

# Structured Data Manager

Software Version 7.66

## Installation Guide



Document Release Date: February 2022  
Software Release Date: February 2022

## Legal notices

### Copyright notice

© Copyright 2017-2022 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) 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. Micro Focus 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 Micro Focus 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:

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

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the [Access Levels descriptions](#).

# Contents

About this document .....	5
Prerequisites .....	5
Related documentation .....	5
 Chapter 1: Prerequisites .....	 7
Obtain Structured Data Manager documentation .....	7
Structured Data Manager installation overview .....	7
 Chapter 2: Requirements .....	 9
Operating system and server requirements .....	9
Supported operating systems and browsers .....	9
Server requirements .....	9
Database to file supported storage locations .....	10
Database requirements .....	10
DB2 database requirements .....	11
Oracle database requirements .....	11
Source database preparation for Oracle .....	12
Target database preparation for a distributed instance with Oracle .....	13
Database preparation for a single instance with Oracle .....	14
Microsoft SQL Server requirements .....	14
Distributed archive requirements .....	15
Sybase requirements .....	15
JDBC URL requirements .....	16
Repository preparation .....	16
Repository preparation for Oracle .....	16
Repository preparation for PostgreSQL .....	17
Optional feature requirements .....	17
Custom target schema or database .....	17
Create the target schema or database .....	18
Create target tables .....	19
 Chapter 3: Installation .....	 20
Install Structured Data Manager .....	20
Install the software using scripted installation .....	22
Manage OBT_HOME .....	23
Use the log files .....	24

View the installation software log file .....	24
Edit the logging properties .....	25
Install Discovery .....	25
Moving the Structured Data Manager installation to a different server .....	26
Chapter 4: Remove Structured Data Manager .....	28
Reinstall the repository .....	28
Appendix .....	30
Copy additional jar files .....	30
Glossary .....	31
Send documentation feedback .....	40

## About this document

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

This guide provides information about:

- meeting installation prerequisites
- installing Structured Data Manager
- creating installation scripts

## Prerequisites

Prerequisites for installing this product include:

- knowledge of operating systems
- database knowledge
- application knowledge

## Related documentation

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

Document Name	Description
<i>Structured Data Manager Troubleshooting Guide</i>	Explains how to diagnose and resolve errors, and provides a list of common errors and solutions.
<i>Structured Data Manager Tutorial</i>	Provides step-by-step instructions to build a sample archiving module, deploy, run, and troubleshoot errors in it.
<i>Structured Data Manager Upgrade Guide</i>	Explains how to upgrade the product and archive schema generated by the earlier versions of the product.
<i>Structured Data Manager Discovery Guide</i>	Explains the purpose, how to install and use Discovery.

# Chapter 1: Prerequisites

Before installing Structured Data Manager, you should make sure that you have the latest documentation, and that you understand the overall installation process.

This section includes:

- [Obtain Structured Data Manager documentation](#)
- [Structured Data Manager installation overview](#)

## Obtain Structured Data Manager documentation

The latest documentation for Structured Data Manager can be found on Micro Focus Software Support Online.

## Structured Data Manager installation overview

Before you can use Structured Data Manager to archive your data, you need to install the software, create the repository, create your environments, set up the database archiving components, and deploy the business flows that archive your data.

Installing Structured Data Manager consists of the following tasks:

1. Review the *Structured Data Manager Concepts Guide* to choose the archive solution that is right for you.
2. Choose where to install your repository.

Your repository can be installed in the source database or another database.

See [Repository preparation, on page 16](#)

3. Make sure that the necessary requirements for your servers, databases, and selected archiving solution have been met.

See also:

- [Operating system and server requirements , on page 9](#)
- [Database requirements, on page 10](#)

Related information: *Structured Data Manager Certification Matrix*

4. If you are upgrading from a previous installation, see the *Structured Data Manager Upgrade Guide*.
5. Install the Structured Data Manager software.

See [Install Structured Data Manager , on page 20](#)

6. Start the Web Console to create the repository and the environment.

Related information:

- *Structured Data Manager Runtime Guide*
- *Structured Data Manager Upgrade Guide*

7. Use the *Structured Data Manager Developer's Guide* to design and develop or customize your archive solution.
8. Use Designer or the Web Console to deploy the business flows you have created.

