

# OpenText™ Structured Data Manager

Software Version CE 25.3.0

## Tutorial

**opentext™**

Document Release Date: September 2025  
Software Release Date: September 2025

## Legal notices

Copyright 2017-2025 Open Text

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

## Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for updated documentation, visit <https://www.microfocus.com/support-and-services/documentation/>.

## Support

Visit the [MySupport portal](#) to access contact information and details about the products, services, and support that OpenText offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- View information about all services that Support offers
- Submit and track service requests
- Contact customer support
- Search for knowledge documents of interest
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

# Contents

About this document .....	6
Prerequisites .....	6
Related documentation .....	6
 Chapter 1: Tutorial introduction .....	8
Tutorial overview .....	8
Tutorial prerequisites .....	8
Structured data management concepts .....	9
Planning your archive .....	9
Designer overview .....	9
Designer main window .....	10
Summary and next steps .....	10
 Chapter 2: Configure the Demarc data .....	11
Load the Demarc data .....	11
Obtain the tutorial solution .....	13
Summary and next steps .....	14
 Chapter 3: Create a cartridge .....	15
Create a cartridge .....	15
Navigate in the Cartridge Editor .....	16
Operations .....	17
Data Source .....	18
Data masking .....	18
Validation .....	18
Column inclusion .....	19
Name override .....	19
Custom Properties and Custom Selection Program .....	20
Preview the cartridge .....	20
Summary and next steps .....	21
 Chapter 4: Create business flows .....	22
Create a business flow .....	22
Split an activity .....	23
Add an interrupt .....	24

Create an undo business flow .....	26
Summary and next steps .....	26
<b>Chapter 5: Create data access cartridges .....</b>	<b>27</b>
Identify the data .....	27
Add a data access rule .....	27
Create a data access cartridge .....	28
Display the runtime parameters for data access cartridge .....	32
Custom DAC detail template .....	35
Summary and next steps .....	36
<b>Chapter 6: Create an archive definition .....</b>	<b>37</b>
Create a project .....	37
Create a model .....	38
Create a new model .....	39
View model properties .....	39
Add tables to the model .....	40
Add transactional and lookup tables .....	41
Add multiple table uses with conditional relationships .....	42
Create the orders data model .....	44
Preview the data .....	45
Define a rule .....	45
Create a rule .....	46
Create a parameter .....	46
Specify rule properties .....	47
Define additional rules .....	49
Examine the data .....	49
Define a rule on the STATUS table .....	50
Define a rule on the PRODUCT Table .....	51
Define a preview-only rule .....	51
Manage connections .....	53
Summary and next steps .....	53
<b>Chapter 7: Deploy and run business flows .....</b>	<b>54</b>
Deployment prerequisites .....	54
Deploy the business flow .....	54
Deploy the undo business flow .....	56
Run the business flow .....	57
Summary and next steps .....	61

Chapter 8: Query the archive ..... 62

    Configure the AQS cache ..... 62

    Query your archived data ..... 63

    Deploy and run the data access cartridge ..... 63

    Run the undo business flow ..... 64

    Summary and next steps ..... 65

Glossary ..... 66

Send documentation feedback ..... 72

# About this document

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

This tutorial is designed to help you get started with Structured Data Manager.

This guide provides information about:

- an example archive
- steps for designing the example archive
- deployment of the example archive
- execution of the example archive

## Prerequisites

Prerequisites for using this product include:

- knowledge of the operating system
- database knowledge
- application knowledge

## Related documentation

Document Name	Description
<i>OpenText™ Structured Data Manager API Reference Guide</i>	Provides reference to the available programming interfaces.
<i>OpenText™ Structured Data Manager Certification Matrix</i>	Provides information about supported Operating Systems, databases, browsers, software integrations and other technology stacks.
<i>OpenText™ Structured Data Manager Concepts Guide</i>	Explains the major concepts of database archiving in general and Structured Data Manager in particular.
<i>OpenText™ Structured Data Manager Developer's Guide</i>	Explains how to use the Designer component to design, build, test, and deploy your archiving projects.
<i>OpenText™ Structured Data Manager Installation Guide</i>	Explains how to install the product.
<i>OpenText™ Structured Data</i>	Lists any items of importance that were not captured in the regular

Document Name	Description
<i>Manager Release Notes</i>	documentation.
<i>OpenText™ Structured Data Manager Runtime Guide</i>	Explains how to use the Web Console component to run, monitor, and administer business flows that move data to and from the database.
<i>OpenText™ Structured Data Manager Troubleshooting Guide</i>	Explains how to diagnose and resolve errors, and provides a list of common errors and solutions.
<i>OpenText™ Structured Data Manager Upgrade Guide</i>	Explains how to upgrade the product and archive schema generated by the earlier versions of the product.
<i>OpenText™ Structured Data Manager Discovery Guide</i>	Explains the purpose, how to install and use Discovery.

# Chapter 1: Tutorial introduction

This section provides you with an overview of the archive building process and the tutorial itself.

- [Tutorial overview](#)
- [Tutorial prerequisites](#)
- [Structured data management concepts](#)
- [Planning your archive](#)
- [Designer overview](#)
- [Summary and next steps](#)

## Tutorial overview

This tutorial is designed to introduce you to Structured Data Manager. The tutorial walks you through the following high level tasks using the sample Demarc database objects:

- designing a model of the database objects to be archived
- defining rules
- previewing your model and rules
- creating a cartridge that archives from database to file
- previewing the results of the cartridge on your data
- creating a business flow to call the cartridge and perform some additional processing
- creating an environment
- creating Web Console users
- deploying the business flow
- running the business flow from the Web Console
- confirming the results of the business flow
- changing the business flow and redeploying it

## Tutorial prerequisites

Before starting this tutorial, ensure all of the following have been completed:

- You have installed Structured Data Manager software on your computer.
- You have installed a database that is supported by Structured Data Manager.



**TIP:** The steps and screen images in this tutorial assume an Oracle, SQL Server, or Sybase database. If you use another database, you can still follow the steps, but you may encounter some small differences in the steps and/or the appearance of the product.

- You have run the Web Console to set up a repository and at least one environment.

For details, see section: Configuring the Web Console, of the *OpenText™ Structured Data Manager Runtime Guide*.

- You have noted all of the following:
  - database administrator user name
  - database administrator password
  - repository username (for example, obt\_rep)
  - repository password
  - encryption key

## Structured data management concepts

For conceptual information about Structured Data Manager, see the *OpenText™ Structured Data Manager Concepts Guide*.

## Planning your archive

For the purposes of this tutorial, the scenario is provided to you, which obviates the need for any planning activities. When you come to build your own archive, you will need to spend considerable time analyzing the data, applications, and environment before you start building your project in Designer.

For information about how to plan for your archiving project, see *OpenText™ Structured Data Manager Concepts Guide*.

## Designer overview

Most of the work for developing this example archive is performed in Designer. Designer is a powerful graphical development environment used to:

- model data
- apply rules
- design cartridges
- design business flows that employ cartridges and implement additional logic
- preview models and cartridges for testing purposes
- deploy cartridges and business flows to a specified environment (local or remote)

Designer drastically improves the productivity of archive developers. Developers no longer need to spend hours writing and debugging complex SQL. They simply drag and drop on the editor to include tables and relate them to other tables. After the model is defined, they can point and click to define rules and policies on tables as desired. When the business flows are ready, the developer can deploy them to any supported environment, local or remote, from Designer.

Navigation and interaction with the various components within Designer is consistent, and once familiar with the patterns in one portion of Designer, you should be able to work with any part of Designer.

## Designer main window

After you have created a project with a connection, the Designer main window is displayed. You will use the main window to define your project.

## Summary and next steps

In this section you learned about:

- the overall structure of the tutorial
- the prerequisites for building the tutorial
- the basic concepts of Structured Data Manager
- the interface of Designer

You are now ready to begin building the example archive by [Configure the Demarc data, on page 11](#).

## Chapter 2: Configure the Demarc data

To follow the instructions in this tutorial, you must have the sample Demarc data set loaded in your database.

This section explains how to obtain and load the Demarc data and includes:

- [Load the Demarc data](#)
- [Obtain the tutorial solution](#)
- [Summary and next steps](#)

### Load the Demarc data

The example in this tutorial is based upon the Demarc data set. You must install this schema and populate it before you can start the tutorial.

#### To load the data into your database

1. Install and configure the database of your choice. See [Tutorial prerequisites](#) for information about which databases you can use.

**NOTE:** The demo data loader does not support Oracle RAC or SQL Server Windows authentication.

2. Install and configure Structured Data Manager according to the *OpenText™ Structured Data Manager Installation Guide*.
3. Open a command window.

On Windows, select **Start > Run**, type `cmd`, and click **OK**.

4. Change to the bin directory where you installed Structured Data Manager.

For example:

- **Windows:** `c:\SDM\SDM760\obt\bin`
- **UNIX:** `cd /home/SDM760/obt/bin`

5. Type the command appropriate for your database:

**NOTE:** Prerequisite for loading the DEMARC data into Oracle, DB2 or Sybase database  
Before loading data, make sure that all their respective libraries are properly configured.  
Refer *Configure the third-party libraries* in *OpenText™ Structured Data Manager Runtime Guide*.

On **Windows**:

```
load_demo oracle  
load_demo sqlserver  
load_demo sybase  
load_demo db2  
load_demo mysql  
load_demo postgresql  
load_demo generic
```

On **UNIX**:

```
./load_demo.sh oracle  
./load_demo.sh sqlserver  
./load_demo.sh sybase  
./load_demo.sh db2  
./load_demo.sh mysql  
./load_demo.sh postgresql  
./load_demo.sh generic
```

**NOTE:** The generic option is for JDBC/ODBC data sources.

6. Respond to the prompts. Default values are displayed next to the prompts inside of square brackets [ ]. It may take a few minutes for the scripts to complete running.

**TIP:** If you want to use a schema name other than DEMARC, enter the desired name when prompted for demo schema/user name.

**TIP:** SQL Server database names are case sensitive. Therefore, if you accept the default name, you must use DEMARC when referencing it.

7. To confirm that the scripts ran properly, check the log files located in <install\_dir>\obt\demo\<db> for any errors:

```
ngfloadlog.log for DB2  
ogfloadlog.log for Oracle  
sgfloadlog.log for SQL Server  
ygfloadlog.log for Sybase
```

8. If you have an ad hoc query tool, you can use it to confirm the presence of Demarc and its tables.

## Obtain the tutorial solution

A completed version of the tutorial is shipped with Structured Data Manager for your reference. You may wish to open this project and review it after you have attempted to complete the tutorial yourself.

### To obtain the tutorial solution project

1. Launch Designer.
  - On Windows, from the Start menu, open Designer by selecting **Structured Data Manager > Designer**.
  - On Linux, use the Designer desktop link, if you opted to create the links at install time.

Otherwise, navigate to `<install_dir>/obt/bin` and type:

```
./designer.sh
```

When you start Designer for the very first time, you are automatically prompted to create a new project.

The first field is the project **Name**. The second field is the **Database connection** used to get the table definitions.

**TIP:** If you are not prompted to create a new project, select **File > New Project**. Or, you can click the New Project icon.

In the Name field, type `DEMARC Orders App v1 soln` as the name of your new project.

2. For Database, if you already created a connection to the database with DEMARC schema, choose the connection from the list. Otherwise, click **New** to set up such a database connection.

For more information on creating projects and database connections, see [Create a project, on page 37](#).

3. After the New Project dialog box is filled out, click **OK**.
4. Select **File > Import**.

The Import Existing Project dialog box appears.

5. Browse to the location of the tutorial solution project.

On Windows, it can be found in `<install_dir>\obt\demo\project`.

On UNIX, it can be found in `<install_dir>/obt/demo/project`.

6. Select `tutorial_soln_<db_type>.hdp`, where `<db_type>` is your database type (oracle, sqlserver, sybase, db2).
7. Click **Open**.

You should now have a complete, working version of the tutorial solution project to which you can compare your own solution.

## Summary and next steps

In this section, you learned:

- how to run the scripts to load the sample data into the database you plan to use for the tutorial
- how to obtain a completed version of the tutorial

The next step in building your archive is to create a project as well as Create an archive definition , on page 37.

# Chapter 3: Create a cartridge

To this point, you have been working in Designer to define a data structure and preview it. After you define the structure and test it, you must describe the operation of a particular archiving model by creating a cartridge.

- [Create a cartridge](#)
- [Navigate in the Cartridge Editor](#)
- [Preview the cartridge](#)
- [Summary and next steps](#)

## Create a cartridge

A cartridge is the mechanism by which Structured Data Manager specifies a versioned instance of a model for one type of archiving. In your cartridge definition, you select the following:

- The type of archive operation to perform (database to database or database to file)
- Which rules to apply
- How to treat transactional and chaining objects
- How to treat lookup objects

For this tutorial, create a cartridge that archives from a database to a file (an XML file).

### To create a cartridge

1. Go to **File > New Cartridge**.
2. In the New Cartridge dialog box, type **Orders\_D2F** as the Name.
3. Select **Database to File** as the Type.
4. If it is not already selected, select the **Model** radio button.

#### Advanced concept

- **Model** means that the cartridge is based upon a defined data model. This type of cartridge is typically used for ongoing archive operations.
- **Schema** means that the cartridge is based upon the database schema rather than a defined data model with relationships. This type of cartridge is typically used for application retirement or the cleanup of orphan tables.

Related information: *OpenText™ Structured Data Manager Developers Guide*

5. If it is not already selected, select **Orders** as the model.
6. Click **OK**. The Database to File Cartridge editor appears.

## Navigate in the Cartridge Editor

When you enter the cartridge editor, the top section displays:

- The title indicates the type of cartridge, in this case, **Database to File**.
- **Model** displays the model used by the cartridge, for example, Orders.
- **Version** indicates the cartridge version.
- **Indexing** specifies whether you have assigned an indexing cartridge to index your data for faster searching.

**IMPORTANT:** The format of the indexing file created depends upon the indexing server configuration on the Web Console. By default it is Apache Solr.

For more information about creating indexing cartridges, see the *OpenText™ Structured Data Manager Developers Guide*.

