



Red Hat JBoss Data Virtualization 6.3 Installation Guide

This guide is for installation teams.

Red Hat Customer Content
Services

Red Hat JBoss Data Virtualization 6.3 Installation Guide

This guide is for installation teams.

Red Hat Customer Content Services

Legal Notice

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document will guide the user through all of the installation options for Red Hat JBoss Data Virtualization.

Table of Contents

Chapter 1. Before You Begin	3
1.1. Back Up Your Data	3
Chapter 2. Platform requirements	4
2.1. Evaluating your architecture and your needs	4
Chapter 3. Download the Product	7
3.1. How to Download JBoss Data Virtualization Installer	7
3.2. Verify Downloaded Files	7
Chapter 4. Installing Prerequisite Components	9
4.1. Install Open JDK on Red Hat Enterprise Linux	9
4.2. Install Maven	9
Chapter 5. Installing Red Hat JBoss Data Virtualization	12
5.1. Installing JBoss Data Virtualization: Graphical Installation	12
5.2. Installing JBoss Data Virtualization Using Text Based Installer	13
5.3. Automated Installation	14
Chapter 6. Installing the JBoss Data Virtualization Development Tools	15
6.1. Installing JBoss Data Virtualization Development Tools	15
Chapter 7. Running Red Hat JBoss Data Virtualization	16
7.1. Starting JBoss Data Virtualization	16
7.2. Start JBoss EAP 6 as a Managed Domain	16
7.3. Stopping JBoss Data Virtualization	17
Chapter 8. Configuring Your Maven Repositories	18
8.1. About The Provided Maven Repositories	18
8.2. Configure Maven to Use the File System Repositories	18
8.3. Configure Maven to Use the Online Repositories	21
8.4. Using Maven Dependencies for Red Hat JBoss Data Virtualization	23
8.5. Offline mode	25
Chapter 9. Integrating Red Hat JBoss Data Virtualization with Red Hat JBoss Data Grid	26
9.1. Installing in Domain Mode	26
Chapter 10. Installing JBoss Data Grid Caching	28
10.1. Configure Red Hat JBoss Data Grid Adaptors	28
Chapter 11. ODBC Support	29
11.1. Install the ODBC Driver on Red Hat Enterprise Linux	29
11.2. Configure the ODBC Environment	29
11.3. Configure the DSN for Linux Installation	29
11.4. Install the ODBC Driver on Microsoft Windows	29
11.5. Configure the DSN for Windows Installation	30
11.6. Install the ODBC Driver on Solaris	30
11.7. Configure the DSN for Solaris Installation	31
11.8. Configure ODBC Options on Red Hat Enterprise Linux	31
11.9. Configure ODBC Options on Microsoft Windows	32
11.10. DSN-less Connection	33
Chapter 12. Running in Cloud Environments	35
12.1. Run Red Hat JBoss Data Virtualization in an Amazon AWS Cloud Instance	35
12.2. Run Red Hat JBoss Data Virtualization in a Google Compute Instance	35

12.3. Connecting Red Hat JBoss Data Virtualization to an Azure Instance	35
Chapter 13. Running Red Hat JBoss Data Virtualization Cartridge on OpenShift Online v2	38
13.1. Cartridge Installation	38
13.2. Use the Data Virtualization Web Interface on OpenShift Online v2	39
Appendix A. Configuration Information	43
A.1. Recommended Translators for Data Sources	43
Appendix B. Revision History	45

Chapter 1. Before You Begin

1.1. Back Up Your Data



Warning

Red Hat recommends that you back up your system settings and data before undertaking any of the configuration tasks mentioned in this book.

Chapter 2. Platform requirements

2.1. Evaluating your architecture and your needs

Minimum sizing recommendations

The following minimum requirements should be thought of as a starting point. These should be adjusted based on expected usage.

JBDS (Teiid Designer) – without application server

- 2 GB RAM will get you started, but more is needed for large models
- Modern Processor
- 500 MB disk space for installed product files
- 2+GB for model projects and related artifacts

The goal of the following sizing recommendations will be to provide a starting point (minimum size) for server.

The following is a minimum recommendation, a starting point. It should also be used when no client information can be obtained in order to make sizing recommendations.

The minimum sizing for the DV server is:

- 16 GB JVM memory size
- Modern multi-core (dual or better) processor or multi-socket system with modern multi-core processors
- 20+ GB Disk Space that's needed for JBoss server product and DV components:
- 1 GB disk for installed product files
- 5+ GB for log files and deployed artifacts
- 50 GB (default) for BufferManager maxBufferSpace
- If Modeshape (repository) will be used, will need to bump up the file space by a minimum 5GB more.
-

There are three considerations that will be used to help determine the minimal JVM footprint; concurrency, data volume and plan processing.

- Concurrency – this takes into account max sessions, the transport thread pool, the engine thread pool / engine (especially max active) settings and connection pool sizes.
- Data Volume – this considers the amount of data read from the data source(s) based on the batch sizes. The default processor batch size is 256 with a target of ~2k bytes per row, flowing thru the system at the size of ~512kb. However, it is recommended on machines with more memory to increase the batch size to 512, making it ~1mb per batch in size.
- Plan Processing – this considers the additional processing on the data that will be done based on the query plan. This will generally require additional memory (i.e., sorting).

The following are the assumptions that will be used in determining size:

- the server is tuned (i.e., thread pools, connection pools, etc) so that each query will execute without waiting or maximum through-put.

- ✦ there will be 1 source query per datasource in the plan (more complex queries will increase the need for more memory)
- ✦ no other apps are running in the same JVM as Teiid (if other apps will be running in the same JVM, then the additional memory requirements will need to be accounted for)
- ✦ executing straight reads, non-transactional (Teiid does a proactive batch fetch which increases memory requirement, this is why batch types is doubled)
- ✦ The default processor batch size is configured at 512 (changed from default of 256), which is recommended on machines with more memory to reduce batch overhead.

This is the formula to estimate minimum JVM size: **(concurrent queries) * (4 * batch bytes) + (2 * (#source queries per plan * approximate source bytes)) + overhead** where:

- ✦ concurrent queries
- ✦ batch bytes - represent the batches flowing thru the system, and using the default of 256 batch size, the byte size of that is ~512kb. However, using the recommended batch size of 512 would mean ~1mb per batch. The doubling of the batch bytes is to account for the effect of storing a batch on the work item in case a partial batch is retrieved while another batch is in process.
- ✦ 2 * batch bytes
- ✦ source queries per plan – number of data sources in the query, but limited based on the assumptions.
- ✦ the on heap size which will be 4 * batch bytes
- ✦ overhead – this would include the adjustment for AS (~300mb), additional Teiid overhead (caching, plans, etc.), connection pool overhead. But only ~300mb will be used, because the others are harder to figure, but know your server will need to take this into account for better performance.

The refined formula that will be used is:

- ✦ (concurrent queries) * (4 * batched bytes) + (2 * source bytes) * #source queries + 300mb
- ✦ (concurrent queries) * (4 * 1mb) + (2 * 512kb) * #source queries + 300mb
- ✦ (concurrent queries) * (4mb) + (1mb) * #source queries + 300mb
- ✦ #concurrency * (5mb) * #source queries + 300mb

Table 2.1. Configuration

concurrency # source queries	100	200
2	1.3 gb	2.3 gb
5	2.8 gb	5.3 gb
10	5.3 gb	10.3 gb

Based on the max concurrent queries, start with the following to tune the system's engine:

- ✦ set maxActivePlans to the max concurrent queries
- ✦ set maxThreads = maxActivePlans * 2 (if transactions will be used, then * 3)
- ✦ set each datasource max pool size = max concurrent source queries (minimum would be max concurrent queries, but if majority of queries are complex in which there are subqueries that cause multiple source queries to be spawned, then max pool size should be increased accordingly),
- ✦ After the above adjustments are done and the server has memory room, then consider increasing the

`processBatchSize` and `connectorBatchSize` (i.e., 512, 1024, respectfully) to increase through-put from the data source and thru the engine. If you're out of memory, then increase the JVM size. A machines that has less than 6GB memory, stick with 512, larger machines use higher sizes.

Chapter 3. Download the Product

3.1. How to Download JBoss Data Virtualization Installer

The JBoss Data Virtualization installer archive is available on the Red Hat Customer portal at <https://access.redhat.com/>.

Prerequisites

- Set up an account on the Red Hat Customer Portal at <https://access.redhat.com/>.
- Ensure your Red Hat subscriptions are up to date.
- Review the supported configurations and ensure your system is supportable.
- Ensure that you have administration privileges for the installation directory.
- Ensure that JAVA_HOME and PATH have been set in the Environment properties for shortcuts to work on Microsoft Windows servers.
- A Java 6, 7 or 8 JDK is required. (Please note that if you intend to use Red Hat SSO, you must have at least Java 7. Java 7 is also a minimum requirement for the Impala, Hive and HBase data sources.)
- Optional: Red Hat JBoss Enterprise Application Platform 6.4.x if you do not want to use the version of Red Hat JBoss EAP that comes bundled with the Data Virtualization installer.