Related information: *Structured Data Manager Runtime User Guide*

9. To remove the Structured Data Manager software, see [Remove Structured Data Manager](#) , on [page 28](#).

# Chapter 2: Requirements

Structured Data Manager has the following requirements for installation. Requirements are dependent on the database you are archiving from, and how you plan to archive your data.

See *Structured Data Manager Concepts Guide*

**TIP:** Check to see that you have the most recent version of this manual before beginning the installation. See [Obtain Structured Data Manager documentation, on page 7](#).

This section includes:

- [Operating system and server requirements](#)
- [Database requirements](#)
- [Repository preparation](#)
- [Optional feature requirements](#)

## Operating system and server requirements

Make sure you meet the operating system and server requirements before installing the Structured Data Manager software.

This section includes:

- [Supported operating systems and browsers, below](#)
- [Server requirements, below](#)
- [Database to file supported storage locations , on the next page](#)

## Supported operating systems and browsers

For the most recent information on supported operation systems and browsers, contact Micro Focus Support for the current version of the *Structured Data Manager Certification Matrix*.

## Server requirements

For all servers required for Structured Data Manager:

- Identify a directory with at least 2 GB of disk space to install the Structured Data Manager.

Platform	Use
UNIX	/user/home/username
Windows	C:\

- To run Structured Data Manager 16 GB of RAM is required.
- To run Structured Data Manager jobs on a server, the server needs:
  - an operating system user with database access from the command line.
  - connectivity to the required databases, for example, the source database and repository.
  - for database to file archiving, the operating system user must have write permissions to the file system.
  - for archiving to a remote system , see the necessary steps in the *Structured Data Manager Developer's Guide*.

## Database to file supported storage locations

Structured Data Manager supports the following storage locations for database to file archiving:

- Consolidated Archive On-Premise
- Amazon Simple Storage Service
- Filesystem
- SSH (Secure Shell)
- Content Manager
- HDFS ( Hadoop File System)

Related information: Managing Environments of the *Structured Data Manager Runtime Guide*

## Database requirements

Structured Data Manager archives data from the following:

- DB2, see [DB2 database requirements , on the next page](#)
- Oracle, see [Oracle database requirements , on the next page](#)
- Microsoft SQL Server, see [Microsoft SQL Server requirements , on page 14](#)
- Sybase, see [Sybase requirements , on page 15](#)
- JDBC URL with compatible driver, see [JDBC URL requirements , on page 16](#)

Depending on the archive method you are using, there are different requirements.

Archive Method	Required
Database to File	<ul style="list-style-type: none"><li>• a repository to store archiving metadata</li><li>• a source database from which to archive eligible data (also referred to as the active database)</li></ul>
Database to Database	<ul style="list-style-type: none"><li>• a repository to store archiving metadata</li></ul>

Archive Method	Required
	<ul style="list-style-type: none"><li>• a source database from which to archive eligible data (also referred to as the active database)</li><li>• for distributed archive only, a target database in which to store archived data (also referred to as the archive database)</li></ul>

The repository can be located on an Oracle or PostgreSQL database installed with the Structured Data Manager software.

See also:

- [Repository preparation, on page 16](#)

Make sure the appropriate database requirements are met before installing the Structured Data Manager software.

**NOTE:** The source database, repository, and target database must use the same character encoding. If the character encoding is not the same, a warning is displayed in the installation summary.

## DB2 database requirements

For the source and target databases information contact Micro Focus Support for the current version of the *Structured Data Manager Certification Matrix*.

- To connect to DB2 you must have permission to create packages.
- For DB2, complex (structured) UDTs are not supported.
- For distributed instances on DB2, you must enable federation on both source and target databases. You also need to create a user mapping between the interface and relocation user so that each can access the federation (dblink). Additionally, the archive access user and the relocation user must be mapped for the federation. Refer to your DB2 documentation for more information about setting up federation.

An administrator account with the following privileges is required:

- create user
- create schema
- grant permissions to non-owned tables
- read database metadata tables/views

## Oracle database requirements

The source and target databases must be the same database version. For the supported Oracle database versions contact Micro Focus Support for the current version of the *Structured Data Manager Certification Matrix*.

