Red Hat JBoss Data Virtualization 6.2

Installation Guide

This guide is for installation teams.

Last Updated: 2017-10-20

Red Hat JBoss Data Virtualization 6.2 Installation Guide

This guide is for installation teams.

Red Hat Customer Content Services

Legal Notice

Abstract

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

Table of Contents

CHAPTER 1. READ ME

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.

1.2. VARIABLE NAME: EAP_HOME

`EAP_HOME` refers to the root directory of the Red Hat JBoss Enterprise Application Platform installation on which JBoss Data Virtualization has been deployed.

1.3. VARIABLE NAME: MODE

`MODE` will either be **`standalone`** or **`domain`** depending on whether JBoss Data Virtualization is running in standalone or domain mode. Substitute one of these whenever you see **`MODE`** in a file path in this documentation. (You need to set this variable yourself, based on where the product has been installed in your directory structure.)

1.4. RED HAT DOCUMENTATION SITE

Red Hat's official documentation site is available at https://access.redhat.com/site/documentation/. There you will find the latest version of every book, including this one.

CHAPTER 2. SOME KEY TERMS USED THROUGHOUT THIS DOCUMENT

2.1. DATA FEDERATION

The Data Federation example demonstrates the ability of JBoss Data Virtualization to integrate multiple data sources so that a single query can return results from one or more data sources. This makes data integration of new sources easier without having to expose any new access details.

> **NOTE**
>
> All the code discussed in this example is available in the dynamicvdb-datafederation quickstart. See the ***EAP_HOME*/`quickstarts/dynamicvdb-datafederation/README.md`** for directions to run the quickstart.

In this example, two different data sources are used:

- Account information, stored in H2 database
- Market Data (stock prices) stored in CSV text file

DynamicPortfolio VDB will define a model for each data source. Since VDB is deployed to JBoss Data Virtualization, it is accessible to the user application using JDBC. When the test query is executed that joins data from the Accounts and MarketData models, JBoss Data Virtualization will access both, the relational and non-relational sources, calculate the portfolio values, and return the results.

2.2. DATA ROLES

The Data Roles example demonstrates how data roles can be used to control access to data. This includes read-only and read-write access roles.

> **NOTE**
>
> All the code discussed in this example is available in the dynamicvdb-dataroles quickstart. See the ***EAP_HOME*/`quickstarts/dynamicvdb-dataroles/README.md`** for directions to run the quickstart.

In this example, the VDB (**`portfolio-vdb.xml`**) is defined with the two different data access roles:

- read-only

  The read-only role restricts access of the VDB to read only access (that is, select). This role is assigned to all the users who have a login credential (you can use the user called 'user' with password 'user').

- read-write

  The read-write role grants read and write access (i.e., insert/update/delete). This role is assigned only to the users with 'superuser' JAAS role (you can use the 'portfolio' user with password 'portfolio').

Following is an example of the read-write data-role in the **`portfolio-vdb.xml`**:

```
<data-role name="ReadWrite">
 <description>Allow Reads and Writes to tables and
procedures</description>

 <permission>
  <resource-name>Accounts</resource-name>
  <allow-create>false</allow-create>
  <allow-read>true</allow-read>
  <allow-update>true</allow-update>
 </permission>

 <permission>
  <resource-name>MarketData</resource-name>
  <allow-create>false</allow-create>
  <allow-read>true</allow-read>
  <allow-update>true</allow-update>
 </permission>

 <!--
  This role must defined in the JAAS security domain, the sample
UserRolesLoginModules based roles file is provided in this sample
directory. Copy these teiid-security-roles.properties and teiid-security-
users.proeprties into
EAP_HOME/modules/system/layers/dv/org/jboss/teiid/conf directory and
replace the old ones.
 -->

 <mapped-role-name>supervisor</mapped-role-name>
</data-role>
```

To see how the users and roles are defined for JAAS, see the ***EAP_HOME*****/quickstarts/src/dynamicvdb-dataroles/security/teiid-security-roles.properties** and ***EAP_HOME*****/quickstarts/src/dynamicvdb-dataroles/security/teiid-security-users.properties** files. The teiid-security-users.properties file is used by JAAS to determine user credentials (that is, login ID and Password) and the teiid-security-roles.properties file maps the username to the **<mapped-role-name>** element.