If you look carefully at the bottom of the editor, a number of tabs are displayed, which correspond to the different parts of the cartridge you can edit. The first tab, Overview, is an overview of the cartridge. Each section on the page has a title that acts as a hyperlink to the corresponding tab.



At the top of each page, you will find a link called **Back to Overview**, which returns you to the Overview page. Of course, you can also return to Overview by clicking the **Overview** tab.

Any change you make in any page is immediately reflected in the Overview page.

## Operations

1. Click the **Operations** tab. On this page, you can specify whether the rows of a table should be copied and purged (default), or just copied.

**NOTE:** The lookup tables in your model (STORE, PRODUCT, CUSTOMER, STATUS, SALESREP, COMMPLAN, and BUNIT) have no purge option.

You can copy the data from the lookup tables, but you cannot purge them afterwards. Structured Data Manager assumes that lookup tables are needed in their entirety and therefore does not purge the copied data. If you need to delete rows from the table, you should create it as a transactional table rather than a lookup.

2. Click Back to **Overview** to return to the Overview page.

## Data Source

Click the **Data Source** tab. The Data Source tab settings apply only to a 3-tier topology. In 3-tier topologies, lookup tables may reside in one of a few possible locations (source or archive database, or both). Therefore, you need to specify where Structured Data Manager should take the lookup table data from during movement. This tutorial is based on a 2-tier topology, so you do not need to change anything on this tab.

## Advanced concept

A 3-tier topology is a configuration in which one cartridge archives data from a source database (tier 1) to an archive database (tier 2) and another cartridge archives from the archive database to a file (tier 3).

For more information about 3-tier topologies, see the *OpenText™ Structured Data Manager Concepts Guide*.

## Data masking

1. Click the **Data Masking** tab.

Data masking of a column means Structured Data Manager obfuscates the data values in the column. This feature is useful for sensitive data, such as Social Security numbers or credit card numbers.

For the purposes of this tutorial, use this feature for the CUSTOMER.CREDITCARD and CUSTOMER.SOCIAL columns.

2. Select the **CUSTOMER** table on the left and expand it.
3. Select the **CREDITCARD** column.
4. In the Mask Type list, select **Credit card number: XXX mask**. This option will substitute X's for the numbers in the CREDITCARD column.
5. Select the **SOCIAL** column.
6. In the Mask Type list, choose **Social Security Number: random**. This option will substitute random numbers for the numbers in the SOCIAL column.
7. Click **Back to Overview** to return to the Overview page.

## Validation

1. Click the **Validation** title or tab. Validation of a column means Structured Data Manager verifies the data has not been changed between the time when the row was selected and the time it is archived.

**TIP:** Since validation occurs between selection and archiving, it does not apply to any non-intrusive environment, where data is selected and archived simultaneously. See the *OpenText™ Structured Data Manager Runtime Guide* for more information about non-intrusive environments.

2. Click the **ORDER\_HEADER** table on the left.
3. Select the **STATUSID** column on the right.
4. Click **Back to Overview** to return to the Overview page.

Notice how the Overview page updates to reflect the validation selection you just made.

5. Expand the ORDER\_HEADER node to see STATUSID listed as the column selected for validation.

## Column inclusion

In some cases, you may need to exclude certain columns from the archive to meet privacy or regulatory policies.

1. Click the **Column Inclusion** tab. The Column Inclusion page enables you to define which columns to include or exclude from the archive.
2. Take note of the radio button group in the middle of the page. It provides three options for specifying the column inclusion behavior:
  - **Include All** is the default behavior. All columns found in the table at the time the job is run are included in the archive. This option is the default choice.
  - **Include Columns** means you explicitly specify which columns to include. By doing so, the runtime job only archives the columns you have selected. If a new column is discovered when the job is launched, it is ignored. If one of the columns you included is missing at runtime, an error results.
  - **Exclude Columns** means you explicitly specify what columns not to include. The runtime job archives all the columns found in the table except for the ones you have excluded.

For the purposes of this tutorial, accept the default choice of including all columns.

3. Click **Back to Overview** to return to the Overview page.

## Name override

By default, Structured Data Manager uses the table and column names as XML element names. In most cases, the database naming standards are different from the XML naming standards, which can sometimes result in the generation of an invalid XML name.

The Name Override page enables you to correct any cases where the database table or column name leads to an invalid XML element name.

### To define an XML element name

1. Select a table or column on the left side of the page and type a new XML name on the right.
2. Click the **Name Override** tab.
3. Scroll through the list of names on the left.

- Red text means that the XML name is invalid.
- Bold text means that the name has been overridden.

For this example, notice that none of the XML names are red. Therefore, no changes are needed.

4. Click **Back to Overview** to return to the Overview page.

## Custom Properties and Custom Selection Program

The Custom Properties and Custom Selection Program tabs provide advanced functionality for experienced Structured Data Manager users. See the *OpenText™ Structured Data Manager Developers Guide* for more information about these tabs.

## Preview the cartridge

You can preview your model-based cartridge in the same way you previewed your model to confirm it is behaving as you intended.

### To preview the cartridge

1. In the toolbar, click the **Preview** tool.
2. In the Parameter Values dialog box, leave the default values and click **OK**.

Notice the differences between this dialog and the one that you saw when you previewed from the model. Because the cartridge definition specifies which rules to apply or not apply, the Parameter Values dialog box does not provide check boxes for all available parameters. Instead, it provides just one check box to enable you to apply or ignore the preview rule.

3. Review the data in the Preview tab.
  - The top part of the window shows the rows of the driving table. Select a row or range of rows in the top part of the window to filter the rows displayed in the bottom part. Use Ctl-click to select more than one row or clear the rows selected.
  - The Excluded By column displays the rule that caused a row to be excluded. All rows that are excluded are displayed in red.
  - Click the column headers to sort the rows by that value. For example, if you click the **Excluded By** column header, the rows are sorted according to the values of that column.
  - Click and drag the column borders to resize the columns in the display.
  - The model structure, including rules, is displayed in the lower left pane. Expand and collapse the node to view the tables and rule you want. Positive numbers in parentheses next to the driving table indicate the number of driving table rows included by that table. Negative numbers in parentheses next to a rule indicate the number of driving table rows excluded by the rule.

**TIP:** Expand the ORDER\_HEADER node in the lower left pane and select the CUSTOMER table. Notice that the SOCIAL and CREDITCARD column values are not masked in preview even though you specified masking in the cartridge. Masking is applied at production runtime. It is not applied during Designer preview.

- If you notice any problems in preview, you can return to the cartridge definition, modify it, and preview the results again by clicking the **Preview** tool.

4. Close the Preview tab when you are finished.

## Summary and next steps

In this section you learned to:

- create a cartridge
- edit cartridge properties in the editor
- preview the data for your cartridge

Once you are satisfied with your cartridge, you could deploy and run it by itself. However, for the purposes of this example, you must perform some additional processing. To do that, you must create a business flow that runs the cartridge along with a Groovy script. When you deploy and run the business flow, the cartridge will be deployed and run with it.

To view the next section of this tutorial, go to [Deploy and run business flows](#) , on page 54.

# Chapter 4: Create business flows

By adding your cartridge to a business flow, you can run it as part of a larger work flow. For example, you can run a pre-processing script, run your cartridge, and then run a post-processing script. You can even go one step further and split the execution of your cartridge into steps and perform some scripted operations in the middle of the cartridge execution.

- [Create a business flow](#)
- [Split an activity](#)
- [Add an interrupt](#)
- [Create an undo business flow](#)
- [Summary and next steps](#)

## Create a business flow

### To create a business flow

1. Select **File > New Business Flow**.
2. For the **Name**, enter the following: **Orders\_D2F\_BF**.
3. Click **OK** to create the business flow.

The business flow always starts from the Start activity. The left side of the editor contains a toolbar with the various objects you can insert into the business flow.

4. Click **Archive** and then click under the Start activity to place it.
5. In the Archive dialog, select **Orders\_D2F** in the **Cartridge** field, if not already selected.
6. Select **Standard** in the Selection field and **Archive** in the Data Movement field.

Advanced Concept:

- **Standard selection** uses the transactional method to select the rows to be copied to the file.
- **Advanced selection** uses a more sophisticated algorithm that attempts to find interrelated rows across multiple tables. See the *OpenText™ Structured Data Manager Developers Guide* for more information about selection algorithms.

In the **Data Movement** field, **Archive** makes a copy of the data in the file and then deletes it from the source database. **Copy** makes a copy of the data in the file.

7. Click **OK**.

You now have a business flow with one cartridge.

## Split an activity

In some cases, it can be useful to run some code during an archive operation. As an example, you might want to perform some selection analysis between the selection and data movement steps of an archive operation and take some action based on that analysis. If you determine that too many or too few rows are selected as eligible for archiving, you might want to abort or postpone the data movement.

In order to perform selection analysis, you must split the archive operation such that you can perform some processing after the Selection activity but before the Copy and Clean Up activities.

### To split the archive operation

1. In the Orders\_D2F\_BF business flow, right-click the **Orders\_D2F** archive operation and select **Split** from the pop-up menu.

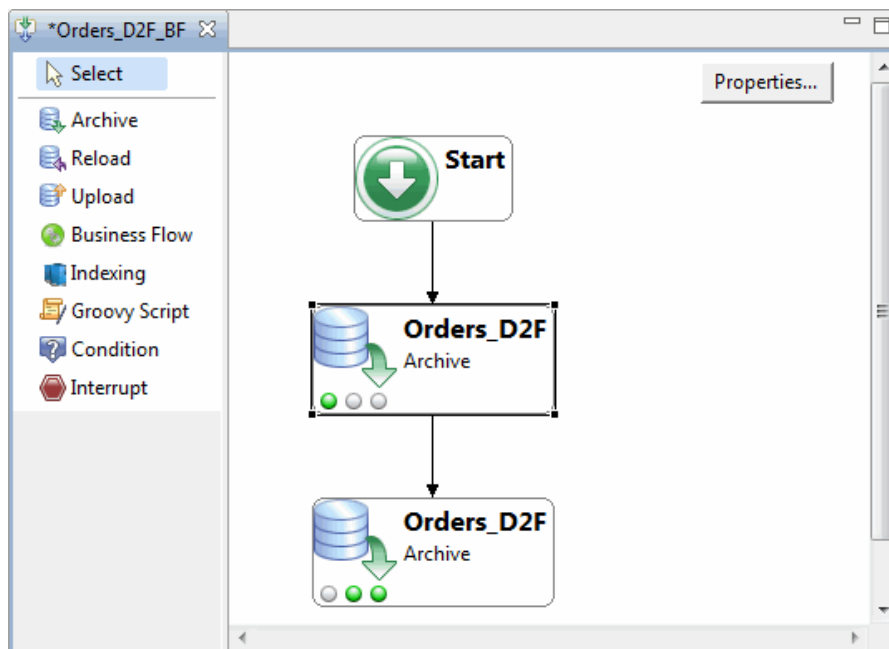
Note, there are three jobs shown in the Split Activity dialog box, one pre-activity (Selection job) and two post-activities (Archive and Clean Up jobs).

You can shuttle the jobs back and forth to specify where the split will occur. However, in this case, the default split creates a break between the Selection and Archive jobs, which is precisely what you need for selection analysis.

**NOTE:** If you run this business flow from a non-intrusive environment, the selection step will be skipped altogether.

In non-intrusive environments, selection and data movement occur simultaneously, hence splitting the activity effectively eliminates selection. For more information about non-intrusive environments, see the *OpenText™ Structured Data Manager Runtime Guide*.

2. Click **OK**. You should now see the following business flow.



You still only have one archive operation, but it is now represented by two activity boxes.

The three round icons in the lower left corner of the activity box correspond to the three pre- and post-activities of the archive operation. The icons are highlighted (green) or not (gray) to indicate how many of the activities are performed by that part of the business flow.

As you can see in this step, the first activity runs the Selection job and the second activity runs the Archive and Clean Up jobs. If you move your cursor over the icons, a tool tip indicates the name of the sub-activity.

## Add an interrupt

You can use interrupts and conditions to control the processing of your business flow. One simple example is to insert an interrupt that determines the number of rows selected for archiving. Based upon that number, it is decided whether to proceed with the Archive and Cleanup activities or exit.

1. For this interrupt, you need a new parameter.
  - a. In the Project Navigator, right-click the **Parameters** node and choose **New Parameter**.
2. From the Parameter dialog box, supply the following property values:
  - For Name: Min\_Selected\_Rows
  - For Parameter Type: Runtime
  - For Label: Minimum eligible rows
  - For Data Type: Number
  - For Length: 9
  - Default: 100



- Validation: Mandatory
  - List of Values: None
3. Click **OK**.
  4. Click the **Interrupt** tool in the toolbar and click again to place it in between the two Orders\_D2F boxes.
  5. In the Interrupt Flow Activity dialog box, select **Exit Successfully** for Type.
  6. For Message, enter the following: **Checking eligible number of rows**.
  7. For Condition, enter the following code:

```
import groovy.sql.*
String query = ""
SELECT COUNT(*) CNT
FROM DRIVING_TABLE_SEL
""

row = INTF_DB.firstRow(query)
println row.CNT
if (row.CNT > Min_Selected_Rows)
return false;
else
return true;
```

This script counts the number of rows (orders) selected for archiving. The counting happens before the rows are copied to the archive database or deleted from the active database. The number of rows is then printed to the log file.

**NOTE:** The code above contains an intentional error in order to access the debugging features later in the tutorial.

After printing to the log file, the condition uses the row count to determine what value to return:

- False means do not interrupt the business flow.
- True means interrupt the business flow as per the interrupt Type, issuing the defined message text.

Note the reference to the parameter, `Min_Selected_Rows`, in determining what to return.

8. Click **OK**.

**TIP:** Save your work. If you have not saved recently, click **Save All** in the toolbar to save what you have created to this point.

## Create an undo business flow

For each business flow that archives data, you should create a corresponding undo business flow. An undo business flow enables you to quickly return the archived data from a particular run of the archive business flow to the active database.

