

# Structured Data Manager

Software Version 7.60

## Installation Guide



Document Release Date: August 2018  
Software Release Date: August 2018

## Legal notices

### Copyright notice

© Copyright 2017-2018 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 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.

### Trademark notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## 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 verify you are using the most recent edition of a document, go to

<https://softwaresupport.softwaregrp.com/group/softwaresupport/search-result?doctype=online help>.

If you subscribe to the appropriate product support service, you will receive new or updated editions of documentation. Contact your Micro Focus sales representative for details.

To check for new versions of software, go to the [Micro Focus Software Licenses and Downloads Portal](#). Alternatively, to check for recent software patches, go to <https://softwaresupport.softwaregrp.com/patches>.

All sites listed in this section require you to sign in with a Software Passport. You can register for a Passport through a link on the site.

## Support

Visit the [Micro Focus Software Support Online website](#) to access contact information and details about the products, services, and support that Micro Focus offers.

This website 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 support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Access the Software Licenses and Downloads portal
- Download software patches
- Access product documentation
- Manage support contracts
- Look up Micro Focus support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most support areas require you to register as a Passport user and sign in. Many also require a support contract. You can register for a Software Passport through a link on the Software Support Online website. For

more information about the different access levels the website uses, go to <https://softwaresupport.softwaregrp.com/web/softwaresupport/access-levels>.

# Contents

About this document .....	6
Prerequisites .....	6
Related documentation .....	6
 Chapter 1: Prerequisites .....	 9
Obtain Structured Data Manager documentation .....	9
Structured Data Manager installation overview .....	9
 Chapter 2: Requirements .....	 11
Operating system and server requirements .....	11
Supported operating systems .....	11
Server requirements .....	12
Database to file supported storage locations .....	12
Storage adapters for database to file archiving .....	13
Supported browsers .....	13
Database requirements .....	13
DB2 database requirements .....	14
Oracle database requirements .....	14
Source database preparation for Oracle .....	15
Target database preparation for a distributed instance with Oracle .....	16
Database preparation for a single instance with Oracle .....	17
Microsoft SQL Server requirements .....	17
Distributed archive requirements .....	18
Sybase requirements .....	18
JDBC URL requirements .....	19
Repository preparation .....	19
Repository preparation for Oracle .....	19
SQL Server repository requirements .....	20
Repository preparation for PostgreSQL .....	20
Embedded repository requirements and considerations .....	21
Start the embedded repository .....	21
Stop the embedded repository .....	21
View the embedded repository configuration information .....	22
Access the embedded repository using SQL .....	22
Back up and restore your embedded repository .....	24
Repository preparation for PostgreSQL .....	24
Optional feature requirements .....	25
Custom target schema or database .....	25

- Create the target schema or database .....26
- Create target tables .....26
  
- Chapter 3: Installation .....28**
  - Install Structured Data Manager .....28
  - Locate Structured Data Manager files .....29
  - Install the software using scripted installation .....30
  - Manage OBT\_Home .....30
  - Use the log files .....31
    - View the installation software log file .....31
    - Edit the logging properties .....32
  - Install Postgres 9.3 database on Linux .....32
  
- Chapter 4: Remove Structured Data Manager .....34**
  - Reinstall the repository .....34
  
- Glossary .....36**
  
- Send documentation feedback .....44**

# 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

- *Structured Data Manager API Reference Guide*  
Provides information about the Groovy script API files for Structured Data Manager.
- *Structured Data Manager Concepts Guide*  
Explains the major concepts of database archiving in general and Structured Data Manager in particular.
- *Structured Data Manager Installation Guide*  
Explains how to use the Installer to install the product.
- *Structured Data Manager Tutorial*  
Provides step-by-step instructions to build a sample archiving module, deploy it, run it, and troubleshoot errors.
- *Structured Data Manager Developers Guide*  
Explains how to use the Designer component to design, build, test, and deploy your archiving projects.
- *Structured Data Manager Runtime Guide*  
Explains how to use the Web Console component to run, monitor, and administer business flows that move data to and from the database.

- *Structured Data Manager Troubleshooting Guide*  
Explains how to diagnose and resolve errors, and provides a list of common errors and solutions.
- *Structured Data Manager Upgrade Guide*  
Explains how to upgrade the product and upgrade the archive schema generated by earlier versions of the product.
- *Structured Data Manager Release Notes*  
Lists any items of importance that were not captured in the regular documentation.
- *Structured Data Manager PeopleSoft Modules Installation and Deployment Guide*  
Explains how to install the PeopleSoft integration kit.
- *Structured Data Manager Oracle E-Business Suite Modules Installation and Deployment Guide*  
Explains how to install the Oracle E-Business Suite integration kit.