An administrator account with the following privileges is required:

- create user
- grant permissions to non-owned tables
- read database metadata tables/views

This document refers to this account as the system account, but it can have any name you assign to it.

This section includes:

- Source Database Preparation for Oracle
- Target Database Preparation for a Distributed Instance with Oracle
- Database Preparation for a Single Instance with Oracle

### Source database preparation for Oracle

1. Make sure there is 3 GB available disk space to create tablespaces.

**TIP:** When you calculate the space requirements, you need to consider how much space you need for rollback segments and temporary segments.

2. Create the following data and temporary tablespaces.

Tablespace	Size	Contains
INTF_DATA	1 GB	Table data for the interface schema.
INTF_TEMP	1 GB	Temporary data for the interface schema.
AA_DATA	25 MB	Table data for the archive access build schema. Create on either the source or target database.
AA_TEMP	25 MB	Temporary data for the archive access build schema. Create on either the source or target database.

Example:

```
create tablespace INTF_DATA datafile '<path>/INTF_DATA_01.dbf' size 1G autoextend on;
```

where *<path>* is the directory location where you want to save the tablespace datafile.

The tablespaces that you create here are available for selection from the Web Console when you deploy the product and business flows.

3. If you are using the distributed archive configuration option, add an entry in `tnsnames.ora` for the target database.
4. Add the following entries to the source database `init.ora` file:

```
_PUSH_JOIN_UNION_VIEW = TRUE  
COMPATIBLE = <current_database_release>
```

## Target database preparation for a distributed instance with Oracle

The target or archive database is required for the distributed archive configuration option.

**NOTE:** Make sure the procedures in this section are complete before you create the repository.

1. Create a target database with the following configuration:

- Default LARGE `init.ora` parameters

The `init.ora` parameter values depend upon the size of the installation, and are recommended by Micro Focus service representatives

- 8 K for the `db_block_size` parameter, or the same as the source database
- `NLS_LANGUAGE` character set must be the same as the source database

2. Create or expand the following tablespaces:

Tablespace	Size	Contains
HIST_DATA	4 GB	Table data for the history schema. This tablespace must be in the target database for a distributed instance.
HIST_TEMP	4 GB	Temporary data for the history schema. This tablespace must be in the target database for a distributed instance
HIST_INDX	4 GB	Indexes on the archived data. This tablespace must be in the target database for a distributed instance
RELOC_DATA	2 GB	Table data for the relocation schema.
RELOC_TEMP	2 GB	Temporary data for the relocation schema.
AA_DATA	25 M	Table data for the archive access schema. Create on either the source or target database.
AA_TEMP	25 M	Temporary data for the archive access schema. Create on either the source or target database.

Example:

```
create tablespace HIST_DATA datafile '<path>/OBT_DATA.dbf' size 1GB autoextend on;
```

where *<path>* is the directory location where you want to save the tablespace datafile.

**NOTE:** For tablespaces in the History schemas, Micro Focus recommends the data files associated with the tablespaces are set for “autoextend”, as the need for space increases over the life of the archive.

The tablespaces that you create here are available for selection from the Web Console when you deploy the product and business flows.

3. Add an entry in `tnsnames.ora` for the source database.
4. Make sure the `db_domain` of the target database is the same as the `db_domain` of the source database.

If the `db_domain` is different, set `global_names` to `FALSE`.

### Database preparation for a single instance with Oracle

These tablespaces must be in the source database for single instance.

1. Create or expand the following tablespaces in the source database

Tablespace	Size	Column
HIST_DATA	4 GB	Table data for the history schema.
HIST_TEMP	4 GB	Temporary data for the history schema.
HIST_INDX	4 GB	Indexes on the archived data.

Example:

```
create tablespace HIST_DATA datafile '<path>/HIST_DATA_01.dbf' size 1G autoextend  
on;
```

where *<path>* is the directory location where you want to save the tablespace datafile.

The tablespaces that you create here are available for selection from the Web Console when you deploy the product and business flows.

2. Add the following entries to the source database `init.ora` file:

```
_PUSH_JOIN_UNION_VIEW = TRUE  
COMPATIBLE = <current_database_release>
```

### Microsoft SQL Server requirements

For the supported source and target SQL server and databases contact Micro Focus Support for the current version of the *Structured Data Manager Certification Matrix*.