2.3. ABOUT MAVEN

Apache Maven is a distributed build automation tool used in Java application development to build and manage software projects. Maven uses configuration XML files called POM (Project Object Model) to define project properties and manage the build process. POM files describe the project's module and component dependencies, build order, and targets for the resulting project packaging and output. This ensures that projects are built in a correct and uniform manner.

Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, such as **http://** when located on an HTTP server, or **file://** when located on a file server. The default repository is the public remote Maven 2 Central Repository.

Configuration of Maven is performed by modifying the **settings.xml** file. You can either configure global Maven settings in the ***M2_HOME*****/conf/settings.xml** file, or user-level settings in the ***USER_HOME*****/.m2/settings.xml** file.

For more information about Maven, see Welcome to Apache Maven.

For more information about Maven repositories, see Apache Maven Project - Introduction to Repositories.

For more information about Maven POM files, see the Apache Maven Project POM Reference.

NOTE

Your Red Hat JBoss product has been built with maven 3.0.x Therefore, this is the recommended maven version for building your own SwitchYard applications.

2.4. CHECKSUM VALIDATION

Checksum validation is used to ensure a downloaded file has not been corrupted. Checksum validation employs algorithms that compute a fixed-size datum (or checksum) from an arbitrary block of digital data. If two parties compute a checksum of a particular file using the same algorithm, the results will be identical. Therefore, when computing the checksum of a downloaded file using the same algorithm as the supplier, if the checksums match, the integrity of the file is confirmed. If there is a discrepancy, the file has been corrupted in the download process.

2.5. VIRTUAL DATABASES

A virtual database (VDB) is a container for components that integrate data from multiple disparate data sources, allowing applications to access and query the data as if it is in a single database and, therefore, using a single uniform API.

2.6. DATA SOURCES

A data source is a repository for data. Query languages enable users to retrieve and manipulate data stored in these sources.

2.7. TEIID DESIGNER

Teiid Designer is an Eclipse-based graphical modeling tool for modeling, analyzing, integrating and testing multiple data sources. The Teiid Designer produces Relational, XML and Web Service Views.

Functions of the Teiid Designer are to:

- resolve semantic differences.
- create virtual data structures at a physical or logical level.
- use declarative interfaces to integrate, aggregate, and transform the data from source to a target format for application compatability.

2.8. CONNECTOR FRAMEWORK

Red Hat JBoss Data Virtualization includes a set of *translators* and *resource adapters* that enable VDBs to access their dependent physical data sources.

Custom translators and resource adapters can be developed for data sources that are not directly supported by Red Hat JBoss Data Virtualization.

2.9. TRANSLATORS

In Red Hat JBoss Data Virtualization, a translator provides an abstraction layer between the *query engine* and the physical data source. This layer converts query commands into source specific commands and executes them using a *resource adapter*. The translator also converts the result data that comes from the physical source into the form that the query engine requires.

2.10. SERVICE

A Red Hat JBoss Data Virtualization service is positioned between business applications and one or more data sources, and co-ordinates the integration of those data sources for access by the business applications at runtime.

A Red Hat JBoss Data Virtualization service manages the following components:

Virtual Database

A virtual database (VDB) provides a unified view of data residing in multiple physical repositories. A VDB is composed of various data models and configuration information that describes which data sources are to be integrated and how. In particular, *source models* are used to represent the structure and characteristics of the physical data sources, and *view models* represent the structure and characteristics of the integrated data exposed to applications.

Access Layer

The access layer is the interface through which applications submit queries (relational, XML, XQuery and procedural) to the VDB via JDBC, ODBC or Web services.

Query Engine

When applications submit queries to a VDB via the access layer, the query engine produces an optimized query plan to provide efficient access to the required physical data sources as determined by the SQL criteria and the mappings between source and view models in the VDB. This query plan dictates processing order to ensure physical data sources are accessed in the most efficient manner.