# 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 19](#)

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 11](#)
- [Database requirements, on page 13](#)

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 28](#)

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 34](#).

# 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 9](#).

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, below](#)
- [Server requirements, on the next page](#)
- [Database to file supported storage locations , on the next page](#)
- [Supported browsers, on page 13](#)

## Supported operating systems

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

The following platforms are supported:

- Micro Focus-UX on Itanium

**NOTE:** Micro Focus recommends installing the most recent Structured Data Manager-UX Quality Pack and any necessary patches.

- Solaris
- IBM AIX
- Microsoft Windows
- Microsoft Windows Server

- Oracle Linux
- Red Hat ES Linux

**NOTE:** On Red Hat version 4 (or higher) or SUSE LINUX Enterprise Server 9 (or higher) platforms, the default GNU C library (GLIBC) version 2.3 must be upgraded to version 2.4 or above before installing Structured Data Manager 7.3.

## Server requirements

For all servers required for Structured Data Manager:

- Identify a directory with at least 800 MB of disk space to install the Structured Data Manager.

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

- 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 such as ACA, 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:

- Digital Safe
- Consolidated Archive On-Premise
- Amazon Simple Storage Service
- Filesystem
- HPE Cloud Object Storage
- SSH (Secure Shell)
- XAM (eXtensible Access Method)
- Vertica
- Content Manager

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

## Storage adapters for database to file archiving

- Digital Safe
- EMC Centera
- Content Manager

## Supported browsers

The Structured Data Manager Web Console has been tested against the following browsers:

- Internet Explorer 11.0
- Mozilla Firefox 3.6
- Google Chrome

## 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 17
- Sybase, see [Sybase requirements](#) , on page 18
- JDBC URL with compatible driver, see [JDBC URL requirements](#) , on page 19

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><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 SQL Server database or in an embedded Java database installed with the Structured Data Manager software.

See also:

- [Repository preparation, on page 19](#)
- [Embedded repository requirements and considerations, on page 21](#)

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

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

The source and target databases require the following DB2 database versions:

- DB2 9.7
  - 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. One of the following Oracle database versions is required:

- Oracle 12c
- Oracle 11g
- Oracle 10g Release 2
- Oracle 9i Release 2—target database must be equal or same as source database
- Oracle 8i Release 2—can be used as a database to file source database only

#### Related information

For specific version numbers, see the *Structured Data Manager Certification Matrix*.

The repository databases require the following Oracle database version:

- Oracle 12c
- Oracle Database 11g Release 2 (11.2.0.3 or higher)

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 specific version numbers, refer to the *Structured Data Manager Certification Matrix*.

The source and target databases require the following SQL Server database versions:

- SQL Server Enterprise Edition 2005 with Service packs 1 and 2
- SQL Server Enterprise Edition 2008
- SQL Server Enterprise Edition 2008 R2
- SQL Server Enterprise Edition 2012
- SQL Server Enterprise Edition 2014
- SQL Server Enterprise Edition 2016

An administrator account with the following privileges is required:

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

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.

**TIP:** To use SQL Server as a repository database, the SQL Server Browser service (sqlbrowser) must be running.

## Sybase requirements

The following Sybase versions are supported for the source database:

- Sybase ASE 12.5
- Sybase ASE 15

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 SQL Server database
- in an embedded Java database installed with the software
- 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](#)
- [SQL Server repository requirements, on the next page](#)
- [Embedded repository requirements and considerations, on page 21](#)
- [Repository preparation for PostgreSQL, on page 24](#)

## Repository preparation for Oracle

The Oracle 11g 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.

## SQL Server repository requirements

The repository database requires the following SQL Server database versions:

- SQL Server Enterprise Edition 2008
- SQL Server Enterprise Edition 2012
- SQL Server Enterprise Edition 2014
- SQL Server Enterprise Edition 2016

If you are using a SQL Server as a repository database, ensure the SQL Server Browser service is running and available.

## Repository preparation for PostgreSQL

You can use PostgreSQL for Windows and Linux platforms as a repository database. PostgreSQL version 9.5 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.

The repository database requires PostgreSQL version 9.5.