An administrator account with the following privileges is required:

- Alter
- Alter any database
- Alter any database DDL trigger
- Alter any database event session
- Alter any event session
- Alter any linked server
- Alter any login

- Alter any role
- Alter any schema
- Alter any user
- Control
- Create any database
- Create procedure
- Create sequence
- Create synonym
- Create table
- Create view
- Delete
- Execute
- Insert
- Select
- Update
- View any database
- View any definition
- View definition

This document refers to this account as the sa account, but it can have any name you assign to it. This account is necessary to enable connections using JDBC.

### **Distributed archive requirements**

For the database to database distributed archive option jobs to run, you must enable the distributed transaction coordinator.

1. On Windows, open the Control Panel.
2. Double-click **Administrative Tools**.
3. Double-click **Services**.
4. Start the Distributed Transaction Coordinator.

For distributed archive, the maximum file size for the XML\_TYPE data type is 8K.

### **Sybase requirements**

For the supported Sybase versions contact Micro Focus Support for the current version of the *Structured Data Manager Certification Matrix*.

An administrator account with the following privileges is required:

- create database
- create login
- create user
- read database metadata tables/views
- grant permissions to non-owned tables

**NOTE:** Sybase is supported for database to file only.

This document refers to this account as the sa account, but it can have any name you assign to it.

## JDBC URL requirements

You can use any database to which you can connect through JDBC as a source or target database, provided that your JDBC database driver supports the necessary functions for Structured Data Manager to operate.

## Repository preparation

Structured Data Manager requires a repository to store archiving metadata. The repository can be located on one of the following locations:

- an Oracle database
- a PostgreSQL database

The same repository is used for all of your database to file and database to database environments, regardless of the source database.

This section includes:

- [Repository preparation for Oracle , below](#)
- [Repository preparation for PostgreSQL, on the next page](#)

### Repository preparation for Oracle

The Oracle repository should reside on an existing database where backups are regularly performed.

1. Identify a persistent database meeting the following requirements:

**Available tablespace**—3200 MB

**Available diskpace**—100 MB

2. Create the following tablespaces:

Tablespace	Size	Contains
REP_DATA	800 MB	Table data for the repository schema.
REP_TEMP	800 MB	Temporary data for the repository schema.

Example:

```
create tablespace REP_DATA datafile '<path>/REP_DATA.dbf' size 1GB autoextend  
on;
```

where *<path>* is the directory location where you want to save the tablespace datafile.

The tablespaces that you create here are available for selection from the Web Console when you create the repository.

## Repository preparation for PostgreSQL

You can use PostgreSQL for Windows and Linux platforms as a repository database. PostgreSQL can be installed when you install Micro Focus SDM.

**NOTE:** Only one Micro Focus SDM repository is allowed on a PostgreSQL database. To create more than one repository you need to create another PostgreSQL database. The new PostgreSQL database can be on the same PostgreSQL server.

For PostgreSQL, Micro Focus recommends creating a new database to install the Micro Focus SDM repository. The name preferred for the new database is “obtrepdb”.

As best practice, the PostgreSQL database should not be shared with any other application. For security reasons, do not use the default “postgres” database for the repository installation.

An administrator account with the following privileges is required:

- create database
- create login
- create user
- read database metadata tables/views
- grant permissions to non-owned tables
- optional feature requirements

## Optional feature requirements

Structured Data Manager includes some optional features that you may or may not require for your environment. If you are planning to use any of these features, you need to perform some additional configuration.

This section includes:

- [Custom target schema or database, below](#)

### Custom target schema or database

Database to database archiving requires a target schema for Oracle, or a target database for SQL Server. The target schema or database is created automatically when you deploy the business flow.

**NOTE:** Use the Web Console to create the target schema or database.

The default target schema or database is created using the naming convention <source\_database\_name>\_HIST, where <source\_database\_name> is the name of your source database or schema. Creating your own target schema or database enables you to select a name that matches your existing naming conventions.

When you create your own target schema or database, you can also choose to create your own target tables. This enables you to use tables with partitions in your database to database archiving environment.

If you create your own target schema or database, you must also register the owner mappings.

See “OwnerMapping groovyscript API” in the *Structured Data Manager API Reference Guide*.