Connector Framework

Translators and *resource adapters* are used to provide transparent connectivity between the query engine and the physical data sources. A translator is used to convert queries into source-specific commands, and a resource adapter provides communication with the source.

2.11. OPEN DATABASE CONNECTIVITY (ODBC)

Open Database Connectivity (ODBC) is a standard C programming language middleware API for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of database systems and operating systems; an application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

2.12. ROLE OF THE DATABASE TOOL

The database in Red Hat JBoss Data Virtualization is required by three items: ModeShape, the Dashboard Builder and the command/audit logging functionality. If you are using Red Hat JBoss Data Virtualization in conjunction with Red Hat JBoss Fuse Service Works, BPEL, and SwitchYard/jBPM integration also make use of this functionality.

CHAPTER 3. PLATFORM REQUIREMENTS

3.1. 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 recommdendation, 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
- 15 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.
- 4 – represents 1) the doubling of batch bytes and 2) plan operations – where this is the number additional copies of the batches that are needed to fulfill plan operations (i.e., sort, joining, etc.). 2 will be used.
- source queries per plan – number of data sources in the query, but limited based on the assumptions.
- approximate source bytes – **`( ( (32bit?5:7) + (4 * avg raw bytes per column) ) * #columns)`**, using 10 as an average raw bytes per column and using a 64-bit machine
- 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 3.1. Table

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 Teiid 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 conncurrent source queries (minimum would be max conncurrent 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 sourse 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.

IMPORTANT

If maxThreads is more than 5 to 1 to maxActivePlans, then consider making adjustments. We've seen 10 to 1 cause the server to throttle down processing and the initial recommendations above are 2 or 3 to 1.

CHAPTER 4. DOWNLOAD THE PRODUCT

4.1. RED HAT CUSTOMER PORTAL

The Red Hat Customer Portal is a website located at https://access.redhat.com/home. It provides a central point from where you can manage and maintain your subscription, access the Red Hat Knowledge Base and engage with Red Hat and our partners.

4.2. DOWNLOAD JBOSS DATA VIRTUALIZATION INSTALLER

The JBoss Data Virtualization installer archive is available on the Red Hat Customer portal at https://access.redhat.com/. You can use the **`.jar`** archive to run the graphical installer. This topic covers the steps to download the archive.

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 or 7 JDK is required.

Procedure 4.1. Download JBoss Data Virtualization Installer

1. **Login to the Red Hat Customer Portal.**
   a. Enter https://access.redhat.com/ into the address bar of a browser.
   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.

4.3. VERIFY DOWNLOADED FILES

Procedure 4.2. Verify File Checksums on Red Hat Enterprise Linux

1. **Obtain checksum values for the downloaded file**
   a. Navigate to https://access.redhat.com/jbossnetwork/. Login if required.

   b. Select your **`Product`** and **`Version`**.

   c. Select the package you want to verify to 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-6.0.0.GA-redhat-4.jar
   MD5 (jboss-dv-installer-6.0.0.GA-redhat-4.jar) =
   0d1e72a6b038d8bd27ed22b196e5887f
   [localhost]$ shasum jboss-dv-installer-6.0.0.GA-redhat-4.jar
   a74841391bd243d2ca29f31cd9f190f3f1bdc02d  jboss-dv-installer-
   6.0.0.GA-redhat-4.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.

Result

The integrity of the downloaded file is verified.

NOTE

No checksum tool is included with Microsoft Windows. Download a third-party MD5 application such as MD5summerfrom http://www.md5summer.org/.

CHAPTER 5. MAVEN REPOSITORIES

5.1. 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 storage 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 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.

5.2. CONFIGURE MAVEN TO USE THE FILE SYSTEM REPOSITORIES

Overview

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 5.1.