Procedure 3.1. Download JBoss Data Virtualization Installer

1. **Login to the Red Hat Customer Portal.**
 - a. Go to <https://access.redhat.com/>.
 - b. Click **Log in** and enter your **Red Hat Login** and **Password** to access the **Customer Portal**. You will need to register for an account if you do not yet have one.
2. **Download JBoss Data Virtualization Installer.**
 - a. Click **Downloads** -> **Red Hat JBoss Data Virtualization**.
 - b. Click **Download** next to the **Red Hat JBoss Data Virtualization [Version] Installer** option.
 - c. Save the file.

3.2. Verify Downloaded Files

Procedure 3.2. Verify File Checksums on Red Hat Enterprise Linux

1. **Obtain checksum values for the downloaded file**
 - a. Go to <https://access.redhat.com/jbossnetwork/>. Log in if required.
 - b. Select your **Product** and **Version**.
 - c. Select the package you want to verify. Once you have chosen them, navigate to the **Software Details** page.

- d. Take note of the **MD5** and **SHA-256** checksum values.

2. Run a checksum tool on the file

- a. Navigate to the directory containing the downloaded file in a terminal window.
- b. Run `md5 downloaded_file`.
- c. Run `shasum downloaded_file`.

Example output:

```
[localhost]$ md5 jboss-dv-installer-[VERSION]-redhat-[VERSION].jar
MD5 (jboss-dv-installer-[VERSION]-redhat-[VERSION].jar) =
0d1e72a6b038d8bd27ed22b196e5887f
[localhost]$ shasum jboss-dv-installer-[VERSION]-redhat-[VERSION].jar
a74841391bd243d2ca29f31cd9f190f3f1bdc02d  jboss-dv-installer-
[VERSION]-redhat-[VERSION].jar
```

3. Compare the checksum values

- a. Compare the checksum values returned by the `md5` and `shasum` commands with the corresponding values displayed on the **Software Details** page.
- b. Download the file again if the two checksum values are not identical. A difference between the checksum values indicates that the file has either been corrupted during download or has been modified since it was uploaded to the server. Contact Red Hat Support for assistance if after several downloads the checksum does not successfully validate.
- c. The downloaded file is safe to use if the two checksum values are identical.



Note

No checksum tool is included with Microsoft Windows. Download a third-party MD5 application such as MD5summer from <http://www.md5summer.org/>.

Chapter 4. Installing Prerequisite Components

4.1. Install Open JDK on Red Hat Enterprise Linux

Procedure 4.1. Install Open JDK on Red Hat Enterprise Linux

1. Subscribe to the Base Channel

Obtain the OpenJDK from the RHN base channel. (Your installation of Red Hat Enterprise Linux is subscribed to this channel by default.)

2. Install the Package

Use the yum utility to install OpenJDK: `yum install java-1.7.0-openjdk-devel`

3. Verify that OpenJDK is Now Your System Default

You can ensure the correct JDK is set as the system default by following the steps below.

- a. As root, run the alternatives command for `java`:

```
/usr/sbin/alternatives --config java
```

- b. Select `/usr/lib/jvm/jre-1.7.0-openjdk/bin/java`.

- c. Then do the same for `javac`:

```
/usr/sbin/alternatives --config javac
```

- d. Select `/usr/lib/jvm/java-1.7.0-openjdk/bin/javac`.

4.2. Install Maven

Prerequisites

The following software must be installed:

- ✦ An archiving tool for extracting the contents of compressed files.
- ✦ Open JDK.

Procedure 4.2. Install Maven

1. Download Maven.

- a. Enter <http://maven.apache.org/download.cgi> in the address bar of a browser.
- b. Download `apache-maven-[latest-version]` ZIP file and save it to your hard drive.

2. Install and configure Maven.

A. On Red Hat Enterprise Linux

- a. Extract the ZIP archive to the directory where you wish to install Maven.

b. Open your `.bash_profile` file in a terminal: **vi ~/.bash_profile**.

c. Add the `M2_HOME` environment variable to the file:

```
export M2_HOME=/path/to/your/maven
```

d. Also add the `M2` environment variable to the file:

```
export M2=$M2_HOME/bin
```

e. Add the `M2` environment variable to the file:

```
export PATH=$M2:$PATH
```

f. Make sure that `JAVA_HOME` is set to the location of your JDK. For example:

```
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk.x86_64
```

g. Make sure that `$JAVA_HOME/bin` is in your `PATH` environment variable.

h. Save the file and exit your text editor.

i. Run this command to ensure the changes take effect: **source ~/.bash_profile**

j. Run the following command to verify that Maven is installed successfully on your machine:

```
mvn --version
```

B. On Microsoft Windows

a. Extract the ZIP archive to the directory where you wish to install Maven. The subdirectory **apache-maven-[latest-version]** is created from the archive.

b. Press **Start+Pause|Break**. The **System Properties** dialog box is displayed.

c. Click the **Advanced** tab and click **Environment Variables**.

d. Under System Variables, select **Path**.

e. Click **Edit** and add the two Maven paths using a semicolon to separate each entry.

- Add the `M2_HOME` variable and set the path to **C:\path\to\your\Maven**.

- Add the `M2` variable and set the value to `%M2_HOME%\bin`.

f. Update or create the `Path` environment variable:

- Add the `%M2%` variable to allow Maven to be executed from the command line.

- Add the variable `%JAVA_HOME%\bin` to set the path to the correct Java installation.

g. Click **OK** to close all the dialog boxes including the **System Properties** dialog box.

h. Open Windows command prompt and run the following command to verify that Maven is installed successfully on your machine:

```
mvn --version
```

Chapter 5. Installing Red Hat JBoss Data Virtualization

5.1. Installing JBoss Data Virtualization: Graphical Installation

The Graphical Installer allows you to install **JBoss Data Virtualization** on your machine using step-by-step GUI instructions. This topic covers the steps needed to run the installer.

Prerequisites

You must have already downloaded the Red Hat JBoss Data Virtualization jar file from the [Customer Portal](#).

Procedure 5.1. Install JBoss Data Virtualization

1. Open a terminal window and navigate to the location where the GUI installer was downloaded.
2. Enter the following command to launch the GUI installer:

```
java -jar jboss-dv-VERSION-installer.jar
```

3. A dialogue box will open followed by the End User License Agreement. If you accept the terms of the agreement, click **I accept the terms of this license agreement** and then click **Next**.
4. Tell Red Hat JBoss Data Virtualization where Red Hat JBoss EAP is installed on your server or specify a new location if you do not have it installed as it comes bundled with the product. (If you have a pre-existing installation of Red Hat JBoss EAP, ensure that it is patched to the latest version of 6.4.x.) Click **Next**.
5. Ensure Teiid Installation and Modeshape Installation are ticked. Click **Next**.
6. You will be prompted to create a new EAP Admin, Dashboard Admin, Teiid data access user and ModeShape user and whether you want to enable OData access. Once created, EAP Admin is added to the ManagementRealm and can be used to access the Management Console. The other users are added to the ApplicationRealm and can be used to access specific components of JBoss Data Virtualization. Enter the new username and password in the appropriate fields and click **Next**.

You must ensure that you remember all of these passwords. They give you access to different parts of the system. The EAP account and password allows you to administer the EAP Server, the Dashboard password is for administrative functions related to the web interface, and the Teiid data access user and ModeShape user are for standard user access.

Note that the username and password are not allowed to match and the password must have at least eight characters, with one alphabetical character, one numeric character and one non-numeric character.

7. You can install Red Hat JBoss Data Virtualization either with default configuration or with additional configuration options. For this exercise, we will be using the defaults only, so select Perform default configuration to install Red Hat JBoss Data Virtualization with default options. Click **Next**.
8. The Configure password vault screen appears. Input your desired password, which must have no fewer than six characters. Click **Next**.

**Note**

The default H2 database is not suitable for production databases. Use it in testing and evaluation environments only.

9. A summary of the installation is displayed. Click **Next** for the installation to commence. This may take a few moments. Once all the components are installed, click **Next**.
10. Click **Generate an automatic installation script** if you wish to generate automatic script. This allows you to quickly reinstall or mass-deploy the product using the settings you have configured during the initial installation, without having to step through the wizard each time.

**Note**

Note that an automatic installation script created for Red Hat JBoss Data Virtualization 6.0 will not work with Red Hat JBoss Data Virtualization 6.1.

11. Click **Done** to exit the installer.

**Note**

Note that after installing JBoss Data Virtualization, if you move the product to another location, you may see some FileNotFound exceptions. This is due the fact that some filepaths are hard-coded by the JBoss EAP Server.