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

## Embedded repository requirements and considerations

The Java DB10.3 database used for the embedded repository is installed with Structured Data Manager software. Java DB is an open source database that uses ANSI-standard SQL. If you choose to use an embedded repository to store your metadata, you should be aware of the following considerations:

- The embedded repository reduces the amount of space required on your production servers.
- The embedded repository must be running before you create the repository user.
- The embedded repository runs slower than a repository installed on your production database.
- The embedded repository must be manually started and stopped before and after using Structured Data Manager.
- The Java DB database must be maintained, and requires different administrative skills for backup and recovery than your source database.

## Start the embedded repository

Use the `launch_repository` script to start the embedded repository.

1. Navigate to the directory containing the startup script.

Example:

```
cd <install_directory>/obt/bin
```

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

2. Start the embedded repository using the appropriate syntax:

For	Syntax
UNIX	<code>./launch_repository.sh &lt;port&gt;</code>
DOS	<code>launch_repository.bat &lt;port&gt;</code>

where *<port>* is the is the port number you are using for the repository. The port number is not required if you are using the default value of 1527.

## Stop the embedded repository

Use the `shutdown_repository` script to stop the embedded repository.

1. Navigate to the directory containing the shutdown script.

Example: `cd <install_directory>/obt/bin/`

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

Stop the embedded repository using the appropriate syntax:

For	Syntax
UNIX	<code>./shutdown_repository.sh</code>
DOS	<code>shutdown_repository.bat</code>

## View the embedded repository configuration information

After you finish deploying database to database or database to file using the embedded repository, you can view the configuration information.

1. Navigate to the directory that contains the embedded repository information

```
cd <install_directory>/obt/dbrep/
```

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

2. Open the `access.cfg` file with a text editor to view the login information.

The `access.cfg` file is created after you select Embedded Repository as your repository location. The password is encoded using one-way encryption.

Example:

```
#Repository Account
#Wed Nov 12 13:27:06 PST 2008
user=obt_rep
value=soYkPfWJ20VeePpcCZ3mt0iw4jHTj0i5itdw9amuShlFg\=
```

3. Open the `server.properties` file with a text file to view the stored connection information.

```
#storing server properties
#Wed Nov 12 13:23:23 PST 2008
port=1527
host=myhost.myhost.net
dbname=obt_rep
```

## Access the embedded repository using SQL

You can access the embedded repository using your favorite JDBC tool or the Interactive JDBC client (ij) that is installed with Structured Data Manager.

To use your favorite JDBC tool, use the following connection information:

Connection Information	Location
JDBC driver	<install_directory>/obt/lib/derbyclient.jar
JDBC URL	jdbc:derby://<host>:<port>/obt_rep;

Where	Is
<install_directory>	the location where you installed the Structured Data Manager software.
<host>	the name of the host machine or IP address on which the embedded repository is running.
<port>	the port number for the host machine. The default value is 1527.

To connect using ij:

1. Start Interactive JDBC (ij).

For	Use
UNIX	/<install_directory>/obt/bin/dbarch_sql.sh
Windows	<install_directory>/obt/bin/dbarch_sql.bat

where <install\_directory> is the location where the Structured Data Manager software was installed.

2. Connect to the driver using the following syntax:

```
driver 'org.apache.derby.jdbc.ClientDriver';
```

3. Connect to the server using the following syntax:

```
connect 'jdbc:derby://<host>:<port>  
/obt_rep;user=<user_name>;password=<password>';
```

Where	Is
host	the name of the host machine or IP address on which the embedded repository is running.
port	the port number for the host machine. The default value is 1527.
username	the name of the repository user.
password	the password for the repository user.

4. After connecting, use SQL commands to query the database.

Command	Description
help;	displays a list of all supported ij commands.
show tables;	displays a list of all tables and schemas.
select tablename, tabletype from sys.systables;	displays a list of all tables and views, and labels each one appropriately.
describe <table_ name>;	displays the table columns. For tables not owned by the obt_rep user, enter the fully qualified table name.

## Back up and restore your embedded repository

### To back up your repository

1. Make sure that the repository is stopped. See [Stop the embedded repository , on page 21](#)
2. Copy the following directory to a backup location:

```
<install_directory>/obt/dbrep/
```

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

### To restore your repository from a copy

1. Make sure that the repository is stopped. See [Stop the embedded repository , on page 21](#)
2. Copy the backup directory you made to the following location:

```
<install_directory>/obt/dbrep
```

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

For information on performing an online backup, see the Apache Derby documentation at <http://db.apache.org/derby/manuals/index.html>.