### To create an undo business flow

1. In the Project Navigator, right-click **Orders\_D2F\_BF** and choose **Create Undo Business Flow**. The New Business Flow dialog box displays.
2. For Name, enter the following: **Undo\_Orders\_D2F\_BF**.
3. Click **OK**. The undo business flow is created with the specified name.

**TIP:** Because Orders\_D2F\_BF only included one cartridge and no additional business logic, the undo business flow should work as generated. If the Orders\_D2F\_BF had included additional logic, such as Groovy scripting, or been run in sequence with other business flows, you would need to add to the generated undo business flow to correctly and completely undo Orders\_D2F\_BF.

## Summary and next steps

In this section you learned to:

- create a business flow
- add a cartridge to the business flow
- split the cartridge into multiple operations
- add a Groovy condition

The next step is to deploy and run this business flow by [Create data access cartridges](#) , on page 27.

# Chapter 5: Create data access cartridges

After you retire your application data, users may still need to access it, albeit less frequently than when it was in your production system. Structured Data Manager provides a number of mechanisms to access archived data, including AQS and database upload/reload. For lightweight query access to retired data, data access cartridges provide many advantages. By quickly adding data access rules and defining master-detail records, you can provide users with basic query access to retired data wherever it resides (file or database locations).

This section of the tutorial describes how to build a data access cartridge to query retired data stored in files.

This section includes:

- [Identify the data](#)
- [Add a data access rule](#)
- [Create a data access cartridge](#)
- [Display the runtime parameters for data access cartridge](#)
- [Custom DAC detail template, on page 35](#)

## Identify the data

For the purposes of this example, assume that you have spoken with your business users and determined that they require access to archived sales orders based on the following criteria:

- **Sales representative.** Sales managers want to be able to see all orders for a particular sales representative.
- **Customer.** Account owners want to be able to review orders from a particular customer.
- **Product.** Product owners want to be able to view orders of a particular product.

Since your business users are not highly technical, you want to provide them basic read access to this data in their Web browsers without the need to resort to SQL. The best way to satisfy this requirement is to build and deploy data access cartridges, and provide your users with a URL to run that cartridge to view the data.

A data access cartridge requires a data model. The first step is to identify or create a model that captures the data your users require. In this case, you will use the Orders model you created in the DEMARC Orders App v1 project in the earlier sections of this tutorial and you will simply add a data access cartridge to the project.

## Add a data access rule

In most cases, a data model that was designed for archiving requires some modification to optimize it for data access. In this scenario, you need to add a parameter and a rule that meets the criteria set by

your business users, to select orders based on sales representative.

### To modify your model for your data access cartridge

1. In Designer, open the **Orders** model.
2. Right-click **ORDER\_HEADER** and select **Add Rule**.
3. Enter **Orders by Salesrep ID** for the **Name**.
4. On the **Properties** tab, for Usage, check only **Data Access**.
5. On the **SQL** tab, for **WHERE**, enter the following clause:  
`"ORDER_HEADER"."SALESREPID"=:salesrep`
6. Click **OK**.

## Create a data access cartridge

With your model ready, create a data access cartridge that leverages your model and the new rule.

### To create a data access cartridge

1. Go to **File > New Cartridge**.
2. In the New Cartridge dialog box, type **Orders\_Simple\_DA** for the Name.
3. Select **Data Access** from the Type list.
4. Choose the **Orders** model if it is not already selected.
5. Click **OK**. The Data Access Cartridge editor displays.

Notice that your model is **Orders** and only the **Orders\_by\_Salesrep** rule appears and is selected. All of the other rules in your model are for archiving and are therefore ignored by the data access cartridge.

6. Optionally, in the **Description** field, enter a description for the data access cartridge, such as **Data Access Cartridge for Orders by Salesrep ID**.

In this case, we only have one data access cartridge. In cases where you have many data access cartridges, you may want to organize them into folders using the Folder(s) field.

7. Click **Master Template** at the bottom of the editor. The master template determines what will be displayed to the user for the master record.
8. In the upper left, select the check boxes next to the following columns:

`ORDER_HEADER.ORDERID`

`ORDER_HEADER.ORDERDATE`

`SALESREP.NAME`

Notice how the columns display as you add them.

9. Click the **ORDER\_HEADER.ORDERID** field name to select it, and change the **Header** on the right to read Order#.

**TIP:** You can click the button next to the Header field to display an edit dialog in which you can add HTML to be used in the header. For example, you might want to add a graphic or format the text.

10. Click SALESREP.NAME to select it and change the **Header** to Sales Rep.
11. Click **Add**.
12. For Name, enter Customer.

**NOTE:** Special characters and spaces in the Pseudo column name are not supported.

13. For Code, enter:  

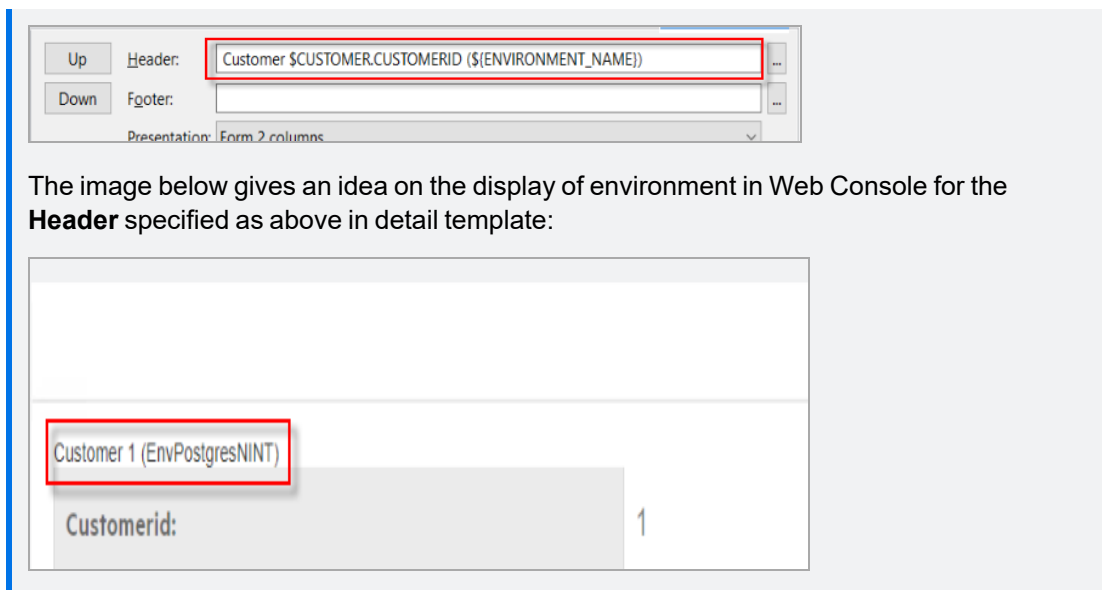
```
""<a HREF="mailto:${CUSTOMER.EMAIL}">${CUSTOMER.FIRSTNAME}  
${CUSTOMER.LASTNAME}</a>""
```
14. Click **OK**.
15. Click the Customer field name to select it.
16. Select **Allow HTML Formatting**.
17. Select the **Customer** column and use the **Up** or **Down** button to place it right after the ORDER\_HEADER.ORDERID.
18. Click **Detail Template** at the bottom of the editor. The detail template determines what will be displayed to the users when they drill down on an order from the master.

**TIP:** You can customize the DAC detail template. For the detailed steps, see [Custom DAC detail template, on page 35](#).

19. In the upper left, select the ORDER\_HEADER table. The available columns below match the currently selected table.
20. For **Header**, click the button adjacent to the field and enter the following:

```
<div>  
<span style="font-size:4em;padding-left:200px;">  
Orders  
</span>  
</div>
```

**NOTE:** Sometimes it is useful to display the environment to which the DAC is associated. To do so, in DAC, you can use system parameter **ENVIRONMENT\_NAME**. Following image gives an idea on the usage of **ENVIRONMENT\_NAME** parameter in **Header** of the **Detail Template** tab:



21. Click **OK**.
22. For **Presentation**, choose **Table**.
23. In the upper left, select the check boxes next to the following columns:  
ORDER\_HEADER.ORDERID  
ORDER\_HEADER.ORDERDATE  
ORDER\_HEADER.SHIPDATE
24. Insert the Customer column you created in step 9 below ORDER\_HEADER.ORDERID.
25. Click the Customer field name to select it.
26. Select **Allow HTML Formatting**.
27. Create a new column below ORDER\_HEADER.SHIPDATE with the following characteristics:

Field	Value
<b>Name</b>	Total
<b>Code</b>	'\$' + ORDER_HEADER.TOTAL

28. Create another new column below ORDER\_HEADER.SHIPDATE with the following characteristics:

Field	Value
<b>Name</b>	Discount
<b>Code</b>	List<String> priceValues = ORDER_LINE.PRICE List<Integer> prices = priceValues.collect{it.toInteger()} int undiscountedTotal =

Field	Value
	<pre>prices.sum() int paidPrice = ORDER_HEADER.TOTAL.toInteger() as int int discount = undiscountedTotal - paidPrice return '\$' + discount</pre>

29. In the upper left, select the **ORDER\_LINE** table.
30. For Header, enter `<p>`:
31. In the upper left, select the check boxes next to the following columns:  
 ORDER\_LINE.QUANTITY  
 ORDER\_LINE.PRICE  
 ORDER\_LINE.NOTE
32. Create a new column above ORDER\_LINE.ORDERLINEID with the following characteristics:

Field	Value
<b>Name</b>	Product
<b>Code</b>	<pre>""&lt;a HREF="http://www.google.com/search?q=\${PRODUCT.NAME[0]}&amp;btnI" target="_blank"&gt;\${PRODUCT.NAME[0]}&lt;/a&gt;""</pre>

33. Select **Allow HTML Formatting** for the new column.
34. When you are done, the Detail Template page should look similar to the following:

The screenshot shows the 'Edit Detail Template' window. On the left, a tree view lists tables: ORDER\_HEADER, ORDER\_LINE, ORDER\_LINE\_DIST, STORE, PRODUCT, and ORDER\_ATTACHMENT\_2. The 'ORDER\_LINE' table is selected. Below the tree is a 'Filter' section with a list of columns and their data types. The columns listed are: Product (Groovy), ORDER\_LINE.ORDERLINEID (NUMBER), ORDER\_LINE.ORDERID (NUMBER), ORDER\_LINE.PRODUCTID (NUMBER), ORDER\_LINE.QUANTITY (NUMBER), and ORDER\_LINE.PRICE (NUMBER). The 'Show' column has checkboxes, with 'Product', 'ORDER\_LINE.QUANTITY', and 'ORDER\_LINE.PRICE' checked. On the right, there are controls for the selected column, including 'Up', 'Down', 'Add', 'Remove', and 'Annotation...' buttons. The 'Header' field is set to 'Quantity'. Below these controls is a preview of the 'Orders' detail template. The preview shows a header section with fields: Order# (4290), Customer (Taylor Department of Treasury), Orderdate (06/25/2003), Shipdate (06/28/2003), Total (\$20), and Discount (\$0). Below the header is a table with columns: Product, Quantity, Price, and Note. The table contains two rows of data: 'DepthTeam Extreme' (Quantity: 2, Price: 10, Note: See/Call store 52) and 'Modular SwiftPlant Fluid Mixing and Storage Facility' (Quantity: 2, Price: 10, Note: See/Call store 31). At the bottom of the window, there is a tabbed interface with tabs: Overview, Master Template, Detail Template, Data Source, and Custom Properties.

35. Save your work.
36. Click **Preview** to confirm that the cartridge is working as expected.
37. Enter a value between 1 and 50 for the salesrep parameter in the Parameter Values dialog. The rows for that salesrepid should then appear in black in the preview. The other rows appear in red. When you are satisfied that your cartridge is working correctly, continue to the next step.

## Display the runtime parameters for data access cartridge

Sometimes you would like to know the runtime parameters values which were applied to detailed template to display a report through DAC.

Below is an illustration of the same for Orders model.

### To modify Orders model for your data access cartridge:

1. In Designer, open the **Orders** model.
2. Right-click **ORDER\_HEADER** and select **Add Rule**.
3. Enter Orders by Order year for the name.
4. On the **Properties** tab, for **Usage**, check only **Data Access**.
5. On the **SQL** tab, for **WHERE**, enter the following clause:  
`"ORDER_HEADER"."ORDER_YEAR">=:order_year`
6. Click **OK**.
7. Right-click **ORDER\_LINE** and select **Add Rule**.
8. Enter Orders by quantity for the name.
9. On the **Properties** tab, for **Usage**, check only **Data Access**.
10. On the **SQL** tab, for **WHERE**, enter the following clause:  
`"ORDER_LINE"."QUANTITY"=:quantity`
11. Click **OK**.

### To create a data access cartridge to display runtime parameters

1. Go to **File > New Cartridge**.
2. In the New Cartridge dialog box, type **Orders\_Params\_DAC** for the name.
3. Select **Data Access** from the **Type** list and choose the **Orders model** if it is not already selected.
4. Click **OK**. The Data Access Cartridge editor is displayed.
5. Make sure that the two rules '**Orders by Order year**' and '**Orders by quantity**' are selected under **Selection Rules** in the **Overview** tab.



**NOTE:** Only the rules and their corresponding rule parameters which are enabled in DAC will be available for use in the Header, Footer and Pseudo columns.

6. Click **Master Template** at the bottom of the editor. The master template determines what will be displayed to the user for the master record. In the upper left, select the check boxes for the columns which you wish to display.
7. Click **Detail Template** at the bottom of the editor. The detail template determines what will be displayed to the users when they drill down on an order from the master. Select the columns which you wish to be displayed.
8. In the upper left, select the **ORDER\_HEADER** table. For **Header**, click the button adjacent to the field and enter the following:

```
<h3>Orders received in or after ${order_year}</h3>
```

where **order\_year** is the runtime parameter of the rule 'Orders by Order year'.