1. Download the repository from this address: https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?product=data.services.platform

2. Click the **Download** button to download a **.zip** file containing the repository.

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-6.1.0.GA-redhat-2-repository</id>
          <name>DV 6.1.0 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-dv-
6.1.0.redhat-3-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-6.1.0.GA-redhat-2-repository</id>
          <name>DV 6.1.0 GA Repository</name>
          <url>file://<!-- path to the repository -->/jboss-dv-
6.1.0.redhat-3-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>
```

Result

The Maven repositories are downloaded, unzipped in a local file system, registered in Maven's **settings.xml** file, and ready to be used when performing Maven builds.

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**

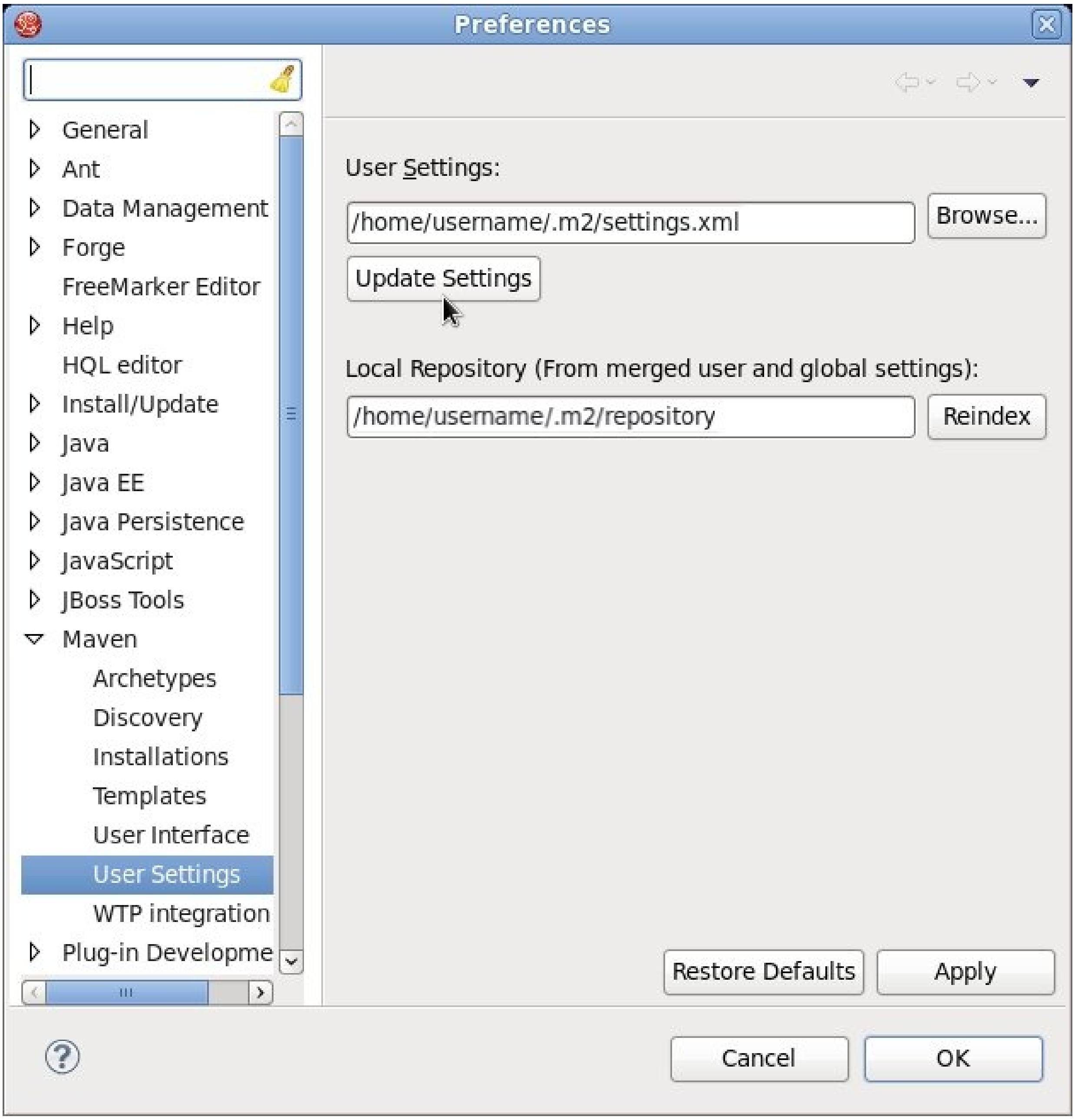