## Repository preparation for PostgreSQL

You can use PostgreSQL for Windows and Linux platforms as a repository database. PostgreSQL version 9.5 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.

The repository database requires PostgreSQL version 9.5.

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, on the next page](#)
- [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](#)
- [Locate Structured Data Manager files](#)
- [Use the log files](#)

## 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, it is necessary to turn off the User Account Control (UAC) before installation.

2. The OBT\_HOME environment variable is used during installation to define where the Installer should set up the home directory for Structured Data Manager. The home directory is where Structured Data Manager stores application data, such as configuration files and log files. By default, OBT\_HOME is set to the location where your operating system stores application data, for example, on Windows, C:\SDM\OBTHOME.

If you prefer to specify another location for Structured Data Manager application data, you must change OBT\_HOME 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. 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.

6. Select the location to create product icons. Click **Next**.
7. Review the Pre-Installation Summary. Click **Install**.

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

8. 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.

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

9. The Launch Options page opens.
  - a. Select the check box to start the Web Console server using the default port 8080, 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.
10. Click **Next**.
11. Click **Done** to close the installation software.

**NOTE:** On Windows 7 SP1 or 8, you need to run Designer as administrator on first launch. After installation, right-click the Designer icon shortcut and select **Run as administrator**.

12. 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

## Locate Structured Data Manager files

Structured Data Manager configuration, log, and archive files are located in the application data directory for your operating system. By default, OBT\_HOME is set to the location where your operating system stores application data, for example, on Windows, C:\SDM\OBTHOME.

**NOTE:** C:\ProgramData is a hidden folder by default on Windows.

To see the ProgramData folder, go to **Control Panel > Folder Options > View** and select **Show hidden files, folders, and drives**.

## 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` file.

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

The `install.properties` file requires the following two lines:

```
INSTALLER_UI=<mode>
```

```
USER_INSTALL_DIR=<location>
```

Where	Is
<mode>	The type of installation. For a scripted installation, the mode should be SILENT.
<location>	The location where you want to install the software.

### Example

```
INSTALLER_UI=SILENT
```

```
USER_INSTALL_DIR=C:/SDM
```

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 `OBT_HOME` environment variable is used during installation to define where the Installer should set up the home directory for Structured Data Manager. 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	<sdm_home>/log
Windows	<sdm_home>\log

1. Navigate to the directory containing the log file.

```
<app_data_dir>/SDM/OBTHOME/config
```

where *<app\_data\_dir>* is the location where your operating system stores application data, or your own custom location.

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

File Name	Description
*_InstallLog.log	The *_InstallLog.log file is generated by the installation software, and includes information on the overall installation.
obt.log	<p>The obt.log file captures all logging information for Structured Data Manager.</p> <p>By default, the UNIX log is stored under obt.log is under /home/username/OBTHOME/log By default, the Windows log is stored under obt.log is under C:\SDM\OBTHOME\log.</p> <p>To change the types of information captured in the log, see <a href="#">Edit the logging properties, on the next page</a>. Multiple log files are numbered in sequence. For example, obt.log1.</p>

## Edit the logging properties

After you have installed the product, you can edit the `log4j.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 10 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 100 MB maximum size.

1. Navigate to the directory that contains the `log4j.properties` file.

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

Example:

```
C:\SDM\OBTHOME\config
```

where `<app_data_dir>` is the location where your operating system stores application data, or your own custom location.

2. Open the `log4j.properties` file by using a text editor and edit appropriately.
3. Save the `log4j.properties` file.

The changes are applied automatically.

## Install Postgres 9.3 database on Linux

To install PostgreSQL 9.3 database along with SDM 7.20 installation on Linux platforms, perform the following steps:

1. Log in as root.
2. Start the `vncserver`.
3. Open the `vncview` on the Windows desktop.
4. Open a terminal and navigate to the following directory:

```
<installSDMDir>/postgresql/
```

5. Run the following to install PostgreSQL 9.5:

```
./postgresql-linux.run
```

For more details, see ADM-6048.

# 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.

3. For SQL Server, remove the repository database, login, and master database user from the source databases.

Type of Object	Default Value
Databases	obt_rep
Logins	obt_rep
Master Database User	obt_rep

4. For PostgreSQL, the previous repository schema must be dropped from the database before reinstalling.
5. 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

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

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

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

# 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 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 Installation Guide (Micro Focus Structured Data Manager 7.60)**

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!