If you choose to create your own target schema or database and target tables, perform the steps in the following sections:

- [Create the target schema or database, below](#)
- [Create target tables, on the next page](#)

## Create the target schema or database

### To create the target schema or database

1. For Oracle, create a schema on the target database for each schema on the source database.

For SQL Server, create a target database for each database on the source database server that data will be archived from.

2. Make sure that the schema or database owner has the following permissions:

Oracle	SQL Server
<ul style="list-style-type: none"><li>• CREATE SESSION</li><li>• CREATE TABLE</li><li>• CREATE DATABASE LINK</li><li>• SELECT ANY DICTIONARY</li></ul>	<ul style="list-style-type: none"><li>• CREATE DEFAULT</li><li>• CREATE PROCEDURE</li><li>• CREATE RULE</li><li>• CREATE TABLE</li><li>• CREATE VIEW</li><li>• SELECT ON SYSSRVROLES</li></ul>

3. Run the owner mapping API to make sure that the target schema or database is updated before you deploy any business flows.

Related information:

See “OwnerMapping JavaScript API” in the *Structured Data Manager API Reference Guide*.

### Create target tables

After you have created your own target schema or database, you can use the Web Console to create the target tables.

See *Structured Data Manager Developer's Guide*

If you choose to create the tables manually, you must make sure of the following:

- Each table contains the following columns.

Column Name	Data Type	Value
OBT_SAVED_ROWID	ROWID	not null
OBT_ROW_SEQ	NUMBER	not null
OBT_WF_RUN_ID	VARCHAR2(128)	not null

- New table names match managed table names in the source database or schema.
- All column names, data types, lengths, and null/not null properties for the managed tables are the same.
- If you choose to create the target driving table, it must also contain the following column:

Column Name	Data Type	Value
OBT_TIMESTAMP	DATE	SYSDATE

# Chapter 3: Installation

The Structured Data Manager installation software installs Designer, Archive Query Server, and the Web Console server.

This chapter includes:

- [Install Structured Data Manager](#)
- [Install the software using scripted installation](#)
- [Manage OBT\\_HOME, on page 23](#)
- [Use the log files](#)
- [Install Discovery](#)
- [Moving the Structured Data Manager installation to a different server](#)

## Install Structured Data Manager

Make sure the latest patches are installed with Structured Data Manager.

1. Obtain the Structured Data Manager installation software, and copy the appropriate file for your platform to a location accessible by all the servers and clients in the configuration.

**NOTE:** If you are using a Windows 64-bit platform, then follow the below steps:

- a. Note down the User Account Control (UAC) settings before you start to install the SDM application.
- b. Disable UAC if it is enabled.
- c. Install the SDM application.
- d. After the successful installation, revert to your previous UAC settings.

2. The folder where Structured Data Manager stores application data, such as configuration files and log files is known as home directory or OBT\_HOME. By default, OBT\_HOME is set to C:\SDM\OBTHOME on Windows and /var/opt/OBTHOME on Linux. You can change these default values during **New Install** of the product by setting the OBT\_HOME environment variable prior to installation.

3. Use the appropriate command to start the installation software:

For	Use
UNIX	sh <part_number>.bin (console mode) sh <part_number>.bin -i gui (GUI mode)
Windows	Right click <part_number>.exe and select <b>Run</b> as administrator.

where *<part\_number>* is the installation software for your platform.

4. Click **Next** when the Introduction page opens.
5. Perform one of the following:
  - a. **New Install** - Select this option if you don't have any prior installation of Structured Data Manager or want to have different OBT\_HOME than the existing directory used with prior version of SDM installed on your machine. Click **Next**.
    - i. Select the default location by choosing the default folder where you choose to install, or enter the location for the software to be installed. Click **Next**.

**NOTE:** For Windows, the character length of this directory path cannot exceed 38 characters. For example, C:\SDM, where C:\ counts as three characters. If you need to use a path with more than 38 characters, use the SUBST command to create a substitute drive.

In the documentation, this directory is referred to as the *<install\_directory>*. The *<install\_directory>* is the location where you installed the software.