9. Similarly, select the **ORDER\_LINE** table and for the **Header** enter the following:

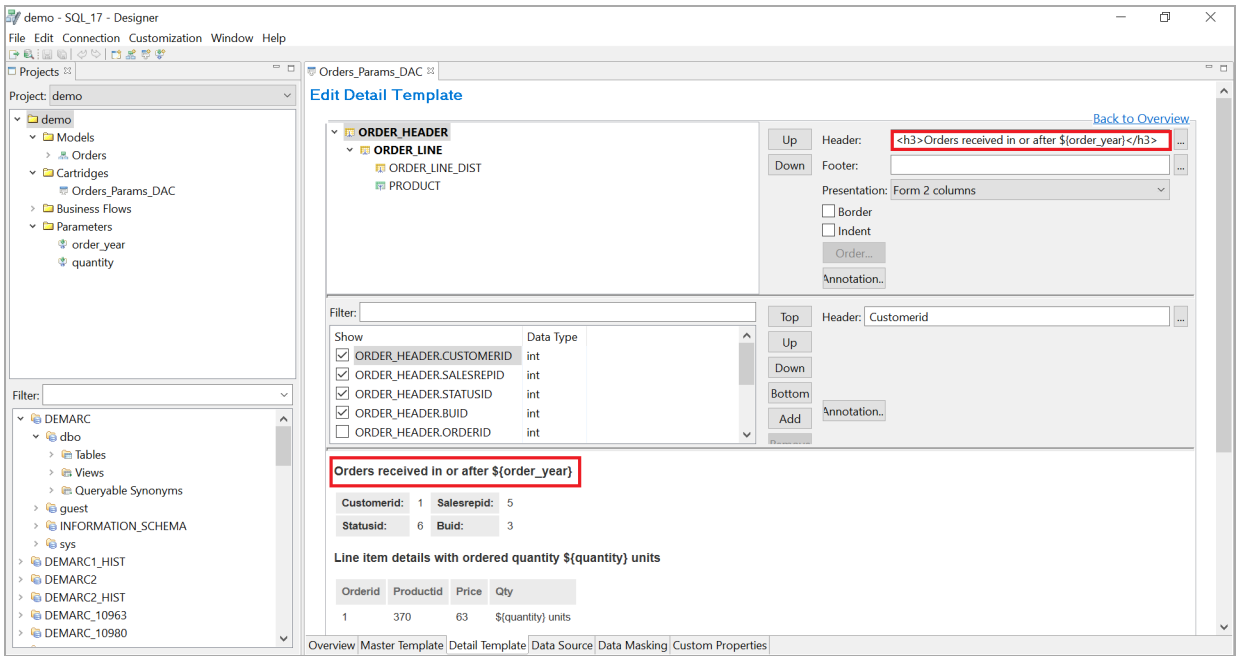
```
<h3>Line item details with ordered quantity ${quantity} units</h3>
```

where **quantity** is the runtime parameter of the rule 'Orders by quantity'.

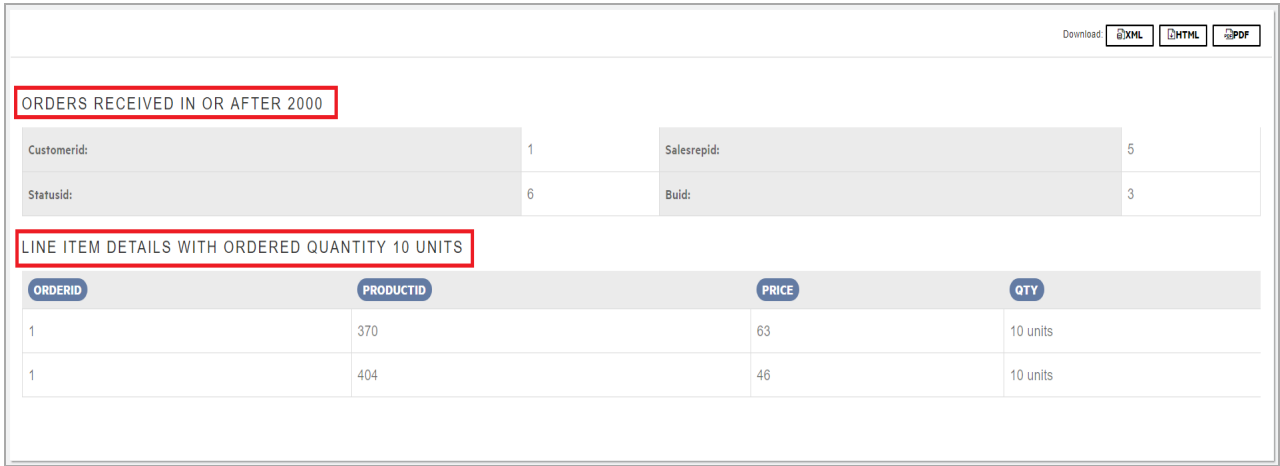
10. Create a new Pseudo column with the following characteristics:

Field	Value
Name	Qty
Code	quantity + ' units'

11. When you are done, the Detail Template page should look similar to the following:



12. Save your work.
13. Click **Preview** to confirm that the cartridge is working as expected.
14. When you run the DAC in the Web Console with values for runtime parameters as order\_year = 2000 and quantity = 10, the following image depicts how the runtime parameters and pseudo column are displayed on the detailed template page.



**NOTE:** While performing Free Text Search, the DAC associated with the Indexing Cartridge will not display the values for the runtime parameters of the DAC because in case of Free Text Search the DAC displays the results of the data which was indexed based on the rules applied to the Indexing Cartridge irrespective of the rules applied on the DAC. Hence one should avoid displaying the runtime parameters for DAC if you wish to associate DAC with the Indexing cartridge.

## Custom DAC detail template

Structured Data Manager provides an option to customize the DAC detail template. You can export the default DAC detail template provided, customize as needed and import it back.

To customize DAC detail template, perform the following:

1. Navigate to the `<Designer_HOME>\.dataaccess\resources` location, open the `dacCustomStyle.css` in a text editor and define all the necessary classes and styles.

**CAUTION:** It is highly recommended not to modify the existing classes in `dacCustomStyle.css`. Make sure to only add on to what is already available.

If you have image files that you want to use it for customization, say logo or icons, then make sure that all the resources are made available in `<Designer_HOME>\.dataaccess\resources` folder.

2. In the **Detail Template** tab, the **Preview** option is set to **Default**. Click **Export Default**.

A dialog showing the .xsl file along with the path is displayed. Click **OK**.

For example, the default DAC detail template, `dac_custom_default.xsl`, is exported under `<Local Folder>\Users\<Username>\AppData\Local\Temp\2\` folder.

3. Navigate to the `<Local Folder>\Users\<Username>\AppData\Local\Temp\2\` location, open the exported .xsl file in a text editor, using the classes and styles defined in the `dacCustomStyle.css` file, modify the .xsl as per your requirement. Save your changes.

**NOTE:** Make sure to provide the relative path starting from `.dataaccess` for the resources you are using. For example, ``

4. In the **Detail Template** tab, click **Import**. The open dialog is displayed. Navigate to the `<Local Folder>\Users\<Username>\AppData\Local\Temp\2\` location and select the .xsl you modified in the earlier step.

A dialog showing the path where the file gets imported is displayed. Click **OK**.

You can now view your customized DAC detail template and the **Preview** option is now set to **Custom**. You can toggle between **Default** and **Custom** preview options.

**NOTE:** If you want the customization to be available in Web Console, make sure that all the resources, such as logo or icons, are copied under `<OBT_HOME>/.dataaccess/resources` folder.

If you want to discard these customized changes, click **Discard Changes**. The settings will change according to the default DAC detail template provided and the **Preview** option points to **Default**.

If you want to enhance your customized DAC detail template, you can export the customized DAC detail template. Click **Export Custom**. A dialog showing the saved location of the file is displayed. Click **OK**.

**IMPORTANT:** If you modify any of the tables or check/uncheck any column, the default DAC detail template gets modified. You need to export this modified default DAC detail template, add the required customizations and import the template again.

## Summary and next steps

In this section, you learned:

- how to identify data
- how to add a data access rule
- how to create a data access cartridge
- how to display the runtime parameters for data access cartridge
- how to customize DAC detail template

The next step in building your archive is to deploy the cartridge by [Deploy and run business flows](#) , on [page 54](#)

For more information, see the instructions for deploying a cartridge in the *OpenText™ Structured Data Manager Developers Guide*.

# Chapter 6: Create an archive definition

This section walks you through the first phase of developing your archive solution for the Demarc data.

- [Create a project](#)
- [Create a model](#)
- [Add tables to the model](#)
- [Preview the data](#)
- [Define a rule](#)
- [Define additional rules](#)
- [Define a preview-only rule](#)
- [Manage connections](#)
- [Summary and next steps](#)

## Create a project

Projects provide you with a way to organize archive solutions. For example, you might collect all of the models, cartridges, business flows, and parameters required to archive the data associated with a particular version of a particular application.

### To create a project

1. Launch Designer.
  - On Windows, from the Start menu, open Designer by selecting **Structured Data Manager > Designer**.
  - On Linux, navigate to `<install_dir>/opt/bin` and type:

```
./designer.sh
```

When you start Designer for the very first time, you are automatically prompted to create a new project.

The first field is the project **Name**. The second field is the **Database connection** used to get the table definitions.

**TIP:** If you are not prompted to create a new project, select **File > New Project**. Or, you can click the Create a new project icon.

In the Name field, type `DEMARC Orders App v1` as the name of your new project.

2. Leave the Database field blank and click **New** to set up a database connection.

3. Type DEMARCDv1 in the **Connection Name** field.
4. Select the database type from the list:
  - Oracle
  - SQL Server
  - Sybase
  - DB2
  - Vertica
  - JDBC URL
5. Click **Next**.
6. Type the connection information in the Source Database page.

**NOTE:** Throughout the tutorial, the steps and screen images assume Oracle, SQL Server, Sybase, or DB2. If you are using another supported database, you will notice some small differences in the steps and appearance of the product.

**TIP:** For SQL Server, Windows Authentication indicates that the operating system login for the machine is the same as the SQL Server login, and once you are logged into the machine, you need not authenticate again for the SQL Server instance. If you do not select this option, the SQL Server login is distinct from the operating system login for the machine, and logging into the machine does not imply that you are authenticated for the SQL Server instance as well.

**TIP:** Only select the **Save Password** option if you work in a development environment where security is not a concern. The saved passwords can be extracted from Designer configuration files.

7. Click **Finish** to close the New Connection window.
8. Click **OK** to save and close the New Project window.

The Designer main window appears and you are ready to begin creating your archive. For a general overview of Designer, see [Designer overview, on page 9](#).

## Create a model

By creating a model, you tell Structured Data Manager about the tables you want to archive as well as their relationships to one another. You can then reference this model from the cartridges that perform the actual movement of data.

This section includes:

- [Create a new model](#)
- [View model properties](#)

## Create a new model

### To create a new model

1. Right-click the **Models** folder in the Project Navigator and select **New Model** from the pop-up menu to create a model.

**TIP:** You can also start creating a new model by selecting **File > New Model**. Or, you can click the **Create a New Model** icon.

2. In the dialog box, type `Orders` for the name.
3. Click **Browse (...)** next to the Driving Table field. The Browse dialog appears allowing you to find the driving table. A driving table is the main table that drives the transaction you are archiving. Most driving tables (with a normalized schema) have subordinate tables.
4. In the Filter Text field, enter `order` and click **Search**.

**TIP:** For Oracle, if there are other schemas with table names starting with “order”, you can type `DEMARC.ORD` to retrieve only the tables in the DEMARC schema containing ‘ORD’ in the name.

**TIP:** For SQL Server and Sybase, if there are other schemas with table names starting with “order”, you can type `DEMARC.dbo.ORD` to retrieve only the tables in the DEMARC schema containing ‘ORD’ in the name.

Notice how all the tables containing the search string appear in the list.

5. Double-click the **ORDER\_HEADER** table under the appropriate schema name for your database to continue.
6. Click **Annotation**.
7. Type the following comment in the dialog box: `The Orders model is based on the ORDER_HEADER driving table.`
8. Click **OK** to close the Annotation dialog box.

**TIP:** The Annotation button appears in many places within the Designer interface. Wherever it appears, you can use it to attach comments to that specific object.

9. Click **OK** to create a new model.

The Orders model editor appears with a simple graphical representation of the `ORDER_HEADER` table.

### View model properties

1. Double-click the `ORDER_HEADER` table in the main window to see the properties of the table.

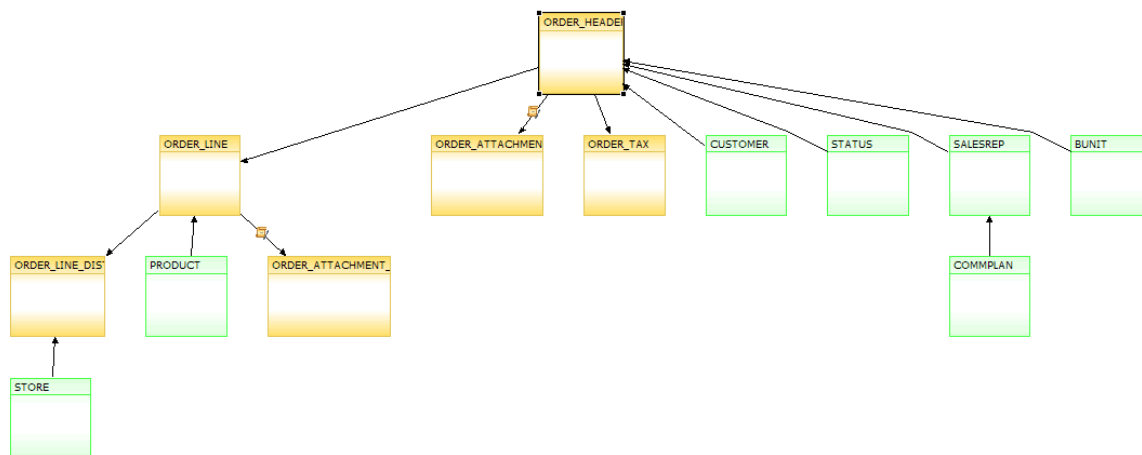
**NOTE:** Take note of the informational message in the title bar of the dialog. The message notifies you which key would be used as the data movement key. If you do not want to use the automatically chosen key, you can explicitly select one from the list of values for **Data Movement Key**.