**Warning**

If you attempt to use a vault with a keystore created with a different JDK than the one in which the data is stored, your server will fail to start. You must consistently use the same JDK when accessing the vault.

5.2. Installing JBoss Data Virtualization Using Text Based Installer

You can install Red Hat JBoss Data Virtualization using the text-based installer. In this mode, you run the installation steps without stepping through the graphical wizard. The GUI installer will run in text mode automatically if no display server is available.

Prerequisites

You must have already downloaded the Red Hat JBoss Data Virtualization jar file from the [Customer Portal](#).

Procedure 5.2. Install JBoss Data Virtualization

1. Open a terminal window and navigate to the location where the GUI installer was downloaded.
2. Enter the following command to start the installation process:

```
java -jar jboss-dv-VERSION-installer.jar -console
```

3. Follow the installation prompts displayed on the terminal. You can either install with default configuration or you can complete additional configuration steps.
4. The final step involving generating automatic installation script. You can use this script to perform headless installation or identical installations across multiple instances.

5.3. Automated Installation

If you need to install a Red Hat JBoss product multiple times with the same configuration, you can save time by using an installation script. By using an installation script with predefined settings, you can perform the entire installation by running a single command, instead of working through the installation step by step each time. You can generate an installation script by running the installer (in graphical or text mode), stepping through with your desired configuration, and then choosing to generate the script when prompted towards the end of the process.

Prerequisites

- You must have downloaded the relevant installer JAR file from <https://access.redhat.com/jbossnetwork/>
- You must have generated the script and saved it as an XML file during a previous installation using the installer (in graphical or text mode).

Procedure 5.3. Installing with a Script

- ```
java -jar jboss-PRODUCT-installer-VERSION.jar SCRIPT.xml
```

## Chapter 6. Installing the JBoss Data Virtualization Development Tools

### 6.1. Installing JBoss Data Virtualization Development Tools

#### Prerequisites

The following software must be installed:

- ✦ Red Hat JBoss Developer Studio (See *Red Hat JBoss Developer Studio Installation Guide*)
- ✦ An archiving tool for extracting the contents of compressed files
- ✦ Open JDK (See *Red Hat JBoss Data Virtualization Installation Guide*) or another supported Java Virtual Machine

#### Procedure 6.1. Install JBoss Developer Studio Integration Stack

1. Start **Red Hat JBoss Developer Studio**.
2. In **Red Hat JBoss Developer Studio**, click **Help > Install New Software...** action from the main menu.
3. On the **Available Software** page, click the **Add ...** button.
4. On the **Add Repository** dialog, enter following details:  
  
Enter JBDSIS (or another unique name) in the **Name** field.  
  
Enter <https://devstudio.redhat.com/9.0/stable/updates/integration-stack/earlyaccess/> in the **Location** field.
5. Click **OK**.
6. In the update site tree view, select the **JBoss Data Virtualization Development** folder and all its children.



#### Note

If JBDSIS is already installed then proceed to next step. You can confirm that JBDSIS is installed on your machine by clicking **what is already installed** link.

7. Click **Next**.
8. Accept any additional dependencies and license agreements, then click **Finish** to complete.

When installation is complete you will be prompted to relaunch **Red Hat JBoss Developer Studio** to insure the new features are fully operational.

## Chapter 7. Running Red Hat JBoss Data Virtualization

### 7.1. Starting JBoss Data Virtualization

#### Procedure 7.1. Starting JBoss Data Virtualization

##### ✦ Start the JBoss EAP Server

You can run **JBoss Data Virtualization** by starting the **JBoss EAP server**. To start the **JBoss EAP server**:

##### A. Red Hat Enterprise Linux

Open a terminal and enter the command:

```
$ EAP_HOME/bin/standalone.sh
```

##### B. Microsoft Windows

Open a terminal and enter the command:

```
$ EAP_HOME\bin\standalone.bat
```



#### Note

To verify that there have been no errors, check the server log: **EAP\_HOME/MODE/log/server.log**. You can also verify this by opening the [Management Console](#) and logging in using the username and password of a registered JBoss EAP Management User. For more information about using the Management Console, see the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.



#### Note

For more advanced starting options, see the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.

### 7.2. Start JBoss EAP 6 as a Managed Domain

#### Order of Operations

The domain controller must be started before any slave servers in any server groups in the domain. Use this procedure first on the domain controller, and then on each associated host controller and each other host associated with the domain. Before you begin, please consult the Red Hat JBoss EAP Documentation at <https://access.redhat.com/documentation/en/red-hat-jboss-enterprise-application-platform/>

#### Procedure 7.2. Start the Platform Service as a Managed Domain

##### 1. For Red Hat Enterprise Linux.

Run the command: **EAP\_HOME/bin/domain.sh**

## 2. For Microsoft Windows Server.

Run the command: `EAP_HOME\bin\domain.bat`

## 3. Optional: Pass additional parameters to the start-up script.

To list all available parameters for the start-up scripts, use the `-h` parameter.

## 7.3. Stopping JBoss Data Virtualization

To stop **JBoss Data Virtualization**, you must stop the **JBoss EAP server**. The way you stop JBoss EAP depends on how it was started. You can stop JBoss EAP by pressing **CTRL+C** in the terminal.



### Note

To stop the JBoss EAP server using alternative methods see the *Red Hat JBoss Enterprise Application Platform Administration and Configuration Guide*.

## Chapter 8. Configuring Your Maven Repositories

### 8.1. About The Provided Maven Repositories

A set of repositories containing artifacts required to build applications based on is provided with this release. Maven must be configured to use these repositories and the Maven Central Repository in order to provide correct build functionality.

Two interchangeable sets of repositories ensuring the same functionality are provided. The first set is available for download and is stored in a local file system. The second set is hosted online for use as remote repositories. If you provided the location of Maven's `settings.xml` file during installation, Maven is already configured to use the online repositories.



#### Important

The set of online remote repositories is a technology preview source of components. As such, it is not in scope of patching and is supported only for use in a development environment. Using the set of online repositories in production environment is a potential source of security vulnerabilities and is therefore not a supported use case. For more information see <https://access.redhat.com/site/maven-repository>.

### 8.2. Configure Maven to Use the File System Repositories

The Red Hat JBoss DV Maven repository is available online, so it is not necessary to download and install it locally. However, if you prefer to install the JBoss EAP Maven repository locally, there are three ways to do it: on your local file system, on Apache Web Server, or with a Maven repository manager. This example covers the steps to download the JBoss EAP 6 Maven Repository to the local file system. This option can help you become familiar with using Maven for development but is not recommended for team production environments.

#### Procedure 8.1.

1. Go to <https://access.redhat.com/jbosnetwork/restricted/listSoftware.html?product=data.services.platform>
2. Download the desired version.
3. Unzip the file on the local file system into a directory of your choosing.
4. Add entries for the unzipped repositories to Maven's `settings.xml` file. The following code sample contains a profile with the repositories and an activation entry for the profile:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/xsd/settings-1.0.0.xsd">
 <localRepository/>
 <profiles>
 <!-- Profile with local repositories required by Data
Virtualization -->
 <profile>
 <id>dv-local-repos</id>
```