- ii. Specify the home directory. Click **Next**.
  - b. **Upgrade** - Select this option if you have prior installation of Structured Data Manager and want to continue using the same OBT\_HOME used with prior version of SDM installed on your machine.
    - i. Specify the folder where previous version of SDM is installed. Click **Next**.
    - ii. Follow the step i of 5a above.
    - iii. The home directory of the previous version of SDM installation is displayed. Click **Next**.

**NOTE:** To copy additional jar files, see section [Copy additional jar files, on page 30](#).

6. Choose the **Web Console/Tomcat Listen Port** (default is 8080) and **Web Console/Tomcat Shutdown Port** (default is 8005). Click **Next**.
7. Select the location to create product icons. Click **Next**.
8. Review the Pre-Installation Summary. Click **Install**.

After the software is installed, the optional PostgreSQL Install dialog opens.

9. To open the PostgreSQL Install wizard, select **Install PostgreSQL**.

**NOTE:**

- PostgreSQL is an option for only Linux and Windows users. Additionally, you must be logged on as a superuser in order to install PostgreSQL.
- To install PostgreSQL on Linux, refer to the <https://www.postgresql.org/download/> link.

If you do not wish to install PostgreSQL, select **Don't Install PostgreSQL**. Click **Next**.

10. The Launch Options page opens.

- a. Select the check box to start the Web Console server using the Web Console/Tomcat Listen Port, or manually start the Web Console server from the command line.
  - b. You can also select to launch your default browser with the default Web Console URL when you close the Installer.
11. Click **Next**.
  12. Click **Done** to close the installation software.

**NOTE:** On Windows , you need to run Designer as administrator on first launch. After installation, right-click the Designer icon shortcut and select **Run as administrator**.

13. Proceed to the *Structured Data Manager Runtime Guide* for instructions about using the Web Console to:

- Install and configure the repository, or import a repository from a previous release

**TIP:** If no repository is detected when you go to the Web Console, you will be guided through creating or importing a repository.

- create environments
- deploy the software and business flows

## Install the software using scripted installation

If you plan to install Structured Data Manager multiple times, you can use scripted installation. Scripted installation uses the `install.properties` properties file.

1. Create a text file called `install.properties`.

The `install.properties` file requires the following:

```
INSTALLER_UI=<UI mode>
```

Refer step 5 in [Install Structured Data Manager , on page 20](#) and specify one of the following:

```
INSTALLATION_MODE_1=New Install
```

```
INSTALLATION_MODE_2=Upgrade
```

The following variables are required for both **New Install** and **Upgrade** scenarios:

```
USER_INSTALL_DIR=<location>
```

```
TOMCAT_PORT_1=<listen port>
```

```
TOMCAT_PORT_2=<shutdown port>
```

The following variable is required only for **New Install** scenario:

```
OBT_HOME_1=<home directory>
```

The following variable is required only for **Upgrade** scenario:

```
PREV_VER_INSTALLATION_FOLDER_1=<previous location>
```

Where	Is
<UI mode>	The type of installation. For a scripted installation, the mode should be SILENT.
<location>	The location where you want to install the software.
<previous location>	The location where previous version of SDM is installed.
<home directory>	Home directory i.e. OBT_HOME
<listen port>	Web Console/Tomcat listen port
<shutdown port>	Web Console/Tomcat shutdown port

#### Example

```
INSTALLER_UI=SILENT  
INSTALLATION_MODE_1=New Install  
USER_INSTALL_DIR=C:\\SDM\\SDM766  
OBT_HOME_1=C:\\SDM\\OBT_HOME  
TOMCAT_PORT_1=8080  
TOMCAT_PORT_2=8005
```

2. Save the `install.properties` file.
3. Use the appropriate command to start the installer.

For	Use
UNIX	<code>&lt;part_number&gt;.bin -f install.properties</code>
DOS	<code>&lt;part_number&gt;.exe -f install.properties</code>

where *<part\_number>* is the installation software for your platform.

## Manage OBT\_HOME

The home directory is where Structured Data Manager stores application data, such as configuration files and log files.

To see where your OBT\_HOME is currently located, look in `<install_directory>\obt\config\obt.env.`, where *<install\_directory>* is the location where you installed the software.

For details on changing the home directory, see the *Structured Data Manager Runtime Guide*.

## Use the log files

Use the procedures in this section to view the installation log files and determine what information you want captured.