## Add tables to the model

Once you have created your model, you can begin to add tables to it. The process of adding tables varies depending upon the type of table:

- Transactional tables contain information about the business transaction. For example, a transactional table might contain detailed tax or payment information related to each business transaction.
- Lookup tables contain helpful, non-transaction information. You might need these lookup values present for the purposes of a rule or for the sake of making an archive file more complete. For example, non-transaction information could be status definitions, or the names of the sales representatives.
- [Add transactional and lookup tables](#)
- [Add multiple table uses with conditional relationships](#)
- [Create the orders data model](#)

The steps in this section take you through adding tables. At the end, your model should look something like this example:





## Add transactional and lookup tables

### To add transactional and lookup tables

1. Right-click the driving table and select **Add Transactional Table** from the pop-up menu.

The Add Transactional Table wizard appears to help you add a subordinate table to the model.

2. Select the **All Tables** tab, if not already selected.
3. Type `%line` in the **Filter** field and click **Search**.
4. Select the **ORDER\_LINE** table under the appropriate schema for your database.
5. Click **Next**.

When specifying a model, you must explain to Designer how the tables relate to each other. Since this sample schema contains primary keys but not foreign keys, you must specify how those tables are related.

6. In the Specify How to Use or Build Keys page of the wizard, you can:
  - **Create a new foreign key associated with the following unique key or index.** Select this option and expand `ORDER_HEADER_PK` to see what columns participate in that key (only `ORDERID` in this instance). Designer discovers any existing unique keys that could serve as a virtual foreign key. Choosing this radio button enables you to choose from these discovered keys. This virtual foreign key is not created in the database, but it complements the knowledge Designer holds about your table relationships.

In this case, Designer found an existing primary key defined on `ORDER_HEADER` (`ORDER_HEADER_PK`). For the purposes of this tutorial, select the `ORDER_HEADER_PK` primary key to which you will attach your virtual foreign key.
  - **Create a new unique key and foreign key.** This option indicates that you want to create a new virtual unique key on `ORDER_HEADER` and attach a new virtual foreign key to it. These keys are not created in the database. They are virtual keys employed only in Designer to relate the tables.
  - **Create a new foreign key based only on a conditional relationship.** This option indicates that you want to create a foreign key against a conditional relationship.

**NOTE:** If you have gone through this process before on this table, Designer remembers the previously specified key and this dialog may appear differently. Ensure you make the selection described in this step for the purposes of this example.

7. Click **Next**.

The Foreign Key page appears showing the available columns in the child table (`ORDER_LINE`) on the left. The list of columns to include in the virtual foreign key is on the right.

**NOTE:** The wizard has already found the column to include by matching the column names in both tables.

8. As an experiment, add another column using the arrow (>) or remove the ORDERID column. A message at the top of the dialog informs you the column count does not match, and both the Next and Finish buttons are disabled.
9. Undo whatever change you made such that the ORDERID column is the only Key Column and click **Next**.
10. The Conditional Relationship page enables you to specify a WHERE clause to define a relationship between the tables.

In this case, you do not need a conditional relationship. Click **Finish** to return to the main Designer window.

11. Right-click the **ORDER\_LINE** table and select **Add Transactional Table** from the pop-up menu.

**NOTE:** Another way to add tables to the model is to drag them from the Database Navigator pane and drop them on the object in the model to which they are related. For example, you can click and drag ORDER\_LINE from the Database Navigator and drop it on the ORDER\_HEADER table. If you add the table this way, you are prompted to define how to add the table (as transactional, lookup, or chaining table) and then you skip to step 13.

12. On the Choose Table page of the wizard, use the **All Tables** tab to find and select **ORDER\_LINE\_DIST**.
13. Accept the default values by clicking **Next** until **Finish** is enabled.
14. Click **Finish**.
15. Right-click **ORDER\_LINE\_DIST** and select **Add Lookup Table** from the pop-up menu.
16. On the Choose Table page of the wizard, use the **All Tables** tab to find and select **STORE**.
17. Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.
18. Right-click **ORDER\_LINE** and select **Add Lookup Table** from the pop-up menu.
19. On the Choose Table page of the wizard, use the All Tables tab to find and select **PRODUCT**.
20. Accept the default values by clicking **Next** until Finish is enabled. Click **Finish**.

## Add multiple table uses with conditional relationships

In some cases, you may need to use the same table more than once in your data model in order to meet the requirements of your application. This concept is known as multiple table use. When you have multiple table uses in your model, you need to ensure that the rows retrieved for each use are disjoint. If not, you may end up with duplicate data, which might break your application.

To guarantee the rows retrieved for each use of a table remain disjoint from one another, you can apply a conditional relationship when adding each table use.

### To add a table more than once with conditional relationships

1. Right-click the **ORDER\_HEADER** table and select **Add Transactional Table** from the pop-up menu.

2. On the Choose Table page of the wizard, use the **All Tables** tab to find and select **ORDER\_ATTACHMENT**.
3. Click **Next**. ORDER\_ATTACHMENT is required twice in the model, once here and then again as a table use related to ORDER\_LINE. Therefore, it requires both a foreign key and conditional relationship.

A conditional relationship is one where a value in a parent table can be referenced by multiple child tables, but the relation is exclusive and determined by a condition.

4. Choose the **Create a new unique key and foreign key** radio button.
5. Click **Next**. On the Unique Key page, you can see the available columns, which could be used for a unique key. Structured Data Manager may have detected some unique keys and placed them in the Key Columns list by default.
6. If necessary, use the shuttle buttons (< and >) such that ORDERID is the only column in the Key Columns list.
7. Click **Next**.

On the Foreign Key page, ORDERID should be in the **Key Columns** list by default. If not, use the shuttle buttons (< and >) to move it to the Key Columns list.

8. Click **Next**.
9. On the Conditional Relationship page, enter the following for the WHERE clause:

```
{FK_ALIAS}.ATTTYPE = 'OH'
```

10. Click **Finish**.

Notice the icon on the link in the model to indicate that it is a conditional relationship. Now you will create the second table use of ORDER\_ATTACHMENT with a different conditional relationship.

11. Right-click the **ORDER\_LINE** table and select **Add Transactional Table** from the pop-up menu.
12. On the Choose Table page of the wizard, use the All Tables tab to find and select **ORDER\_ATTACHMENT**.
13. Click **Next**.
14. Choose the **Create a new unique key and foreign key** radio button.
15. Click **Next**.
16. On the Unique Key page, use the shuttle buttons (< and >) such that ORDERLINEID is the only column in the **Key Columns** list.
17. Click **Next**.
18. On the Foreign Key page, you can see the available columns, which could be used for a foreign key. In this case, Structured Data Manager has placed no columns in the Key Columns list by default. Therefore, you receive an error message at the top of the dialog indicating that you must select a foreign key.

19. Use the shuttle buttons (< and >) such that ORDERID is the only column in the **Key Columns** list.
20. Click **Next**.
21. Enter the following for the WHERE clause:  
`${FK_ALIAS}.ATTTYPE = 'OL'`
22. Click **Finish**.

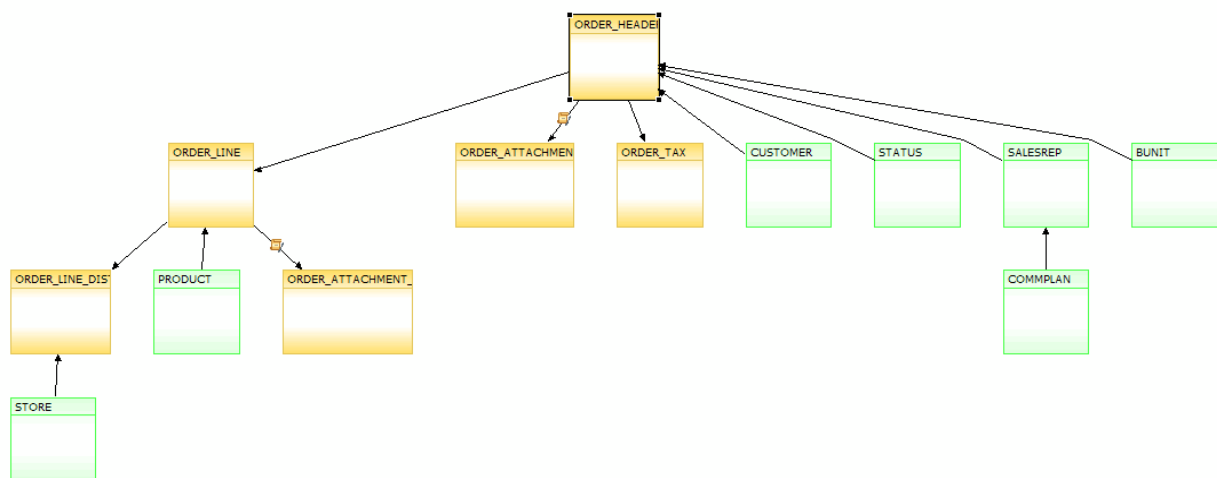
Notice that this second use of the table is named ORDER\_ATTACHMENT\_2 to distinguish it from the first use.

## Create the orders data model

To complete the remainder of your model, add the following tables, accepting the defaults in the wizard:

Table Name	Parent Table	Add as
ORDER_TAX	ORDER_HEADER	Transactional
CUSTOMER	ORDER_HEADER	Lookup
STATUS	ORDER_HEADER	Lookup
SALESREP	ORDER_HEADER	Lookup
COMMPLAN	SALESREP	Lookup
BUNIT	ORDER_HEADER	Lookup

Your model should now contain all these shapes and lines:



### Advanced concept

The line colors in the model editor are meaningful. Black means that the relationship has been validated. The relationship is checked whenever you open the project. The icons on the relationships with the ORDER\_ATTACHMENT table uses indicate the presence of conditions.

If the lines and the boxes ever change to red, it means something has changed in the underlying database and the relationship is no longer valid. The reasons for this change can include:

- the database constraint has been dropped
- you have deleted the virtual constraint
- you have edited the database connection and are now connecting to a database where the objects no longer exist
- the database is unavailable

**TIP:** Save your work. If you have not saved recently, click **Save All** in the toolbar to save what you have created to this point.

## Preview the data

You have now created the data structure. Take your first look at the data and consider what needs to be archived.

### To preview the data

1. Click **Preview** in the toolbar. If preview is disabled in the toolbar, click in the model to enable it.

**TIP:** Clicking the column headers sorts the data by that column's values. For example, click the ORDERDATE column header to sort the results by ORDERDATE. Click the ORDERID header to sort by ORDERID.

2. Scroll through the orders and notice that the ORDERIDs are consecutive numbers.
3. On the Orders Data tab, click the close icon (X) to close it and return to the model.

## Define a rule

With the model in place, you should next consider the scope of data to be archived. Rules enable you to refine your model to reflect your business and legal requirements for archiving. You can define rules in any order. Typically, your first rule defines the scope of your archiving operation. For example, in this case, the first rule specifies the age at which records become eligible for archiving.

- [Create a rule](#)
- [Create a parameter](#)
- [Specify rule properties](#)

## Create a rule

A rule is created on the table to which it applies in your model.

### To define a rule

1. In the Orders model, right-click the **ORDER\_HEADER** shape and select **Add Rule** from the pop-up menu.
2. For Name, enter the following **Orders Shipped N Months Ago**.
3. Click **Annotation**.
4. For **Category**, enter **Legal**.
5. In the text field, enter the following: **Only consider archiving Orders shipped N months ago, where N is the number of months currently approved by Legal**.
6. Click **OK**.

## Create a parameter

In most cases, you should parameterize your rules such that you can adjust their behavior depending upon your latest business rules.

For example, if your IT management team changes the required archiving window from greater than two years to greater than three years, you can change your solution by modifying a parameter rather than rewriting your rules.

### To create a parameter

1. Click **Parameters** to add a parameter for N in the rule.
2. Click **Add**.
3. For Name, enter **Min\_Months\_to\_Retain**.
4. Click **OK**.
5. For Label, enter **Months to retain in production**.
6. Change the Data Type to **Number**. The parameter type must be a Number in order to compare it to other numeric values.
7. Change the Length to **10**.
8. Change the Default to **60** (months). By default, only orders older than 60 months are eligible for archiving.
9. Choose **Mandatory** for Validation if it is not already selected.
10. Choose **Static** for List of Values.
11. Click the **Browse** button to the right of List of Values. For the purposes of this example, assume that you only want users to be able to change the retention period in quarterly increments.

12. Next to Add, type 60 for Id and 60 for Label.
13. Click the **Add** button.
14. Repeat step 12 and step 13 for 63, 66,69, and 72.
15. Click **OK**.
16. Click **OK** to accept the parameter definition and return to the Rule dialog box.

## Specify rule properties

### To specify rule properties

1. Click the **Properties** tab.
2. For **Usage**, clear all of the options except **Archive**.  
This setting indicates that the rule should only be applied to cartridges that perform archiving.

3. For **Customization**, select **Mandatory**.

4. Click the **SQL** tab.

5. Under Database, expand the **Any** node.

6. Select the **Oracle** node.

7. Click in the field labeled **WHERE....** This field contains the text to be added to the WHERE clause in the generated SQL statement when the archive cartridge is deployed.

Add WHERE clause criteria to indicate that any order qualifies if its ship date is older than a certain number of months. The exact number of months is determined at runtime by the parameter Min Months to Retain.

You can type the WHERE clause appropriate to your database directly into the WHERE... field. Or, you can use the Parameters and Columns list on the right to insert the correct parameter and column names automatically.

These WHERE clauses qualify any order with a ship date older than 24 months for archiving.

8. Add the following WHERE clause:

```
add_months("ORDER_HEADER"."SHIPDATE",  
:Min_Months_to_Retain)<sysdate
```

**TIP:** If you enter a parameter reference in your WHERE clause, for example, :Min\_Months\_to\_Retain, without first defining it, Designer automatically creates it for you. You would then need to open the parameter definition by clicking **Parameters** and change its properties as necessary, for example, setting its type, length, and default value.