You have updated your Maven configuration, but this configuration is not reflected in JBoss Developer Studio.

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.



User Settings: /home/username/.m2/settings.xml

Local Repository (From merged user and global settings): /home/username/.m2/repository

Figure 5.1. Update Maven User Settings

5.3. CONFIGURE MAVEN TO USE THE ONLINE REPOSITORIES

The online repositories required for Red Hat JBoss Data Virtualization is 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.

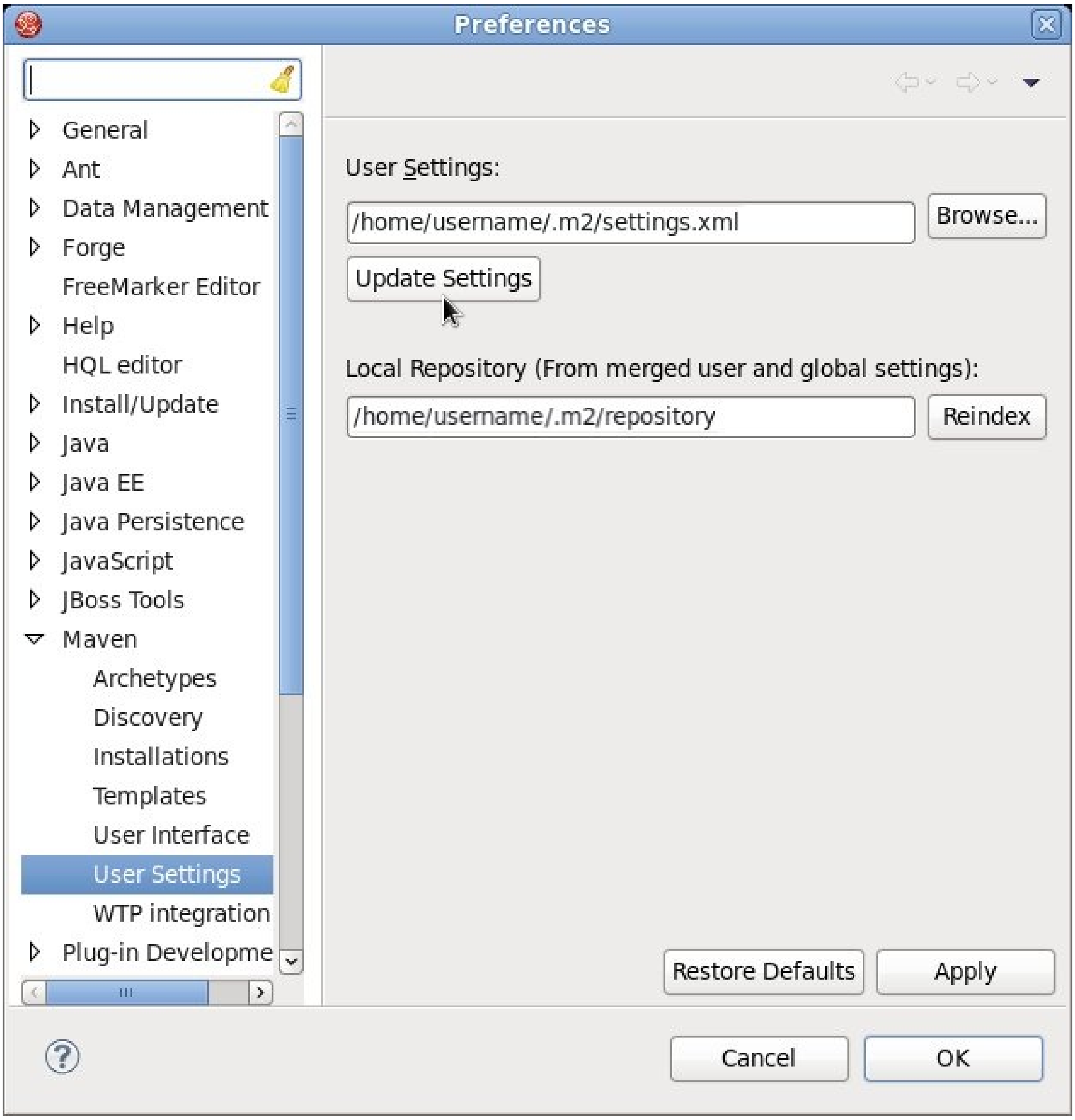
If you did not configure the Maven repository during installation, you can configure it using the following procedure. (It is also possible to do this using the project's POM file, but this is not recommended.)

Procedure 5.2. Configuring Maven to Use the Online Repositories

1. Add entries for the online repositories to Maven's **`settings.xml`** file as in the code sample below:

```
<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.



