus.com