9. If you are currently connected to an Oracle database, click **Validate** to confirm your WHERE clause is syntactically correct.
10. If necessary, click **OK**, to clear the Validate dialog. If the statement fails validation, check for typos.

11. Select the SQL Server node.
12. Add the following to the WHERE clause:  

```
dateadd(M, cast(:Min_Months_to_Retain as int),  
[ORDER_HEADER].[SHIPDATE]) < getdate()
```
13. If you are currently connected to a SQL Server database, click **Validate** to confirm your WHERE clause is syntactically correct.
14. Select the **Sybase** node.
15. Add the following WHERE clause:  

```
dateadd(MM, cast(:Min_Months_to_Retain as int),  
[ORDER_HEADER].[SHIPDATE]) < getdate()
```
16. If you are currently connected to a Sybase database, click **Validate** to confirm your WHERE clause is syntactically correct.
17. If necessary, click **OK**, to clear the Validate dialog. If the statement fails validation, check for typos.
18. Select the **DB2** node.
19. Add the following WHERE clause:  

```
("ORDER_HEADER"."SHIPDATE" + :Min_Months_to_Retain  
MONTHS) < CURRENT DATE
```
20. If you are currently connected to a DB2 database, click **Validate** to confirm your WHERE clause is syntactically correct.

**TIP:** If you select **Any** in the Database field, the code you enter is always used, regardless of the database or version to which the cartridge is deployed. If **Any** is supplied and a particular database version is also supplied, **Any** is used for all databases other than the database and database version with the specific syntax defined.

21. If necessary, click **OK**, to clear the Validate dialog. If the statement fails validation, check for typos.
22. Click **OK** to close the Rule dialog and return to the model.
23. Click **Preview** to check how this change affects the data.
24. Designer prompts you with a Parameter Values dialog box, where you select which, if any, rules to apply. In this case, you have only one rule and it is already selected by default.

The dialog box also includes a field for you to provide a value for the Months to retain in production parameter. For the purposes of this tutorial, leave the default value, 60, and click **OK**.

**NOTE:** For more information about eligibility analytics, see *OpenText™ Structured Data Manager Concepts Guide*. The *OpenText™ Structured Data Manager Developers Guide* also contains an advanced tutorial about eligibility analytics.



You have now restricted the archive data to orders shipped more than two years ago from today. Rows excluded by the rule appear in red and the name of the rule appears in the Excluded By column.

Do not close the preview tab yet.

**TIP:** Save your work. If you have not saved recently, click **Save All** in the toolbar to save what you have created to this point.

## Define additional rules

The application of the first rule restricted the set of rows you may archive, but, if you look closely at the data, several orders you may not want archived are still included in the archive.

This section includes:

- [Examine the data](#)
- [Define a rule on the STATUS table](#)
- [Define a rule on the PRODUCT Table](#)

### Examine the data

To understand the further refinements you may need, you must examine the preview data more carefully.

#### To examine the preview data

1. In the main Designer window, from the Database Navigator (lower left pane):
  - For Oracle, expand the DEMARC node and the Tables node.
  - For SQL Server and Sybase, expand the DEMARC node, the dbo node, and then the Tables node.
2. Double-click the **STATUS** table.
3. Click the **Data** tab. Notice that some STATUSID values have a Meaning of Legal Hold (11, 12, 13) and Open (1, 2, 8, 9, 14).

Typically, you would not archive an order if its status is anything other than Closed (3, 4, 5, 6, 7, and 10). To keep such orders from being archived, you will need to create an additional rule that excludes them.
4. Return to the Database Navigator and double-click the **Product** table.
5. Click the **Data** tab.
6. Examine the Category Column.

Notice that some of the products have a category of Recall. You might not want to archive orders where one or more of the involved products is in a recall state.
7. Close the Preview tab and PRODUCT table tab.

## Define a rule on the STATUS table

### To account for orders that are not closed

1. Return to the model in the main Designer window.
2. Right-click the **STATUS** table and select **Add Rule** from the pop-up menu.
3. In the Rule dialog box, enter `Order_Closed` in the Name field.
4. Click the **Properties** tab.
5. For Usage, select only the **Archive** option.
6. For Customization, choose **Mandatory**.
7. Click the **SQL** tab.
8. Select **Any** in the Database field.
9. Click in the **WHERE...** field.
10. Under the Status node, double-click **MEANING** from the Parameters and Columns box.
11. Type `'Closed'`. The entire WHERE clause should now read: `"STATUS"."MEANING"='Closed'`  
When applied, this rule excludes any records where the status of the order is something other than a closed state. Only closed orders are included for archiving as a result.
12. Click **Validate** to confirm the statement is syntactically correct.
13. Click **OK**, to clear the Validate dialog. If the statement failed validation, check for typos.
14. Click **OK**, to create the rule.
15. Click **Preview** to verify the effect in the Preview tab.
16. In the Parameter Values dialog, uncheck the **Orders Shipped N Months Ago rule**.
17. Click **OK**.
18. Rows excluded by the rule appear in red and the name of the rule appears in the Excluded By column.

**TIP:** Scan the list for STATUSID values of 1, 2, 8, 9, 11, 12, 13, and 14. Those should be the rows excluded by the rule.

19. After the Order Closed rule is behaving as expected, close all of the tabs except the Orders model tab. You can now create a rule to deal with those products that are in a Recall state.

## Define a rule on the **PRODUCT** Table

### To account for orders that are in a recall state

1. Right-click the **PRODUCT** table and select **Add Rule** from the pop-up menu.
2. In the Rule dialog box, type **Product Not Recalled** in the Name field.
3. Click the **Properties** tab.
4. For Usage, check only **Archive**.
5. For Customization, choose **Mandatory**.
6. Click the **SQL** tab.
7. Click the **WHERE...** field.
8. Under the Product node, double-click **CATEGORY** from the Parameters and Columns box.
9. Type `<> 'Recall '`. The entire WHERE clause is now: `"PRODUCT" . "CATEGORY" <> 'Recall '`
10. Click **Validate** to confirm that the statement is syntactically correct.
11. Click **OK**, to clear the Validate dialog. If the statement failed validation, check for typos.
12. Click **OK**, to create the rule.
13. Click **Preview** to verify the effect in the Preview tab.
14. Deselect all but the **Product Not Recalled** rule.
15. Click **OK**. Rows excluded by the rule appear in red and the name of the rule appears in the Excluded By column.
16. After you have confirmed the Product Not Recalled rule is functioning properly, close your preview tab.

## Define a preview-only rule

You may have noticed in the earlier sections that previewing large amounts of data can take some time. When in the development phase for your archive, a quick response time is helpful because you are frequently making model enhancements and immediately previewing them. Limiting the data returned for testing purposes is often worthwhile. A preview rule enables you to define a rule that applies only in preview in Designer. The rule is not applied when you generate and deploy your cartridge for actual usage.

**TIP:** Before you create a preview-only rule to limit the number of rows returned, you must consider the minimum amount of data that you need to adequately test your archive. If you return too little data, you may not be able to completely evaluate the effect of your model and rules. You may need to iterate a few times to strike the proper balance between performance and testing needs.

### To define a preview-only rule

1. Right-click the **ORDER\_HEADER** table in the model and select **Add Rule** from the pop-up menu.
2. For Name, type `Limit preview results`.
3. Click the **Properties** tab.
4. Check **Preview**.
5. For Customization, choose **Optional**.
6. Click the **SQL** tab.
7. Under Database, expand the **Any** node.
8. Select the **Oracle** node.
9. Click in the **WHERE...** field.
10. Type `ROWNUM<=10500`.
11. If you are connected to an Oracle database, click **Validate** to confirm the WHERE clause is syntactically correct.
12. Click **OK**, to close the Validation dialog. If the statement failed validation, check for typing errors.
13. Select the **SQL Server** node.
14. Enter the WHERE clause:  
`[ORDER_HEADER].[ORDERID]>=2000`
15. If you are connected to a SQL Server database, click **Validate** to confirm the WHERE clause is syntactically correct.
16. Click **OK**, to close the Validation dialog. If the statement failed validation, check for typing errors.
17. Select the **Sybase** node.
18. Enter the WHERE clause:  
`[ORDER_HEADER].[ORDERID]>=2000`
19. If you are connected to a Sybase database, click **Validate** to confirm the WHERE clause is syntactically correct.
20. Click **OK**, to close the Validation dialog. If the statement failed validation, check for typing errors.
21. Select the **DB2** node.
22. Enter the WHERE clause:  
`"ORDER_HEADER"."ORDERID">=2000`
23. Click **OK**.

24. Click **Preview** in the toolbar.
25. Deselect all the rules except the Limit preview results. The number of rows returned should match the criteria, for example, for Oracle, you should get 10500 rows.

**TIP:** Save your work. If you have not saved recently, click **Save All** in the toolbar to save what you have created to this point.

## Manage connections

For the purposes of this tutorial, you only need one database connection. Realistically, you are likely to have many different database connections from which to choose. You can manage the cache and database connection definitions in Designer through the Connection menu.

### To manage connections

1. Select **Connection > Edit Connections** from the menu.

The Connections dialog appears.

2. Using the buttons in this dialog, you can:
  - refresh a database or local cache
  - edit the connection definitions
  - delete connections
  - create new connections
3. Click **Refresh**.
4. Click **Close**.

**TIP:** Each project is associated with a particular connection by default. This association persists with the project after it is saved and is restored when the project is opened again.

## Summary and next steps

In this section you learned to:

- create a project with a connection
- create a model
- create rules
- preview data
- manage your connections

The next step in the archiving building process is to create a cartridge to actually archive the data by [Create business flows](#) .

# Chapter 7: Deploy and run business flows

When the business flow definition is complete, you are ready to deploy it to the local or remote system where you plan to execute it. Alternatively, you could also generate it on the file system for future deployment on another system by you or someone else.

This section describes how to set up the deployment environment, deploy and run a business flow in the environment, and monitor the business flow while it is running.

- [Deployment prerequisites](#)
- [Deploy the business flow](#)
- [Deploy the undo business flow](#)
- [Run the business flow](#)
- [Summary and next steps](#)

## Deployment prerequisites

Before you deploy your business flow, you must perform the following tasks in the Web Console:

- Start the Web Console service
- Navigate to the Web Console URL with your browser
- Install the repository
- Create a deployment environment
- (Optionally) create users

For detailed information on how to perform these tasks in the Web Console, see *OpenText™ Structured Data Manager Runtime Guide*. In particular, review the tutorial on configuring the Web Console.

## Deploy the business flow

In order to run your business flow, you must first deploy it.

**NOTE:** Before performing the steps in this section, you must meet all of the prerequisites in [Deployment prerequisites](#), above.

### To deploy your business flow

1. In the Project Navigator, right-click the **Orders\_D2F\_BF** and select **Deploy** from the pop-up menu.

2. Select **Deploy Remotely** enter values for your Web Console installation. If your Web Console is local, simply enter **localhost** for the Web Console URL.
3. Check **Include Documentation** to generate a PDF document that describes the business flow/cartridge structure, including your annotations.
4. Click **Next**.
5. Click **Next**.

The Deploy Environment page displays.

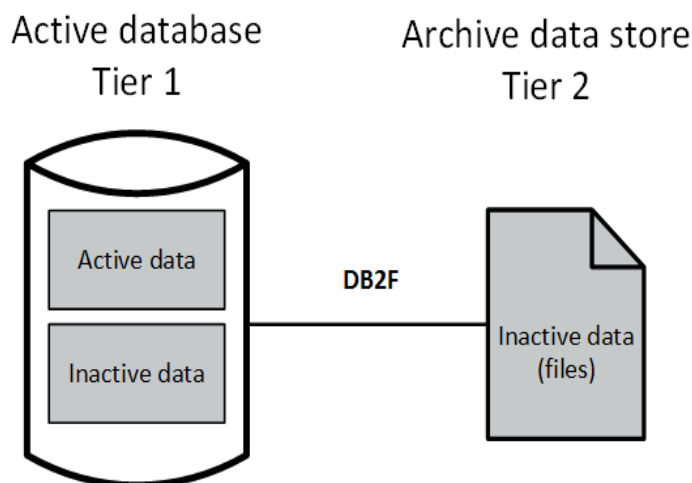
6. Select the environment to which you want to deploy this business flow, for example, Oracle\_OLTP.
7. Click **Next**.
8. Select **2 Tier Configuration**, for the purposes of this example.

**NOTE:** If you did not deploy database to database archiving as part of your environment setup, you are not prompted for topology. If database to database archiving is not present, the only option is to archive from the active database and Deployment Assistant need not prompt you for topology.

### Advanced concept

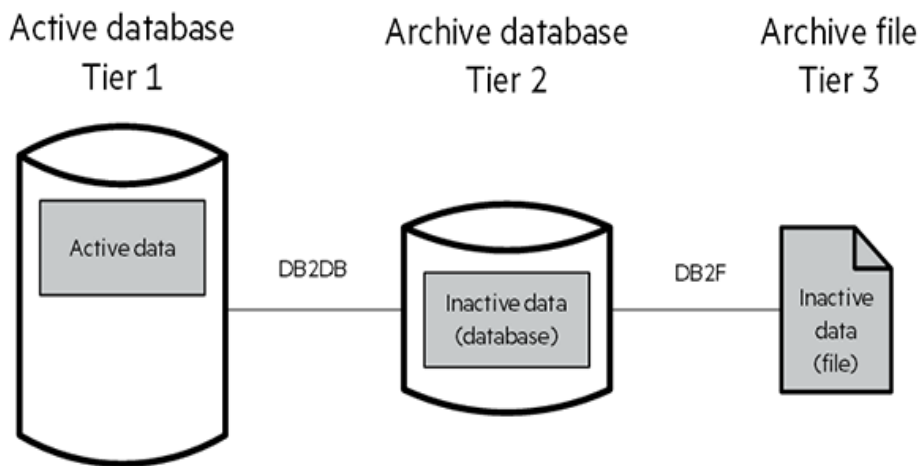
On the Topology page, you select the tier from which you want the business flow to archive. Database to file business flows can be part of a two- or three-tier architecture.