User Settings: /home/username/.m2/settings.xml

Local Repository (From merged user and global settings): /home/username/.m2/repository

Figure 5.2. Update Maven User Settings

Result

Maven has been configured to use the online repositories provided for your Red Hat JBoss product.

IMPORTANT

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.

5.4. DEPENDENCY MANAGEMENT

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-2.2.0.redhat-7.pom**

The BOM file to use is **org.jboss.dv.component.management:dv-dependency-management-all-2.2.0.redhat-7.pom**.

5.5. USING MAVEN DEPENDENCIES FOR RED HAT JBOSS DATA VIRTUALIZATION

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.

Here is a sample:

```
<?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 6.2.0 and EAP 6.4 component
versions.
          -  Parent is set to the DV 6.2.0 parent management POM, which will
          -  bring in the correct toolchain (plugin) versions.
          -  DependencyManagement dependencies include the DV 6.2.0 and EAP
6.4
          -  BOMs -  which will bring in the correct compile-time (and other
          -  scoped) versions.
          -->

    <name>Example POM for DV 6.1.0</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>2.2.0.redhat-7</version>
    </parent>

    <dependencyManagement>
        <dependencies>
            <!-- DV BOM -->
            <dependency>
                <groupId>org.jboss.dv.component.management</groupId>
                <artifactId>dv-dependency-management-all</artifactId>
                <version>2.2.0.redhat-7</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>

        </dependencies>
    </dependencyManagement>

</project>
```

5.6. OFFLINE MODE

5.6.1. Use case for 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.

5.6.2. Download repository for Maven project

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.

5.6.3. Generate the offline repository

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 6. INSTALLING PREREQUISITE COMPONENTS

6.1. INSTALL OPEN JDK ON RED HAT ENTERPRISE LINUX

Procedure 6.1. 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`**.

6.2. INSTALL MAVEN

Prerequisites

The following software must be installed:

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

Procedure 6.2. 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.**

   - **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
```

- **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
```

Result

Maven is successfully installed and configured on your machine.

CHAPTER 7. INSTALLING THE PRODUCT

7.1. RED HAT JBOSS DATA VIRTUALIZATION INSTALLATION METHODS

There are three different ways to install Red Hat JBoss Data Virtualization.

- Graphical mode: Graphical mode launches a graphical wizard which provides step-by-step instructions for installing and configuring the Red Hat JBoss Data Virtualization. Additional setup, including the Quickstarts and Maven Repository, is also possible with the installer.
- Text mode: You can launch the installer in the text mode as well. Text mode provides step-by-step instructions for installing and configuring the Red Hat JBoss Data Virtualization.
- Automated script mode: You can install multiple identical instances of Red Hat JBoss Data Virtualization using the automated script. This automated script is generated after the first installation instance.

7.2. INSTALLING JBOSS DATA VIRTUALIZATION: BASIC 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 a Red Hat JBoss EAP 6.4 instance ready. Download the installer version (not the Zip) from the Red Hat website and follow the defaults. The EAP Getting Started Guide can be found here: https://access.redhat.com/site/documentation/en-US/JBoss_Enterprise_Application_Platform/6.4/html/Getting_Started_Guide/index.html. Once installed, you must patch it to 6.4.3 or greater.
- You must have already downloaded the **Red Hat JBoss Data Virtualization** jar file from the Customer Portal.
- You must have a JVM installed, such as OpenJDK or the non-open source Azul Zing.

Procedure 7.1. 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-installer-VERSION.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. Add the filepath and 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.

Result

Red Hat JBoss Data Virtualization is successfully installed and configured.

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.

7.3. 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 7.2. 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-installer-VERSION.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.