This section includes:

- [View the installation software log file, below](#)
- [Edit the logging properties, on the next page](#)

### View the installation software log file

The installer captures logging events and appends them to a log file.

By default, the log files are saved and the logging level is set to INFO.

Mode	See
UNIX	<Install_dir>/obt/log
Windows	<Install_dir>\obt\log

#### To view install log files

1. Navigate to the directory containing the log file. For example:  
`<Install_dir>/obt/log/`
2. Open the following log file using a text editor:

File Name	Description
Structured_Data_Manager_Install_<Date_Time>.log	The Structured_Data_Manager_Install_<Date_Time>.log is generated by the installation software, and includes information on the overall installation.

#### To view log files

1. Navigate to the directory containing the log file. For example:  
`<OBT_HOME>/log/`
2. Open one of the following log file using a text editor:

File Name	Description
obt.log	The obt.log file captures all logging information for Structured

File Name	Description
	<p>Data Manager.</p> <p>By default, the UNIX log is stored under <code>obt.log</code> is under <code>&lt;OBT_HOME&gt;/log</code> By default, the Windows log is stored under <code>obt.log</code> is under <code>&lt;OBT_HOME&gt;\log</code>.</p> <p>To change the types of information captured in the log, see <a href="#">Edit the logging properties, below</a>. Multiple log files are numbered in sequence. For example, <code>obt.log1</code>.</p>

## Edit the logging properties

After you have installed the product, you can edit the `log4j2.properties` file to change the logging properties. For example, you can change:

- where the log files are kept
- what information is logged
- the maximum size of the log file
- how much logging information is kept

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

1. Navigate to the directory that contains the `log4j2.properties` file.

Mode	See
UNIX	<code>&lt;OBT_HOME&gt;/config</code>
Windows	<code>&lt;OBT_HOME&gt;\config</code>

Example:

```
<OBT_HOME>\config
```

2. Open the `log4j2.properties` file by using a text editor and edit appropriately.
3. Save the `log4j2.properties` file.

The changes are applied automatically.

## Install Discovery

To install Discovery and Discovery service, see *Installation* section in *Structured Data Manager Discovery Guide*.

## Moving the Structured Data Manager installation to a different server

To move the Structured Data Manager installation to a different server

### On Windows server

1. Stop all SDM processes, including Web Console, Designer and AQS.
2. Map the network drive of new server( for e.g, Z:/ ) with the existing *<install\_directory>* of SDM (for e.g C:/)

**TIP:** Refer <https://kb.netgear.com/19864/How-do-I-map-a-network-drive-in-Windows>

3. Copy current SDM installation to a new server.

For example:

- Installation folder = C:\SDM\SDM766
- OBTHOME folder = C:\SDM\OBTHOME

4. Open Command Prompt.

**NOTE:**

- Maintain the SDM installation folder name as original
- Use *<robocopy>* command to copy the SDM installation from current server to the new server

5. Run the command `C:\>robocopy C:\SDM Z:\SDM /e`

**NOTE:** If the installation and OBTHOME folders are in two different location, run the robocopy command twice.

6. Restart Server Structured Data Manager processes in the new SDM Server, such as Web Console and AQS and Discovery.
7. Launch the Web Console of the new SDM Server
8. Edit **AQS\_CACHE** location under **Active Environment > Locations** to verify/update the IP Address of the initial SDM Server.
9. Continue to use SDM including Designer and Discovery.

### On Linux server

1. Stop all Structured Data Manager processes, including Web Console, Designer and AQS.
2. Copy current SDM Installation to new server by mounting current SDM installation filesystem via NFS to a mount point in the new SDM server where it has to be copied.

**NOTE:**

- Current SDM file system `/home/user/SDM/` needs to be mounted in the new SDM server, with a mount point `/mnt`.
  - Here, mount point `/mnt` is `/home/user` in the new SDM Server.

Example commands Mount current SDM server (15.114.243.123)

- `15.114.244.135:/home/user/ /mnt;`
    - Here, `15.114.244.135:/home/user/` is `/mnt`.
  - `cd /home/user/sdm766b197/.`
  - `rsync -avzh /home/tester/sdm766b197 /mnt`
3. Restart Server Structured Data Manager processes in the new SDM Server, such as Web Console and AQS and Discovery.
  4. Launch the Web Console of the new SDM Server
  5. Edit **AQS\_CACHE** location under **Active Environment > Locations** to verify/update the IP Address of the initial SDM Server.
  6. Continue to use SDM successfully including Designer and Discovery.