- In 2-tier mode, the business flow extracts data from the active (tier 1) database and stores the data in files (XML or CSV) in the archive (tier 2) data store.



- In 3-Tier mode:
  - For Oracle or SQL Server, a separate database to database business flow is responsible for moving data from an active database (tier 1) to an archive database (tier 2).

- The database to file business flow is then responsible for moving data from the archive database (tier 2) to a data store on tier 3.



See the *OpenText™ Structured Data Manager Developers Guide* for more information about topologies.

9. Click **Next**. The Summary page shows a summary of the options you have selected.
10. Click **Finish**. You may have to wait a few minutes before the Deployment Finished dialog appears.
11. When the Deployment Finished dialog appears, click **Show Log** to show the log file. Review the log and ensure there are no errors.
12. If you discovered errors in the previous step, click **OK** and step back through the Deployment Assistant to correct the problems. If there were no errors, click **OK** to close the log file.
13. Click **OK** to close the Deployment Finished dialog.
14. In the Deployment Assistant, if you specified Include Documentation, you should find a PDF file with your business flow's documentation located in `install_dir\OBTHOME\businessflow\environment_name`. For example:  

```
C:\SDM\OBTHOME\businessflow\Oracle_OLTP\
Orders_D2F_BF.1_0_0_0.pdf
```
15. Examine the PDF file. You should find all of your annotations along with other useful information about the structure and design of your business flow.  
  
In this same directory, you should also find the generated business flow file, `Orders_D2F_BF.1_0_0_0.busflow`.

## Deploy the undo business flow

After the `Orders_D2F_BF` is successfully deployed, you should deploy the corresponding business flow.



### To deploy the undo business flow

1. In the Project Navigator, right-click the **Undo\_Orders\_D2F\_BF** business flow and select **Deploy** from the pop-up menu.
2. Select **Deploy Remotely** and enter values for your Web Console installation. If your Web Console is local, simply enter localhost for the Web Console URL.
3. Click **Next**. The Deploy Environment page displays.
4. Select the environment to which you want to deploy this business flow, for example, Oracle\_OLTP.
5. Click **Next**.
6. Click **Finish**.

**NOTE:** You may have to wait a few minutes before the Deployment Finished dialog appears.

## Run the business flow

After your business flow is successfully deployed, you are ready to run it. You can run jobs from the Web Console or the command line. For this tutorial, run the job through the Web Console.

### To run the business flow

1. If the Web Console is not open in your browser, navigate to the URL, for example, `http://localhost:8080/WebConsole`.
2. Log in as the admin user, whose password you set when installing the repository, or as another user with administrative privileges.

**NOTE:** For the purposes of this tutorial, you must be the admin user or a user with comparable privileges in the Web Console. If not, the pages and links described in the steps that follow may not all be visible to you.

3. Click **Business Flow Management** on the Web Console home page.

**TIP:** If you have more than one environment available in your Web Console, make sure that the currently active environment is the one where you plan to run your business flow. To check the active environment and change it if necessary, navigate to **Environment Management** in the Web Console and then set the active environment.

4. Before you launch the business flow, you should confirm that the parameters are set appropriately. In **Tasks**, click **System Parameters > Database to file** to review the parameters and their values.

**TIP:** If the cartridge is of the type indexing, go to **Tasks > System Parameters > Indexing** to review the parameters and their values.

The database to file parameters display first. You can see links to other parameters, such as database to database (if configured), cartridges, or business flows, in the navigation bar on the left.

5. Ensure that **Compression** algorithm is **No compression**.

Notice that Extract file format is set to **CSV normalized** by default, which means that the archive file will be comma separated values (CSV), rather than XML. Leave this setting as is.

6. Click **Apply**.
7. (Optional) Since the business flow will create files on the file system, you might also want to confirm the exact location where Structured Data Manager will create files. To perform this procedure, you need to be the admin user or another user with Manage Environment privileges.
  - a. Click **Environment Management > Environment > Active Environment**
  - b. Click **Locations**.
  - c. Click the **LOCAL\_ARCHIVE\_FS** system parameter. A pane opens at the bottom of the page displaying the settings. Note down the **Directory Expression** property value. This path is where files will be created. You could change this path, but, for the purposes of this tutorial, the default location is fine.

**NOTE:** The Directory Expression typically contains a variable `${archivePath}`. The value of this variable is set in **Settings > File Archive**. Suppose `archivePath` is `C:\SDM_Archive`, the environment is `Oracle_OLTP`, and the Directory Expression value is `${archivePath}/archivedata/${environment}/`. In that case, your archive files are stored in `C:\SDM_Archive\archivedata\Oracle_OLTP`.

**TIP:** Later in this tutorial when you come to [Deploy and run the data access cartridge](#), on page 63, you will learn how to create a new location for your environment.

8. Click **Launch**.

**NOTE:** The undo business flow, `Undo_Orders_D2F_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_D2F_BF` after `Orders_D2F_BF` has run.

9. Click **Orders\_D2F\_BF**. The Launch page for that business flow appears. Notice the two parameters are the ones you created in Designer (Months to retain in production and Minimum eligible rows).
10. Click **Run**.
11. Confirm that you want to run the business flow when prompted.

The business flow is launched and you are taken to a monitoring page that will periodically refresh with the latest status.

12. When the job completes, notice that the business flow failed with an error:

```
Failure in action: action_2
```

```
Exception while executing a Groovy script
```

```
Job run failed, there may be more information available in the Job Monitor.
```

```
Recover groupID 1 using the recovery script
```

The job failed because of the error we intentionally added in Designer to the interrupt condition. To determine the precise nature of the error, you need to examine the log file.

13. In a text editor of your choice, open `..\obt\log\Oracle_OLTP\Oracle_OLTP_run_x.log`, where x is the number of the run. Look for errors. You should find an error message like one of the following in the log:

```
For Oracle:ORA-00942: table or view does not exist.
```

```
For SQL Server:Invalid object name 'DRIVING_TABLE_SEL'.
```

```
For Sybase:Invalid object(table) name DRIVING_TABLE_SEL
```

**TIP:** Another good place to look for information about errors is in the `obt.log` located in the `..\obt\log` subdirectory where you installed the Structured Data Manager.

14. Close the open log files when you are done.
15. Back in the Web Console, to return to the main entry for the job, go to **Tasks > Monitoring > Overview**. Notice that the business flow's status is failed.
16. Click the business flow name, **Orders\_D2F\_BF**, in the status table.
17. Click **Cancel Job** and confirm that you want to cancel the job. This step is necessary because canceling the job performs some cleanup operations on the failed job. If you do not cancel the job first, you may have problems later when you try to re-deploy the corrected business flow and its associated cartridge.
- It may take some time for the job to be canceled.
18. Once the job is in CANCELLED status, return to Designer.
19. In the business flow, double-click **Exit Successfully**.
20. Modify the FROM clause by changing FROM DRIVING\_TABLE\_SEL to FROM ORDER\_HEADER\_SEL. Your corrected script should look as follows:

```
import groovy.sql.*

String query = ""

SELECT COUNT(*) CNT
FROM ORDER_HEADER_SEL

""

row = INTF_DB.firstRow(query)

println row.CNT
```

```
if (row.CNT > Min_Selected_Rows)
    return false;
else
    return true;
```

21. Click **OK**.
22. Re-deploy the business flow following the previous instructions in [Deploy the business flow, on page 54](#). Because the business flow was previously deployed, this time in the Deployment Assistant you will see an additional page, Job History Retention, which gives you the option of dropping or retaining job history for Orders\_D2F\_BF.

For the purposes of this example, assume that you need to retain the previous job history and leave **Drop job history of existing Business Flows** unchecked and click **Next** and finish deploying the business flow.

23. Return to the Web Console.
24. Launch the business flow as you did before. When the job completes this time, it should have done so successfully.
25. Click **Monitoring** from the menu at the top of the page to confirm that the job completed successfully.
26. In a text editor of your choice, open `..\\obdt\\log\\Oracle_OLTP\\Oracle_OLTP_run_x.log`, where x is the number of the run. You should find a line indicating the number of rows that were selected for archiving, for example:

```
2014-06-12 16:40:40,060: INFO: Terminate Success thread 0 : Terminate Success:
7293
```

27. Close the log file.
28. Confirm that the archive XML files were created in the `..\\archivedata\\<env_name>` subdirectory where you installed Structured Data Manager.
29. Return to the Web Console and go back to Launch.
30. Click the business flow, **Orders\_D2F\_BF**.
31. Run the business flow again. When the job completes this time, it should have done so successfully but with far shorter history.
32. Click **History** in the navigation bar on the left. Notice how this run of the business flow stopped after the Archive Selection Step. Because the previous run of the business flow archived all of the eligible rows and purged them from the database, this second run did not meet the interrupt condition (rows selected for archiving did not exceed 100). Therefore, the business flow exited successfully and stopped processing. Zero rows were processed.

Optional Exercise:

33. Return to Designer and run preview on the Orders\_D2F cartridge. You should no longer see any rows in black (eligible for archiving) in the preview pane. All of the eligible rows have been removed from the source database. Hence, you only see rows in red (excluded from archiving).

## Summary and next steps

In this section you learned to:

- create an environment and user in the Web Console
- deploy your business flow
- find the generated business flow and its PDF documentation
- run the business flow from the Web Console
- monitor the execution of the business flow from the Web Console
- troubleshoot a problem in the execution
- revise the business flow in Designer to fix the problem
- re-deploy the business flow and run it again

Once your business flow runs successfully, you can access the XML through SQL and MS Excel by following the instructions in [Query the archive, on page 62](#).

# Chapter 8: Query the archive

After your data has been archived to files, you can access the data in the following ways:

- Directly access your archived data with the Archive Query Server (AQS) cache. Structured Data Manager caches your archived data for query access in PostgreSQL. You can use SQL reporting and development tools against the cache on Windows and UNIX. You can also join your data with existing data in a database.
- Use the data access cartridge to view your data.

This section describes how you can use these two methods to access your data.

- [Configure the AQS cache](#)
- [Query your archived data](#)
- [Deploy and run the data access cartridge](#)
- [Run the undo business flow](#)
- [Summary and next steps](#)

See also: *OpenText™ Structured Data Manager Runtime Guide* for more information about AQS.

## 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.

### To configure the AQS cache

1. If the Web Console is not open in your browser, navigate to its URL, for example, `http://localhost:8080/WebConsole`.
2. Log in as the admin user, whose password you set when installing the repository, or as another user with administrative privileges.
3. On the Web Console home page, click **Environment Management > Environment > Active Environment > Locations**.
4. Click **AQS\_Cache**.
5. 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.
6. 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.

7. Go to **Business Flow Management > Tasks > System Parameters > Database to File**.
8. Scroll to the **AQS Cache Options** section.
9. Set **Enable AQS Cache** to true.
10. 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*.
11. Since you have already run your archive, you must refresh the cache to populate it with all of your archived records.
12. Go to **Environment Management > Environment > Active Environment > Locations**.
13. Click **AQS\_Cache**.
14. Click **Refresh All Cache**.

**NOTE:** **Refresh All Cache** will not work if the archived data is in Content Manager as part of a Structure Records Management Solution (SRMS). In that case, you would need to migrate the archived data from SRMS.

## Query your archived data

Once your archived data is cached, you can query through any SQL client. For example, you could query it using psql as follows:

**NOTE:** This example assumes that you chose driving table schema for the target schema name parameter.

```
psql.exe -h localhost -p 5432 -U postgres -d postgres
```

Enter the password when prompted.

```
select "ORDERID", "ORDERDATE" from dbo."ORDER_HEADER";
```

If you want to create an MS Excel spreadsheet with selected archived data, you could use a command similar to the following:

```
COPY (select "ORDERID", "ORDERDATE" from dbo."ORDER_HEADER")TO E'C:\\myfile.csv'  
(format csv);
```

myfile.csv can be opened in MS Excel and further manipulated as you wish.

## Deploy and run the data access cartridge

Loading your data into MS Excel and manipulating it is often useful. However, in some cases, your users may prefer lighter weight access to the data. You can provide it to them by deploying the data access cartridge that you created in the previous section, [Create a data access cartridge, on page 28](#)

### To deploy your data access cartridge

1. In Designer in the Project Navigator, right-click the **Orders\_Simple\_DA** cartridge and select **Deploy** from the pop-up menu.
2. Select **Deploy Remotely** and enter values for your Web Console installation.
3. Click **Next**. The Target page displays.
4. Choose the environment to which you want to deploy this business flow, for example, `Oracle_OLTP`.
5. Choose **AQS\_CACHE** and **OLTP\_DB**. The cartridge will union the records from source and archive when you run it.
6. Click **Finish**.
7. Go back to the Web Console home page by clicking the OpenText logo in the upper left corner.
8. Click **Data Access** to see a list of deployed data access cartridges.
9. Click **Orders\_Simple\_DA** to run it.
10. Enter a value for salesrep between 1 and 50.
11. Click **Run**.

**TIP:** Use the **csv**, **xml**, **html**, and **pdf** links to download a concatenation of the details for each individual result in the list. That is, the download is not just the summary list but all of the detail record as well.

## Run the undo business flow

Now that you have successfully run your business flow, and obtained and analyzed its data through Excel and your data access cartridge, you can return the source database to its original state by running the undo business flow. Returning the data to the source database will also enable you to proceed with the other tutorials available in the OpenText™ Structured Data Manager documentation set. See the *OpenText™ Structured Data Manager Developers Guide* and *OpenText™ Structured Data Manager Runtime Guide* for those more advanced tutorials.

### To run the undo business flow

1. Return to the Web Console.
2. In **Business Flow Management**, click **Tasks > Monitoring > Overview**.
3. Click the name of the completed business flow, **Orders\_D2F\_BF**. It will be the run that shows some number of rows in the Rows column. Remember that the first run failed and was canceled and the last run, mentioned in step 31, archived no rows because there were not enough eligible rows.
4. Click **Undo**.