```

<repositories>
 <repository>
 <id>dv-[VERSION]-repository</id>
 <name>DV [VERSION] GA Repository</name>
 <url>file://<!-- path to the repository -->/jboss-dv-
[VERSION].redhat-[VERSION]-maven-repository/maven-repository</url>
 <layout>default</layout>
 <releases>
 <enabled>>true</enabled>
 <updatePolicy>never</updatePolicy>
 </releases>
 <snapshots>
 <enabled>>false</enabled>
 <updatePolicy>never</updatePolicy>
 </snapshots>
 </repository>
</repositories>
<pluginRepositories>
 <pluginRepository>
 <id>dv-[VERSION].GA-redhat-[VERSION]-repository</id>
 <name>DV [VERSION] GA Repository</name>
 <url>file://<!-- path to the repository -->/jboss-dv-
[VERSION].redhat-[VERSION]-maven-repository/maven-repository</url>
 <layout>default</layout>
 <releases>
 <enabled>>true</enabled>
 <updatePolicy>never</updatePolicy>
 </releases>
 <snapshots>
 <enabled>>false</enabled>
 <updatePolicy>never</updatePolicy>
 </snapshots>
 </pluginRepository>
</pluginRepositories>
</profile>
</profiles>
<activeProfiles>
 <!-- Activation of the Data Virtualization profile -->
 <activeProfile>dv-local-repos</activeProfile>
</activeProfiles>
</settings>

```

## Troubleshooting

**Q:** Why do I still see errors when building or deploying my applications?

**A:** **Issue**

When you build or deploy a project, it fails with one or both of the following errors:

- ✘ **[ERROR] Failed to execute goal on project *PROJECT\_NAME***
- ✘ **Could not find artifact *ARTIFACT\_NAME***

**Cause**

Your cached local Maven repository might contain outdated artifacts.

### Resolution

To resolve the issue, delete the cached local repository – the `~/ .m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows – and run `mvn clean install -U`. This will force Maven to download correct versions of required artifacts when performing the next build.

---

**Q: Why is JBoss Developer Studio using my old Maven configuration?**

**A: Issue**

You have updated your Maven configuration, but this configuration is not reflected in JBoss Developer Studio.

### Cause

If JBoss Developer Studio is running when you modify your Maven `settings.xml` file, this configuration will not be reflected in JBoss Developer Studio.

### Resolution

Refresh the Maven settings in the IDE. From the menu, choose **Window** → **Preferences**. In the **Preferences** Window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in JBoss Developer Studio.



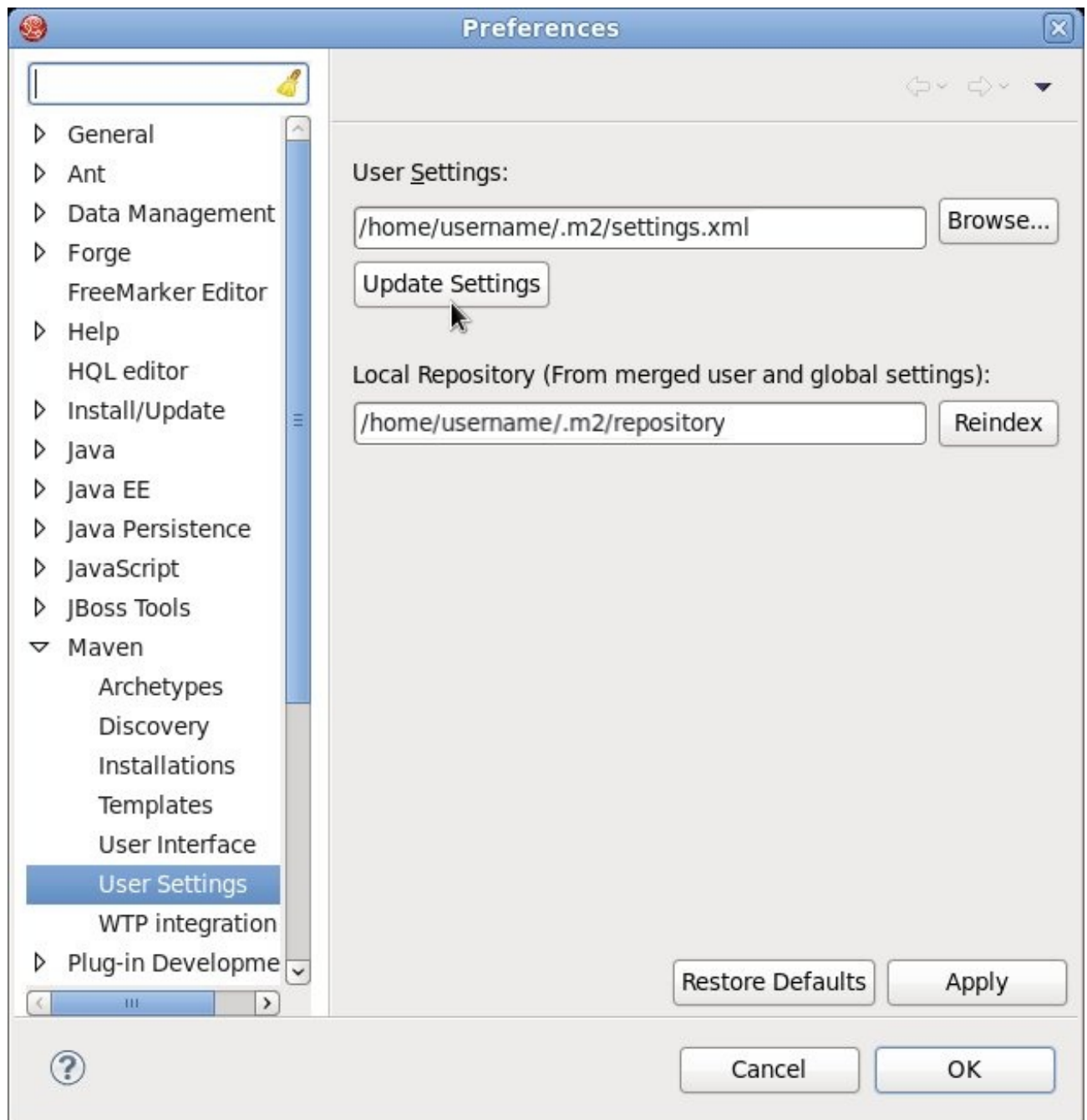


Figure 8.1. Update Maven User Settings

### 8.3. Configure Maven to Use the Online Repositories

The online repositories required for Red Hat JBoss Data Virtualization are located at <http://maven.repository.redhat.com/techpreview/all/>.

If you provided the location of Maven's `settings.xml` file during installation, Maven is already configured to use the online repositories.

#### Procedure 8.2. Configuring Maven to Use the Online Repositories

1. Add entries for the online repositories to Maven's `settings.xml` file:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
```

```

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">

 <profiles>
 <!-- Profile with online repositories required by Data
Virtualization -->
 <profile>
 <id>dv-online-profile</id>
 <repositories>
 <repository>
 <id>jboss-ga-repository</id>

 <url>http://maven.repository.redhat.com/techpreview/all</url>
 <releases>
 <enabled>>true</enabled>
 </releases>
 <snapshots>
 <enabled>>false</enabled>
 </snapshots>
 </repository>
 </repositories>
 <pluginRepositories>
 <pluginRepository>
 <id>jboss-ga-plugin-repository</id>

 <url>http://maven.repository.redhat.com/techpreview/all</url>
 <releases>
 <enabled>>true</enabled>
 </releases>
 <snapshots>
 <enabled>>false</enabled>
 </snapshots>
 </pluginRepository>

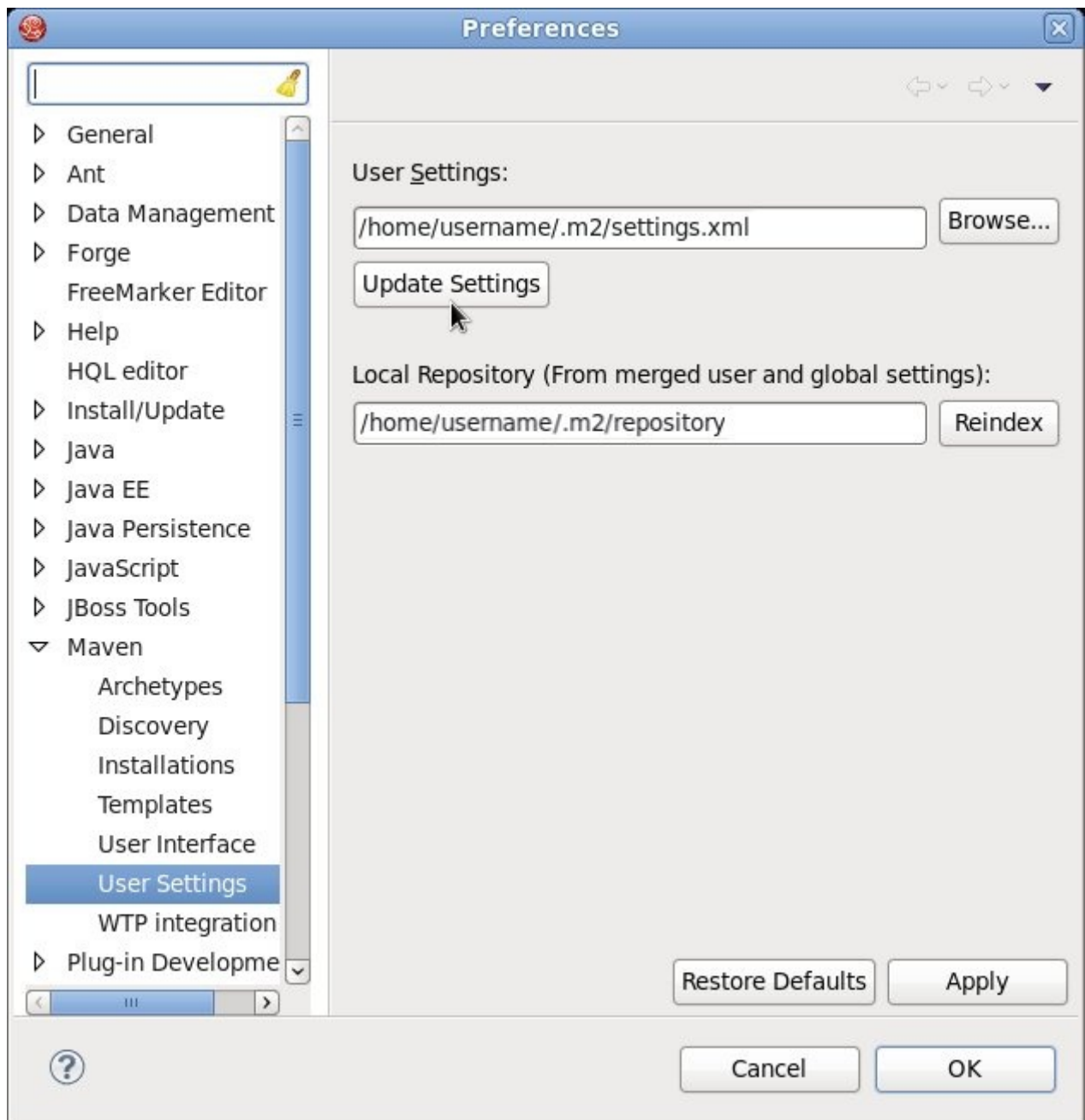
 </pluginRepositories>
 </profile>
 </profiles>

 <activeProfiles>
 <!-- Activation of the Data Virtualization profile -->
 <activeProfile>dv-online-profile</activeProfile>
 </activeProfiles>

 </settings>