# Chapter 4: Remove Structured Data Manager

This section includes:

- [Reinstall the repository , below](#)

Database objects installed with Structured Data Manager, and files and folders created after the installation, are not removed by the uninstall software.

**CAUTION:** Do not remove any files or directories manually. You could inadvertently delete your archived data files.

Before uninstalling the software, ensure that you have stopped all applications (Web Console and Designer).

See *Structured Data Manager Runtime Guide*

## To uninstall Structured Data Manager in Windows

1. Select Uninstall from the Structured Data Manager program group in the Start menu.

## To uninstall the Structured Data Manager software in UNIX

1. Navigate to the directory containing the uninstallation software.

Example:

```
cd <install_directory>/uninstaller
```

where *<install\_directory>* is the location where you installed the software.

2. Enter the following command at the prompt.

```
sh ./uninstall -i console
```

## Reinstall the repository

If you plan to reinstall the repository, you need to perform the following tasks.

1. Delete the environments you created.

Related Information: *Micro Focus Structured Data Manager Runtime Guide*

2. For Oracle, remove the repository user that was created during the installation process.

The default value for the repository user is `obt_rep`, and the user is located where you installed the repository.

Type of Object	Default Value
----------------	---------------

Databases	obt_rep
Logins	obt_rep
Master Database User	obt_rep

3. For PostgreSQL, the previous repository schema must be dropped from the database before reinstalling.
4. Make a backup copy of the `connection-sources.xml` and `hpdbackiving.auth` files from the following directory:

Mode	See
UNIX	/home/username/OBTHOME/config
Windows	C:\SDM\OBTHOME\config

5. Delete the `connection-sources.xml` and `hpdbackiving.auth` files.

**NOTE:** Do not remove the template file, `connection-sources.xml.sample`.

6. Use the Web Console or scripted deployment to install the repository, create your environments, and deploy your business flows.

Related Information: *Structured Data Manager Runtime Guide*

**NOTE:** If you decide to uninstall the entire product, see [Remove Structured Data Manager](#), on the previous page.

# Appendix

This section includes:

- [Copy additional jar files, below](#)

## Copy additional jar files

During repository import, to automatically copy any additional jar files (for example, third-party-classes-1.0.jar) from SDM 7.65 to SDM later version (for example SDM 7.66). The new property **import.additional.files** is available in the **outerbay.properties** file.

By default all the jars in JDBC directory from previous installation (for example SDM 7.65) are copied to the current installation (for example SDM 7.66).

If you want this functionality to be available in the versions prior to SDM 7.65, then manually copy the below lines and make appropriate changes.

```
## This property is used to import any additional files added by the user in
# the existing SDM installation to the new SDM installation.
# The JDBC drivers get copied during SDM version upgrade but for other files
# you need to specify in this property.
# The JDBC drivers are copied to $OBTPA_TOP/lib/jdbc, $OBTPA_TOP/lib,
# $OBTPA_TOP/ui/plugins/noneclipsedependencies and
# $OBTPA_TOP/webconsole/apache-tomcat/webapps/WebConsole/WEB-INF/lib
# Usage Example:
# import.additional.files=$OBTPA_TOP/lib/third-party-classes-1.0.jar,\
# $OBTPA_TOP/ui/plugins/noneclipsedependencies/my-util-classes.jar,\
# $OBT_HOME/config/myUtilConfig.properties,\
# $OBT_HOME/extensions/my-ext-classes.jar
# Where,
# $OBTPA_TOP is <SDM Install Folder>/obt
# $OBT_HOME is SDM home directory
# A point to note here, never use absolute path for $OBTPA_TOP or $OBT_HOME
# Also, while doing the upgrade the designer MANIFEST file is updated
# appropriately.
import.additional.files=
```

# 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 Micro Focus 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 Micro Focus 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.

**IDOL**

See [Intelligent Data Operating Layer \(IDOL\)](#).

**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.

**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 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 Micro Focus Structured Data Manager 7.66 Installation Guide**

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!