5. Confirm that you want to perform the undo. It may take some time for the undo business flow to complete.
6. To confirm that the archived rows have been returned, you can query your source database using an ad hoc query tool or return to Designer and run preview on the Orders\_D2F cartridge. Recall that, after you archived, no eligible rows appeared in black anymore because all eligible rows had been archived and removed from the source database. After successfully running undo, you should now see some number of rows in black (eligible for archiving).

## Summary and next steps

In this section you learned to:

- configure the AQS cache
- import your data into MS Excel
- run the undo business flow

You have now completed the basic Structured Data Manager tutorial. The Structured Data Manager documentation set contains a number of other tutorials that you may wish to try out now:

### *OpenText™ Structured Data Manager Developers Guide*

- Rapid Access to Retired Data
- Advanced Data Masking
- Eligibility Analytics
- Database to Database Archive and Reload

### *OpenText™ Structured Data Manager Runtime Guide*

- Configuring the Web Console
- Deploying and Running Business Flows
- Retiring Applications
- Archiving to Vertica
- Viewing Eligibility Analytics

# Glossary

## **active database**

The database from which you plan to move or copy data. Typically, this database is your online transaction processing (OLTP) or production database. In a two- or three-tiered configuration, the active database resides on tier one and is the source for data movement operations.

## **active environment**

The Web Console views and acts upon only one environment at a time, the active environment. To switch the active environment, you use the Change Active option in the Web Console.

## **activity**

In Designer, a component of a business flow, which is added by using the toolbar. For example, you can add archive and reload activities to your business flow. Note, activities in a business flow are different from what you see at runtime and therefore do not necessarily map directly to what you see in the Web Console.

## **advanced selection**

A method of data selection that discovers all of the interrelated rows from multiple tables and conceptually places them in the same application partition for archiving.

## **annotation**

In Designer, a comment associated with the project, or one of its objects or components. These comments are collected and published in a PDF file when you right click a project or business flow and select Generate Documentation.

## **application partitioning**

The concept of partitioning related rows together during data selection, regardless of whether they are in one or more tables. Application partitioning is unique to Structured Data Manager and contrasts with the more common table partitioning offered by the

database management software, which only groups related rows from one table.

## **archive data store**

The location where the data is to be archived. Can be a separate database, separate space on the same database, or an XML file. In a two-tiered configuration, the archive data store resides on tier two and can be a database or XML. In a three-tiered configuration, the archive data store is a database on tier two and XML on tier three, and is the target for data movement operations.

## **archive query server**

The component that provides SQL access to XML database archives.

## **Consolidated Archive**

A managed, scalable repository that consolidates electronic communications, attachments, and files, and provides complete control over corporate information assets, facilitating compliance with internal corporate governance policies and procedures as well as externally mandated laws and regulations.

## **business flow**

A series of activities, such as archive operations and scripts, that run in sequence. You build business flows in Designer.

## **business flow status**

The Web Console shows the last run of each business flow. The states are Complete/Error/Running.

## **cartridge**

An instance of model- or schema-based eligibility criteria used to move or copy data from one location to another. Cartridges capture the application and business rules to ensure referential integrity of the data. For any one model in your project, you may have many cartridges that use it.

## **chaining table**

The lower level table in a many-to-one or a many-to-many relationship between higher level and lower level tables in the model hierarchy.

**classification**

The Content Manager (formerly Records Manager) classification to be applied to the data moved by Structured Data Manager. This classification specifies where to place the data when it is ingested by Content Manager. For more information, see the Content Manager documentation.

**collection**

The configuration of a directory location and file pattern to match a set of archived XML files, thus allowing SQL access to the archived data.

**comma separated values (CSV)**

A database-to-file output format that stores the data as values separated by commas and a metadata file. Each line in the CSV file corresponds to a row in a table. Within a line, fields are separated by commas, each field belonging to one table column. CSV files provide a simple format that many applications can import.

**command**

Command files or JavaScript files launched by the Web Console on your behalf with status displays.

**condition**

In Designer, the way you branch your business flow to run or skip an activity based on some criteria.

**configuration parameter**

A type of parameter that has its values set by an administrator (someone who has repository privileges from the Web Console) through the administrator interface. Typically, this type of parameter represents values that should be changed very infrequently, perhaps only at deployment time.

**console user**

The Web Console identifies individual users, who are distinct from database users. The properties for a user are User Name, Full Name, Password, Enabled, Description, Email, Phone, and Privileges.

**console user name**

The login name associated with a Web Console user.

**constraint**

A column or a list of columns that enables you to identify rows in the database and relate them to one another.

**custom properties**

User-created name/value pairs in cartridges and business flows. These values are exposed at runtime as parameters.

**customization**

A change that an administrator or DBA makes to a project provided by a third party, typically for a packaged application like Oracle PeopleSoft or Oracle E-Business Suite. As long as the customization is allowable by the project, the user can merge the customization into newer revisions of the third party project.

**customization mode**

A Designer mode that provides visual cues to indicate customizations in the model. In a project with locked files, customization mode is on by default, but you can toggle it on and off from the toolbar in the model editor.

**data access cartridge**

A cartridge that provides lightweight query access to retired or archived data. Data access cartridges are designed by the archive developer but can be run by business users with no technical expertise.

**data masking**

The process of replacing private or confidential data during movement with a specified mask. You can choose from pre-defined masks that are part of OpenText or create your own mask. A mask may or may not be reversible upon reload from the archive data store.

**data movement**

The method used by Structured Data Manager to actually move data (transactional, bulk or partitioned for database to database, and copy or archive for database to file).

**data transparency**

The ability to access archived data through your standard application interfaces for data access. Data transparency enables users to access archived data as though it were still in the active database.

**database constraint**

A constraint that exists in the database and can be discovered and referenced from Designer.

**database to database**

A movement in which data goes from an active database to an archive database, or separate tablespaces inside the active database. Typically, the archive database is located on cheaper storage devices.

**database to file**

A movement in which data goes from an active database to a file (XML, JSON or CSV format), which is offline but still accessible through SQL using the archive query server and a client tool of your choice.

**deployment assistant**

The user interface component within Designer used to deploy or generate business flows.

**description**

A technical description created by the developer for her own reference. These descriptions do not appear in the generated PDF file for the cartridge or business flow.

**Designer**

The user interface component used to develop, test, and deploy your archiving solution. Designer is a powerful graphical development environment for archive solutions.

**distributed instance**

A configuration option for database-to-database archiving where the data you archive is stored on a separate database from the source or active database.

**DRE**

See [Dynamic Reasoning Engine \(DRE\)](#).

**driving table**

A driving object is a root of a model hierarchy. Its relationship to the child tables drives the selection of transactions.

**dynamic list of values**

A list of values for a parameter that obtains its members from a SELECT statement that returns identifiers and labels.

**dynamic parameter**

A type of parameter that has its value set by a Groovy script that runs at deployment time to obtain a value. For example, this type of parameter can supply the type or version of a database or application, which can be obtained programmatically at deployment time.

**Dynamic Reasoning Engine (DRE)**

A platform technology that uses high performance pattern-matching algorithms to search for content stored in OpenText repositories. Performs core information operations for contextual analysis and concept extraction, enabling solutions for the categorization, summarization, personalization, hyperlinking, and retrieval of all forms of information.

**environment**

The source and (optional) target credentials against which you plan to run commands. You can define multiple environments within your installation to identify various source and target databases.

**error**

One of the ways in which you can interrupt a business flow. Error indicates that the business flow failed for some reason.

**exclusive rules**

One of the ways in which Structured Data Manager determines whether to include or exclude rows from the archive operation. Exclusive rules require all rows in the constraint table to match for inclusion. Exclusive rules exclude the instance if the condition on any child is false, like STATUS='CLOSED'.

**exit**

One of the ways in which you can interrupt a business flow. You can exit successfully or with a warning.

**export**

The way that you save an Structured Data Manager project to an exchange format (.hdp) from the File menu. See also *import*.

**export data**

The way that a user can send data to CSV format from Preview using the toolbar item.

**generate documentation**

The process of collecting and grouping all annotations into a PDF file that also describes the business flow or cartridge structure.

**history schema**

For database-to-database archiving, the schema in the target database where the archived data is stored.

**Knowledge Discovery (IDOL)**

See [Knowledge Discovery \(Intelligent Data Operating Layer \(IDOL\)\)](#), below.

**import**

The way that you transfer projects from exchange format (.hdp) into the Project Navigator. You can also use import to migrate cartridges created in 5.1 to 6.x. See also *export*.

**inclusive rules**

One of the ways in which Structured Data Manager determines whether to include or exclude rows from the archive operation. Inclusive rules require only one row in the constraint table to match the rule and be included. Inclusive rules include the instance if the condition on any child is true, like `PRODUCT_RECALLED='Y'`.

**indexing cartridge**

A cartridge that indexes your data for better searching. For example, you might associate an indexing cartridge with a database-to-file archiving cartridge to improve performance when querying the archive data files.

**Knowledge Discovery (Intelligent Data Operating Layer (IDOL))**

An information processing layer that collects indexed data from connectors and stores it in a structure optimized for fast processing and retrieval, integrating unstructured, semi-structured, and structured information from multiple repositories.

**interrupt**

The way to stop or pause a business flow (pause, error, exit with warning, exit successfully).

**local deployment**

The generation and deployment of your cartridge or business flow to an environment on your local, Designer client. Deployment files are generated locally and then deployed to the designated, local environment.

**lookup table**

A table that contains helpful non-transactional information. For example, non-transactional information could be status definitions, or the name of the sales representative.

**managed table**

A table in the model that is copied and then purged from the active database by a cartridge. Transactional, chaining, and driving tables in a model are all typically managed tables.

**model**

A model identifies the tables and table relationships representing a business entity or related business entities. A project can have multiple models. Each model contains a driving table and all of its child and descendent tables.

**model compatibility**

Each model in your project can have one or more dynamic parameters associated with it to verify the compatibility with the target environment. If the compatibility parameter returns false, then the cartridge referencing the model will not deploy or run and throw an error. For example, the script could return false for Oracle 10.2 and true for Oracle 11.1 to indicate that a cartridge referencing the

model can only deploy and run against Oracle 11.1.

**model-based cartridge**

A cartridge that moves data based upon a defined data model with relationships. This type of cartridge is typically used for ongoing archive operations.

**non-intrusive environment**

In a non-intrusive environment, data is archived without an interface schema and a generic JDBC driver is used. A non-intrusive environment enables you to copy or archive data from read-only sources, which is especially helpful in cases where the data is associated with older technologies that might not support basic SQL statements or when the database administrator or company policy prohibits write access to the production environment.

**OLTP database**

The online transaction processing database that typically is your active or source database.

**pause**

One of the ways in which you can interrupt a business flow. Pausing suspends the business flow while awaiting operator intervention.

**Content Manager**

Enterprise document and records management software designed to simplify the capture, management, security, and access to information. Content Manager enables organizations to more easily comply with regulations and corporate policies, and it helps secure information from inappropriate access and misuse.

**reload**

The act of taking data from an archive data store and loading it into the active database.

**remote deployment**

The generation and deployment of your cartridge or business flow to an environment on a system that is remote from your Designer client. Deployment files are generated locally and then deployed to the designated, remote environment.

**repository**

The location that holds business flow metadata, product configuration data, and data collected during runtime. The repository can be located on your active database or another logical database.

**rule**

Qualifications added to the model in order to include or exclude data based on certain criteria. For example, you might add a rule to exclude from archiving any orders that are not yet closed.

**runtime parameter**

A type of parameter that has its values set by the operator executing the job in Console or on the command line. Typically, this type of parameter represents operational values that tend to change frequently and therefore need to be set each time the job is run.

**schema-based cartridge**

A cartridge that moves data based upon the database schema rather than a defined data model with relationships. This type of cartridge is typically used for database retirement or the cleanup of orphan tables.

**selection**

The form of data selection to use (standard or advanced) for choosing data. When deploying a cartridge or adding it to a business flow, you must specify the selection method.

**single instance**

A configuration option for database-to-database archiving where the data you archive is stored on the same database (Oracle) or the same server (SQL Server) as the source or active database.

**source**

The location (database) from which you are copying or moving data.

**SQL access server**

See *archive query server*.

**standard selection**

A method of data selection that restricts itself to the rows identified by the model. Unlike

advanced selection, it does not attempt to traverse related rows across multiple tables.

**structured records management**

A type of solution that extracts structured data from a source application and moves it into XML format. The XML is then ingested into the corporate records management system for long term management and eventual disposal according to corporate policy.

**table use**

A database table, view, or synonym that is referenced in Designer, for example, in the model. The same table can be used multiple times in a model. For example, a table could be appear as a transactional table and a lookup table in the same model.

**target**

The location (database or XML) to which you are copying or moving data.

**tier**

A level in your database archiving configuration. You can have two- or three-tiered configurations. In a two-tiered configuration, tier one contains your active database and tier two your archive data store, which can be a database or XML. In a three-tiered configuration, tier one contains your active database, tier two an archive database, and tier three XML.

**transactional data movement**

Transactional movement uses set-based data movement and is the default method of movement.

**transactional table**

A table that contains information about the business transaction. For example, a transactional table might contain detailed tax or payment information related to each business transaction.

**unique identifiers (UIDs)**

A 16 hexadecimal identifier calculated based on the content of a Designer file. This value is used to determine if the user has customized key pieces of a project.

**unmanaged table**

A table in a model that is copied but not purged from the active database by a cartridge.

Lookup tables in a model are typically unmanaged tables.

**Vertica**

Column-oriented SQL database management software for storing and analyzing structured data. Used to manage large, fast-growing volumes of data and provide fast query performance for data warehouses and other query-intensive applications.

**virtual constraint**

A constraint that you define in Designer that only exists within Structured Data Manager.

**Web Console**

A browser-based interface where you can create and manage your deployment environments, and deploy, run, administer, and monitor your business flows.

# Send documentation feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on OpenText OpenText™ Structured Data Manager 25.3.0 Tutorial**

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [swpdl.sdm.docfeedback@microfocus.com](mailto:swpdl.sdm.docfeedback@microfocus.com).

We appreciate your feedback!