```

2. If you modified the **settings.xml** file while JBoss Developer Studio was running, you must refresh Maven settings in the IDE. From the menu, choose **Window** → **Preferences**. In the **Preferences** Window, expand **Maven** and choose **User Settings**. Click the **Update Settings** button to refresh the Maven user settings in JBoss Developer Studio.



**Figure 8.2. Update Maven User Settings**

If your cached local Maven repository contains outdated artifacts, you may encounter one of the following Maven errors when you build or deploy your project:

- ✧ Missing artifact *ARTIFACT\_NAME*
- ✧ [ERROR] Failed to execute goal on project *PROJECT\_NAME*; Could not resolve dependencies for *PROJECT\_NAME*

To resolve the issue, delete the cached local repository – the `~/ .m2/repository/` directory on Linux or the `%SystemDrive%\Users\USERNAME\.m2\repository\` directory on Windows. This will force Maven to download correct versions of required artifacts during the next build.

## 8.4. Using Maven Dependencies for Red Hat JBoss Data Virtualization

In order to use the correct Maven dependencies in your Red Hat JBoss Data Virtualization project, you must add relevant Bill Of Materials (BOM) and parent POM files to the project's `pom.xml` file. Adding the BOM and

parent POM files ensures that the correct versions of plug-ins and transitive dependencies from the provided Maven repositories are included in the project.

The Maven repository is designed to be used only in combination with Maven Central and no other repositories are required.

The parent POM file to use is **org.jboss.dv.component.management:dv-parent-[VERSION].pom**

The BOM file to use is **org.jboss.dv.component.management:dv-dependency-management-all-[VERSION].pom**.

In order to use the Maven dependencies for Red Hat JBoss Data Virtualization, the relevant Bill Of Materials (BOM) and parent POM files must be added to the project POM file. By adding the BOM and parent POM files, you are ensuring that the correct versions of plug-ins and transitive dependencies are included in the project.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">

 <modelVersion>4.0.0</modelVersion>

 <!-- Example POM file using the DV 6.3.0 and EAP 6.4 component versions.
 - Parent is set to the DV 6.3.0 parent management POM, which will
 - bring in the correct toolchain (plugin) versions.
 - DependencyManagement dependencies include the DV 6.3.0 and EAP 6.4
 - BOMs - which will bring in the correct compile-time (and other
 - scoped) versions.
 -->

 <name>Example POM for DV 6.3.0</name>
 <groupId>org.jboss.dv</groupId>
 <artifactId>dv-example</artifactId>
 <version>0.0.1</version>
 <packaging>pom</packaging>

 <parent>
 <!-- DV version (parent) -->
 <groupId>org.jboss.dv.component.management</groupId>
 <artifactId>dv-parent</artifactId>
 <version>2.3.0.redhat-10</version>
 </parent>

 <dependencyManagement>
 <dependencies>
 <!-- DV BOM -->
 <dependency>
 <groupId>org.jboss.dv.component.management</groupId>
 <artifactId>dv-dependency-management-all</artifactId>
 <version>2.3.0.redhat-10</version>
 <type>pom</type>
 <scope>import</scope>
 </dependency>
 </dependencies>
 </dependencyManagement>
</project>
```

```
</dependencies>
</dependencyManagement>

</project>
```

## 8.5. Offline mode

### 8.5.1. Using a Custom Offline Repository

When you move from the development phase of a project to the *deployment* phase, it is typically more convenient to pre-install all of the artifacts required by your application, rather than downloading them from the Internet on demand. In this case, the ideal solution is to create a custom offline repository, which contains the artifacts needed for your deployment. Creating a custom offline repository by hand, however, would be difficult, because it would need to include *all* of the transitive dependencies associated with your application bundles and features.

The ideal way to create a custom offline repository is to generate it, with the help of the Apache Karaf **features-maven-plugin** plug-in.

If you have a Maven project and you need to create an offline repository for building this project and its runtime dependencies, you can use the [maven dependency plugin](#).

For example, from the top-level directory of a Maven project (such that the current directory has a **pom.xml** file), you should be able to run the following Maven command:

```
mvn org.apache.maven.plugins:maven-dependency-plugin:2.8:go-offline -
Dmaven.repo.local=/tmp/cheese
```

Which downloads all the Maven dependencies and plug-ins required to build the project to the **/tmp/cheese** directory.

To generate the custom offline repository, open a new command prompt, change directory to **ProjectDir/custom-repo**, and enter the following Maven command:

```
mvn generate-resources
```

Assuming that the Maven build completes successfully, the custom offline repository should now be available in the following location:

```
ProjectDir/custom-repo/target/features-repo
```

## Chapter 9. Integrating Red Hat JBoss Data Virtualization with Red Hat JBoss Data Grid

### 9.1. Installing in Domain Mode

When you have more than one Red Hat JBoss EAP instance in your server farm and you start them all in domain mode, all of the configuration options for this server farm can be centrally managed. For example, you can deploy a VDB or create a data source across all the instances, with one single CLI-based call. Red Hat JBoss Data Virtualization extends this configuration concept to allow you to deploy your VDBs and translators across the whole server farm.

When domain mode is combined with the "HA (high availability)" profile, you can cluster the Red Hat JBoss EAP server instances that are deployed. (The HA profile is set as the default in the "domain.xml" file.) When you start the server using the "domain.xml" file, the distributed caching that is used for ResultSet caching and Internal Materialized caching is automatically configured. The usage of Admin API is same in both standalone mode and domain mode.

When multiple Red Hat JBoss Data Virtualization instances are available in a cluster, you can make use load balancing and fail-over features.

1. To start the server in "Domain" mode, install Red Hat JBoss Data Virtualization on all the servers that are going to be part of the cluster. Select one of the servers as the "master" domain controller. (The rest of the servers will be slaves that connect to the "master" domain controller for all the administrative operations.)
2. Once you have configured all the servers, start the "master" node: **`/bin/domain.sh`**
3. Start the "slave" nodes: **`/bin/domain.sh`**

The slave nodes fetch their domain configuration settings from the "master" node.

See the section "Start JBoss EAP 6 as a Managed Domain" for more information.

4. Once all the servers are running, complete the installation to run in domain mode by executing this command on the "master" node: **`/bin/jboss-cli.sh --file=[JBOSS_HOME_DIR]/cli-scripts/teiid-domain-mode-install.cli`**

This only needs to be run once per domain (cluster) install. This script will install Red Hat JBoss Data Virtualization in the ha and full-ha profiles. It will also re-configure the main-server-group to start the HA profile. Once in domain mode, you can not statically deploy resources by dropping them in the domain/deployments directory, so this script will deploy the default resources (such as file, ldap, salesforce and ws connectors) using the CLI interface.

5. If you need to install Red Hat JBoss Data Virtualization in profiles other than HA, edit the teiid-domain-mode-install.cli file before installing it, and make the appropriate changes to the profile, socket-bindings, and server-groups.
6. Once VDBs have been deployed, users can now connect their JDBC applications to Red Hat JBoss Data Virtualization.



## Note

Teiid Designer cannot connect to the Red Hat JBoss Data Virtualization Server in "domain" mode. Red Hat recommends using other types of deployment strategies (such as CLI, web-console) for deploying and testing, as it is expected you will be using Domain mode in production environments. Teiid Designer is to aid development-time activities only and should thus only be used in testing environments.

## Chapter 10. Installing JBoss Data Grid Caching

### 10.1. Configure Red Hat JBoss Data Grid Adaptors

The `infinispan-cache` and `infinispan-cache-dsl` translators are not pre-configured to work when the server starts. You must manually configure them if you wish to use Red Hat JBoss Data Grid as a data source.

1. Navigate to the `docs/teiid/datasources/infinispan/` directory.
2. Execute the appropriate script: `add-infinispan-cache-translator.cli` or `add-infinispan-cache-dsl-translator.cli`.



## Chapter 11. ODBC Support

### 11.1. Install the ODBC Driver on Red Hat Enterprise Linux

#### Prerequisites

- ✧ Administrative permissions are required.

#### Procedure 11.1. Install the ODBC Driver on Red Hat Enterprise Linux

##### 1. Download the driver

Download the correct ODBC driver package (`jboss-dv-psqlodbc-[version]-X.rpm`) from <https://access.redhat.com/jbossnetwork/>.

##### 2. Install the package

Run `sudo yum localinstall jboss-dv-psqlodbc-[version]-X.rpm`.



#### Note

Installation packages for different operating systems can be downloaded from <https://access.redhat.com/jbossnetwork/>.

### 11.2. Configure the ODBC Environment

- ✧ Configure the Environment

Run the `/opt/redhat/jboss-dv/v6/psqlodbc/etc/setenv.sh` script:

```
[localhost etc]$./setenv.sh
```

This script adds the required directories to the `LD_LIBRARY_PATH` and `PATH` environment variables. This script has to be run every time you want to use the driver.

### 11.3. Configure the DSN for Linux Installation

- ✧ Edit the `/opt/redhat/jboss-dv/v6/psqlodbc/etc/odbc.ini` file and update it with the correct username, password, and database. The database name is the VDB name.

ODBC is enabled in JBoss Data Virtualization on port **35432** by default.

### 11.4. Install the ODBC Driver on Microsoft Windows

#### Prerequisites

- ✧ Administrative permissions are required.

#### Procedure 11.2. Install the ODBC Driver on Microsoft Windows

1. Download the correct ODBC driver package (**jboss-dv-psqlodbc-[version]-X.zip**) from <https://access.redhat.com/jbossnetwork/>.
2. Unzip the installation package.
3. Double-click the **jboss-dv-psqlodbc-[version]-X.msi** file to start the installer.
4. The installer wizard is displayed. Click **Next**.
5. The End-User License Agreement will be displayed. Click the **I accept the terms in the License Agreement** if you accept the licensing terms and then click **Next**.
6. If you want to install in a different directory other than the default directory shown, click the **Browse** button and select a directory. Click **Next**.
7. You are presented with a confirmation screen. Review the choices you have made and, when you are happy, Click **Next** to begin installation.
8. When it has finished, a screen will appear to inform you the installation process has ended. Click **Finish** to exit the wizard.



### Note

Installation packages for different operating systems can be downloaded from <http://access.redhat.com>.

## 11.5. Configure the DSN for Windows Installation

### Procedure 11.3. Configure the DSN for Windows Installation

1. Set the ODBC driver basic options.
2. Set the ODBC driver datasource options.
3. Set the ODBC driver global options.

## 11.6. Install the ODBC Driver on Solaris

### Prerequisites

- ✦ Administrative permissions are required.

### Procedure 11.4. Install the ODBC Driver on Solaris

#### 1. Download the driver

Download the correct ODBC driver package (**jboss-dv-psqlodbc-[VERSION]-X.zip**) from <https://access.redhat.com/jbossnetwork/>.

#### 2. Unzip the installation package

Unzip the installation package to **/opt** directory.

#### 3. Set the PATH Property

Set PATH property so that the ODBC binaries are used from the directory where you have unzipped the driver.

```
$ export PATH = $PATH:/opt/redhat/jboss-dv/v6/psqlodbc/bin
```

#### 4. Set environmental variable

Set ODBCINI environmental variable to existing odbc.ini.

```
$ export ODBCINI=/opt/redhat/jboss-dv/v6/psqlodbc/etc/odbc.ini
```



#### Note

If you are using the Bourne Shell as your Solaris terminal, you can add the two export commands above to your `~/.profile` file, you will not need to run them every time. Likewise, if you are using Bash, save them in your `~/.bash_profile` file instead.

## 11.7. Configure the DSN for Solaris Installation

- ✱ Edit the `/opt/redhat/jboss-dv/v6/psqlodbc/etc/odbc.ini` file and update it with the correct username, password, and database. The database name is the VDB name.

ODBC is enabled in JBoss Data Virtualization on port **35432** by default.

## 11.8. Configure ODBC Options on Red Hat Enterprise Linux

### Procedure 11.5. Configure ODBC Options on Red Hat Enterprise Linux

1. Run this command to install the driver manager: `yum install unixODBC`.
2. Run this command to verify that your PostgreSQL driver has installed correctly: `odbcinst -q -d`.
3. To create the DSN, open the configuration file in a text editor: `sudo vi /opt/redhat/odbc.ini`



#### Note

You must either use `sudo` or be logged in as root to open this file.

4. Add the following configuration settings to the file:

```
[<DSN name>] Driver = /usr/lib/psqlodbc.so
Description = PostgreSQL Data Source
Servername = <Teiid Host name or ip>
Port = 35432 Protocol = 7.4
UserName = <user-name>
Password = <password>
Database = <vdb-name>
ReadOnly = no
ServerType = Postgres
ConnSettings = UseServerSidePrepare=1
ByteaAsLongVarBinary=1
```

```
Optimizer=0
Ksqo=0 Trace =
No TraceFile = /var/log/trace.log
Debug = No DebugFile = /var/log/debug.log
```

5. Save the file and exit the text editor.
6. Run this command to test the DSN:

```
isql <DSN-name> [<user-name> <password>] < commands.sql
```

To connect without DSN, use this DSN-less connection string:

```
ODBC;DRIVER={PostgreSQL};DATABASE=<vdb-name>;SERVER=<host-name>;PORT=<port>;Uid=<username>;Pwd=<password>
```

If you run `isql` but you encounter an error whereby you see this message: "Can't open lib '/opt/redhat/jboss-dv/v6/psqlodbc/lib64/psqlodbc.so' : file not found" it means that some of the postgres libraries are missing.

To fix this issue, run this command as root: **yum install postgres**

To verify that the packages are now installed, run this command: **rpm -qa|grep post**

You should see the `postgresql` and `postgresql-jdbc` packages listed.

## 11.9. Configure ODBC Options on Microsoft Windows

### Prerequisites

- ✦ You must have logged into the workstation with administrative rights.
- ✦ You need to have used the Control Panel's Data Sources (ODBC) applet to add a new data source name.

Each data source name you configure can only access one VDB within a Teiid System. To make more than one VDB available, you need to configure more than one data source name.

### Procedure 11.6. Configure the Data Source Name (DSN) on Microsoft Windows

1. From the **Start** menu, select **Settings - Control Panel**.
2. The **Control Panel** displays. Double-click **Administrative Tools**.
3. Double-click **Data Sources (ODBC)**.
4. The **ODBC Data Source Administrator** applet displays. Click the tab associated with the type of DSN you want to add.
5. The Create New Data Source dialog box displays. In the **Select a driver for which you want to set up a data source** table, select **PostgreSQL Unicode**.
6. Click **Finish**.
7. In the **Data Source Name** edit box, type the name you want to assign to this data source.
8. In the **Database** edit box, type the name of the virtual database you want to access through this data source.

- In the **Server** edit box, type the host name or IP address of your Teiid runtime.



### Note

If you are connecting via a firewall or NAT address, you must enter either the firewall address or the NAT address.

- In the **Port** edit box, type the port number to which the system listens for ODBC requests. (By default, Red Hat JBoss Data Virtualization listens for ODBC requests on port 35432.)
- In the **User Name** and **Password** edit boxes, supply the user name and password for the Teiid runtime access.
- Leave SSL Mode to disabled. (SSL connections are unsupported at present.)
- Provide any description about the data source in the **Description** field.
- Click on the **Datasource** button and configure the options. Tick Parse Statements, Recognize Unique Indexes, Maximum, Text as LongVarChar and Bool as Char and set MaxVarChar to 255, Max LongVarChar to 8190, Cache Size to 100 and SysTable Prefixes to dd\_:

On the second page, click LF, Server side prepare, default, 7.4+ and set the Extra Opts to 0x0.

- Click **Save**.

You can optionally click **Test** to validate your connection if Red Hat JBoss Data Virtualization is running.

**Table 11.1. Primary ODBC Settings for Red Hat JBoss Data Virtualization**

Name	Description
Updateable Cursors and Row Versioning	Should not be used.
Use serverside prepare and Parse Statements and Disallow Premature	It is recommended that "Use serverside prepare" is enabled and "Parse Statements"/"Disallow Premature" are disabled.
SSL mode	See Security Guide
Use Declare/Fetch cursors and Fetch Max Count	Should be used to better manage resources when large result sets are used.

Logging/debug settings can be utilized as needed.

Settings that manipulate datatypes, metadata, or optimizations such as "Show SystemTables", "True is -1", "Backend genetic optimizer", "Bytea as LongVarBinary", "Bools as Char", etc. are ignored by the Teiid server and have no client side effect. If there is a need for these or any other settings to have a defined affect, please open an issue with the product/project.

Any other setting that does have a client side affect, such as "LFto CR/LF conversion", may be used if desired but there is currently no server side usage of the setting.

## 11.10. DSN-less Connection

You can also connect to a Red Hat JBoss Data Virtualization VDB using ODBC without explicitly creating a DSN. However, in these scenarios your application needs a DSN-less connection string. You may want to do this if you are working with multiple computers and do not want to keep distributing the ODBC data source name.

Here is the string for Linux, UNIX, and similar operating systems:

```
ODBC;DRIVER={PostgreSQL};DATABASE=<vdb-name>;SERVER=<host-name>;PORT=
<port>;Uid=<username>;Pwd=<password>;c4=0;c8=1;
```

Here is the string for Windows:

```
ODBC;DRIVER={PostgreSQL Unicode};DATABASE=<vdb-name>;SERVER=<host -
name>;PORT=<port>;Uid=<username>;Pwd=<password>;c4=0;c8=1;
```

## Chapter 12. Running in Cloud Environments

### 12.1. Run Red Hat JBoss Data Virtualization in an Amazon AWS Cloud Instance

#### Procedure 12.1. Running Red Hat JBoss Data Virtualization in an Amazon Cloud

1. Open ports by updating the security group. (At a minimum, you will need to open the TCP, HTTP and SSH ports.)
2. To start the server, add the following parameters to bind the management and host ports - **Djboss.bind.address.management=0.0.0.0** and **-b 0.0.0.0** example  
`./standalone.sh -Djboss.bind.address.management=0.0.0.0 -b 0.0.0.0`

3. To access the AWS instance from Teiid Designer, go to the JBDS preferences and select **General**, **Network Connections SSH2**

Next, under the **Key Management** tab, use **Load Existing Key** to add the key generated by Amazon.

4. To create a server connection, on the **Server Configuration Overview Panel**, under **Server Behavior**, select **Remote System Deployment**. Also ensure you check **Server is externally managed...**

Click the **New Host** button, select the **SSH Only** option and click **Next**.

Set the Host Name to match the Amazon public IP address and make the connection name the same.

Click Finish

5. Open the **Remote Systems** tab.

Right mouse click the new connection and click **connect**. Fill in the User ID. (You do not need to provide a password if your SSH key is configured.)

6. Go back to the server configuration overview panel and confirm that the Host drop-down has selected the new host that you have created.
7. Start the server. (This switches the state of the server you already started.)

### 12.2. Run Red Hat JBoss Data Virtualization in a Google Compute Instance

#### Procedure 12.2. Run Red Hat JBoss Data Virtualization in a Google Compute Instance

1. Open the necessary ports: **Google Developers Console - Compute - Compute Engine - VM Instance - [name of your instance] - Network**.
2. Upload your public SSH key: **Google Developers Console - Compute - Compute Engine - VM Instance - [name of your instance] - SSH KEys**.
3. Bind the management ports (jboss.bind.address.management) to an external interface. (The default value for management ports is **localhost**).

### 12.3. Connecting Red Hat JBoss Data Virtualization to an Azure Instance

Follow these instructions to connect to your Azure database:

### Procedure 12.3. Connecting to Azure

1. Get the JDBC connection string for your Azure database. In the Azure management portal, click on the database then find the Connect to your database section.
2. Click on the View SQL Database connection strings... link. This will give you the JDBC strings you require.
3. The cloud database requires that the IP address of the machine accessing it be registered in its firewall rules. On the Azure database dashboard, click Manage allowed IP addresses, then add the IP address of your server. This is straightforward if the DV instance resides on your local server.
4. If the DV instance is deployed on OpenShift, ssh into your OpenShift instance. On the command line, enter `ping $OPENSIFT_GEAR_DNS`. The first line of the ping response will look like this:

```
PING ec2-54-221-126-53.compute-1.amazonaws.com
(10.181.128.66) 56(84) bytes of data.
```

You can infer the IP of your server from the ec2 name:

```
ec2-54-221-126-53 --> (54.221.126.53)
```

Register the derived ip address in the cloud db firewall rules.

5. JBoss Data Virtualization uses a model-driven approach. First, you will create a source model by connecting to the source and importing its structure.

In Teiid Designer, open the Teiid Designer perspective. Then create a new Model Project by selecting File - New - Teiid Model Project. On the first wizard page, enter MyProject for the project name - then click Finish. The project will be created.

6. In Model Explorer, click MyProject, then Right-Click - Import... - JDBC Database - Source Model. Click Next.
7. In the Import Database via JDBC wizard, click the New... button to create a new Connection Profile. Select Generic JDBC for the type. Enter AzureCP for the Name. Click Next.
8. Next to the Drivers combo box, select the New Driver Definition icon to create a new driver. Select the Generic JDBC Driver template, then enter AzureDriver for the Name.
9. On the JAR List tab, click the Add JAR/Zip... button, then select a SQL Server type 4 driver jar that you've previously downloaded to your file system. Click OK to finish creating the Driver Definition.
10. For the connection, enter the general properties using your database name and so forth as provided in the cloud db connection string.
11. Enter the Optional properties from the connection string.
12. Click the Test Connection button to ensure a successful connection. Click Finish.
13. Continue in the JDBC import wizard, selecting the tables and so forth that you want to import. Choose a name for your source model (such as AzureSourceModel). Upon completion of the import wizard, the source model will be created within your model project.

After the source model has been created, you can now preview data from the cloud database source:



**Procedure 12.4. Previewing Data**

1. In Designer, make sure your server is running.
2. Select a table, then click the 'running man' icon in the toolbar. This will display the content in your cloud database table.
3. You can also create more complex transformations using the source model that you just created: Create additional views which transform and join sources in any way you like.

## Chapter 13. Running Red Hat JBoss Data Virtualization Cartridge on OpenShift Online v2

### 13.1. Cartridge Installation

The Data Virtualization Cartridge provides Teiid, ModeShape and the Dashboard Builder for the OpenShift environment.

1. Create an OpenShift user account.
2. Launch the OpenShift Web Console and go to the Applications page at <https://openshift.redhat.com/app/console/applications>
3. Go to the Applications tab and select **Add Application...**
4. Go to the xPaaS section and select the JBoss Data Virtualization cartridge.
5. Enter the name of the application (for example, "jbossdv").
6. If your account allows it, select a medium gear.
7. Click the **Create Application** button.
8. The Red hat JBoss Data Virtualization Cartridge will now deploy.



#### Important

Copy and save the username and password information somewhere secure for future use.

9. Click on **Continue to the application overview page** to see the Data Virtualization cartridge overview.

Verify that the status is **Started**.

10. If you click on the application link you will see the Data Virtualization Welcome page. This page contains cartridge information and some helpful links.
11. After the cartridge has deployed, go to this address to check its status: [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com](http://[MYAPP]-[MYDOMAIN].rhcloud.com)

A user has been automatically generated with user, odata and rest roles.

12. Access ModeShape at [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/modeshape-webdav](http://[MYAPP]-[MYDOMAIN].rhcloud.com/modeshape-webdav) and [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/modeshape-rest](http://[MYAPP]-[MYDOMAIN].rhcloud.com/modeshape-rest)

Two ModeShape users, msuser and msadmin, are generated with the installation.

13. Access the Dashboard Builder at [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/dashboard](http://[MYAPP]-[MYDOMAIN].rhcloud.com/dashboard)

A dashboard administrator, dbadmin, is generated with the installation. (The Teiid 'user' is allowed dashboard read-only user access).



## Important

To obtain your password, ensure that you add your public OpenShift ssh key to the application's main page and then connect to the machine via ssh. Click "Want to log in to your application?" and, once you are connected to the machine, open a terminal and run "env | grep PASSWORD" to obtain the password.

## 13.2. Use the Data Virtualization Web Interface on OpenShift Online v2

### Procedure 13.1. Install a Box Grinder plug-in

1. Firstly, assign a MySQL database to your OpenShift Data Virtualization instance. Go to the overview page for your Data Virtualization application. Click on the **Add MySQL 5.5** link (found under the Databases section.)

The MySQL cartridge will deploy. When complete, you will see a success screen



## Important

Ensure that you save the credentials information for later reference.

2. Under **Tools and Support**, click on the **Add phpMyAdmin 4.0** link. This adds the web interface for easy management of your MySQL database.

The MySQL database and management interface deployments are now complete.

3. Load some data into the database using the phpMyAdmin interface. Use the link that was displayed when you installed the phpMyAdmin cartridge at <https://myApp-myDomain.rhcloud.com/phpmyadmin>

Use the Root User and Root Password you received when the cartridge was installed.

In the left tree panel, click on the database that matches the name of your DV application (for instance, jbossv1). Select the **SQL** tab. On the SQL tab, copy this DDL and paste it into the text area:

```
CREATE TABLE PricesTable
(
 ProdCode CHAR(40) NOT NULL,
 SalePrice DECIMAL,
 PRIMARY KEY (ProdCode)
);

INSERT INTO PricesTable VALUES
('GC1020', 3499.0),
('GC1040', 19990.0),
('GC1060', 75000.0),
('GC3020', 10200.0),
('GC3040', 38000.0),
('GC3060', 95000.0),
('GC5020', 28000.0),
('GC5040', 59900.0),
('GC5060', 110000.0),
```

```
('IN7020' , 4000.0),
('IN7040' , 16000.0),
('IN7060' , 42000.0),
('IN7080' , 69000.0),
('SL9020' , 4999.0),
('SL9040' , 9999.0),
('SL9060' , 14999.0),
('SL9080' , 19999.0);
```

Finally, click the **GO** button on the far right of the management interface.

The PricesTable table has now been created and populated. You can verify the contents by clicking on it.

- Restart the Data Virtualization cartridge. This is required to initialize the MySQL datasource. Click the **Restart application** icon on the application overview page
- Go to <http://www.developerforce.com/events/regular/registration.php?d=70130000000EjHb> , to register for a Salesforce account. (You will receive an email with a link to login into your account.)
- Log in to Salesforce interface with your user name and password, then go to **Personal Setup - Reset My Security Token**. Reset the security token. You will get another email with the security token.
- When you establish connectivity to the Salesforce instance in the example below, use the Salesforce username. (The password is the combination of Salesforce password and security token.)
- go to the deployed Web interface on OpenShift at <http://APP-NAME-DOMAIN-NAME.rhcloud.com/dv-ui>. (APP-NAME is the name you gave the application and DOMAIN-NAME is your OpenShift domain name).
- From the **Data Library** screen, click the **Create Data Services** link. You will be directed to the **Create Data Service** page.
- Click on the **Manage Sources** button. You will be directed to the **Manage Data Sources** page
- Click on the MySQLDS source: it is configured, but inactive. Select the mysql5 translator for the source and click the **Save Changes** button. Upon deployment, the source should become active (a green check mark will appear).
- Click the Add Source button. This will create a default H2 source called MyNewSource.
- Click on MyNewSource.

In the displayed properties, enter SalesforceDS for the Name, Under Source Type, click on the salesforce button. Click the Ok button as this will change the source to salesforce type and set the translator to salesforce. Under Connection Properties, enter the user name for your salesforce account. Enter the password and token combination for the password.

- Click on the Save Changes button. Click Ok for each dialog box. The salesforce source will deploy (it will take a couple of minutes to finish).
- Click the back link to go back to the Create Data Service page
- The next phase is to create a data service which accesses the MySQL database table. Enter MySQLService for the service name. Enter a description for the service.
- Click on MySQLDS on the Service Helpers Active Sources tab.

18. Click on the dv61.PricesTable under Tables, then select both columns via the checkboxes.
19. To create the Service View Definition, click the Create Service View button. This will populate the Service View Definition text area.
20. Click the Test Service button to test the service.
21. Click Create Data Service. This will accept your entries and create the service. You will be redirected to the Data Service Details page.
22. The steps to create a salesforce-only service follow the same pattern. Go back to the Data Library then click the Create Data Service button to create a new service.
23. Enter SalesforceProductService for the service name. Enter a description for the service.
24. Click on SalesforceDS on the Service Helpers Active Sources tab
25. Click on Product2 under Tables, then select the Name and ProductCode columns via the checkboxes.
26. To create the Service view definition, click the Create Service View button. This will populate the Service View Definition text area.
27. Test the service as before to see sample data, then click Create Service to create the service.
28. The steps to create a mashup service once more follow the same pattern. Go back to the Data Library then click the Create Data Service button to create a new service.
29. Enter MashupService for the service name. Enter a description for the service.
30. In the Service Helpers section, click on the Join Definition tab. (It is here that you define the join.)
31. Select MySQLDS in the Available Sources drop-down. Under Source Tables, click the PricesTable. Then click the left-hand button to specify it as the Left table.
32. Select SalesforceDS in the Available Sources drop-down. Under Source Tables, click the Product2 table. Then click the right-hand button to specify it as the Right table.
33. In the left-hand side Prices table, check the ProdCode and SalePrice columns. In the right-hand side Product2 table, check the Name column.
34. Leave the JoinType selection on Inner Join.
35. Select ProdCode for the left-hand side Join Criteria column.
36. Select ProductCode for the right-hand side Join Criteria column.
37. Click the Apply button to generate the Service Definition DDL.
38. Click the Test Service button to see example data.
39. Click the Create Service button to create the Mashup service. You will be redirected to the Data Service Details page. The Data Service Details page shows a sample of data and also provides connection instructions and URLs for the different connection options.
40. Click the **Back to Library** link to go back to your Data Library. On the Data Library page you will see the three services that you just created.



## Note

Notice that for each service, there are more actions available to you:

- ✦ Edit Service - redirects to the Edit Data Service page.
- ✦ Duplicate Service - makes a copy of the selected service.
- ✦ Test Service - redirects to the Data Service Details page.
- ✦ Delete Service - delete the selected service.
- ✦ Save to File - A service is backed by a teiid 'Dynamic VDB'. This action saves the dynamic VDB xml to a file.

## Appendix A. Configuration Information

### A.1. Recommended Translators for Data Sources

The following table provides a list of data sources and translators that are supported by Red Hat.

**Table A.1. Recommended Translators for Data Sources**

Data Source	Driver	Translator
Actian Analytics Express 2.0	2.0	actian-vector
Amazon Redshift	postgresql 9.2	redshift
Apache Accumulo 1.5.0	N. A.	accumulo
Apache Cassandra 2.2.4	N. A.	cassandra
Apache Hive 2.0.0	2.0.0	hive
Apache Solr 4.9.0	N. A.	solr
Apache Spark 1.6.0	Hive 1.2.1	hive
Cloudera Hadoop 5.5.1	5.5.1	impala
EDS 5.x	current Teiid Driver	teiid
Files – delimited, fixed length	N. A.	file
Generic Datasource-JDBC ansi	N. A.	jdbc-ansi
Generic Datasource-JDBC simple (postgresql84)	postgresql 8.4	jdbc-simple
Greenplum 4.x	postgresql 9.0	greenplum
HBase 1.1	phoenix 4.5.1 HBase 1.1	hbase
Hortonworks Hadoop	2.3.4	hive
IBM DB2 10	4.12.55	db2
IBM DB2 9.7	4.12.55	db2
Informix 12.10	4.10.JC5DE	informix
Ingres 10	4.1.4	ingres
JBoss Data Grid 6.4 (library mode)	N. A.	infinispan-cache
JBoss Data Grid 6.4 (remote client - hotrod)	N. A.	infinispan-cache-dsl
LDAP/ActiveDirectory v3	N. A.	ldap
MariaDB	mysql 5.1.22	mysql5
ModeShape/JCR 3.1	3.8.4	modeshape
MongoDB 3.0	N. A.	mongodb
MS Access 2013	N. A.	access
MS Excel 2010	N. A.	excel
MS Excel 2013	N. A.	excel
MS SQL Server 2008	4.0.2206.100	sqlserver
MS SQL Server 2012	4.0.2206.100	sqlserver
MySQL 5.1	mysql 5.1.22	mysql5
MySQL 5.5	mysql 5.1.22	mysql5
Netezza 7.2.x	7.2.1.1	netezza
Oracle 11g RAC	12.1.0.2.0	oracle
Oracle 12c	12.1.0.2.0	oracle
PostgreSQL 8.4	postgresql 8.4	postgresql

Data Source	Driver	Translator
PostgreSQL 9.2	postgresql 9.2	postgresql
REST/JSON over HTTP	N. A.	ws
RHDS 9.0	N. A.	ldap
Salesforce.com API 22.0	NA	salesforce
Salesforce.com API 34.0	NA	salesforce-34
SAP_Hana 1.00.102.01.1444147999	1.00.82.00_03942 70-1510	hana
SAP Netweaver Gateway	odata 4	sap-nw-gateway
Sybase ASE 15	jconn4-26502	sybase
Sybase IQ 16 Express	jconn4-v7	sybaseIQ
Teradata Express 15	15.10.00.05	teradata
Vertica 7.2.1	7.2.1-0	vertica
Webservices	N. A.	ws
XML Files	N. A.	FILE



### Note

MS Excel is supported in so much as there is a write procedure.



### Note

The MySQL InnoDB storage engine is not suitable for use as an external materialization target.



## Appendix B. Revision History

<b>Revision 6.30-16</b>	<b>Wed Oct 26 2016</b>	<b>David Le Sage</b>
Updates for 6.3.		
<b>Revision 6.2.0-032</b>	<b>Thu Dec 10 2015</b>	<b>David Le Sage</b>
Updates for 6.2.		