Result

The Red Hat JBoss Data Virtualization is successfully installed and configured.

7.4. 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 7.3. Install JBoss Developer Studio Integration Stack

1. Start your **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.jboss.com/updates/8.0/integration-stack/ in the `Location` field. (For users of JBDS version 7, leave the URL as https://devstudio.jboss.com/updates/7.0/integration-stack/ and use the `Check for Updates` feature).

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.

Result

Teiid Designer and ModeShape Tools are installed successfully.

CHAPTER 8. STARTING AND STOPPING THE PRODUCT

8.1. STARTING JBOSS DATA VIRTUALIZATION

Prerequisites

The following software must be installed:

- Red Hat JBoss Data Virtualization

Procedure 8.1. 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**:

  - **Red Hat Enterprise Linux**
    Open a terminal and enter the command:

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

  - **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*.

8.2. 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.

Procedure 8.2. 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.

Result

The JBoss EAP 6 Managed Domain instance starts.

8.3. 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.*

Result

The server will shut down. Note that this process will take a few minutes.

CHAPTER 9. INSTALLING JBOSS DATA GRID CACHING

9.1. CONFIGURE 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

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 10. ODBC SUPPORT

10.1. INTRODUCTION

10.1.1. JBoss Data Virtualization and ODBC

JBoss Data Virtualization provides Open Database Connectivity (ODBC) access to deployed VDBs by emulating the PostgreSQL database server.

ODBC-compliant client applications can connect to a VDB using the supported PostgreSQL ODBC driver. The supported PostgreSQL ODBC driver is available for download at http://access.redhat.com.

Before an application can use ODBC, you must first install the ODBC driver on the machine the application is running on and create a Data Source Name (DSN) that represents a connection profile for your VDB.

> **WARNING**
>
> JBoss Data Virtualization currently only supports plaintext password authentication for ODBC. If the client/server are not configured to use SSL, the password will be sent in plaintext over the network. If you need secure passwords in transit and are not using SSL, then consider installing a security domain that will accept safe password values from the client (for example encrypted or hashed).

10.2. INSTALL THE ODBC DRIVER ON RED HAT ENTERPRISE LINUX

10.2.1. Install the ODBC Driver on Red Hat Enterprise Linux

Prerequisites

- Administrative permissions are required.

Procedure 10.1. 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/.

10.2.2. 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.

10.2.3. 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.

10.3. INSTALL THE ODBC DRIVER ON MICROSOFT WINDOWS

10.3.1. Install the ODBC Driver on Microsoft Windows

Prerequisites

- Administrative permissions are required.

Procedure 10.2. 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.

10.3.2. Configure the DSN for Windows Installation

Procedure 10.3. 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.

10.4. INSTALL THE ODBC DRIVER ON SOLARIS

10.4.1. Install the ODBC Driver on Solaris

Prerequisites

- Administrative permissions are required.

Procedure 10.4. Install the ODBC Driver on Solaris

1. **Download the driver**
   Download the correct ODBC driver package (**`jboss-dv-psqlodbc-6.0.0-`*`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.

10.4.2. 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.

10.5. ODBC SETUP

10.5.1. 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 10.5. 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.
9. 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.

10. 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.)
11. In the **`User Name`** and **`Password`** edit boxes, supply the user name and password for the Teiid runtime access.
12. Leave SSL Mode to disabled. (SSL connections are unsupported at present.)
13. Provide any description about the data source in the **`Description`** field.

14. 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, Updatable Cursors, bytea as LO, default, 7.4+ and set the Extra Opts to 0x0.

15. Click **`Save`**.

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

Result

You can now use applications such as Excel, Access to query the data in the VDB

10.5.2. Configure ODBC Options on Red Hat Enterprise Linux

Procedure 10.6. 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 PostGreSQLdriver 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
```

NOTE

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>
```

10.6. DSN LESS CONNECTION

You can also connect to JBoss Data Virtualization VDB using ODBC without explicitly creating a DSN. However, in these scenarios your application needs DSN less connection string. The below is a sample connection string.

For Windows:

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

For *nix:

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

CHAPTER 11. AUTOMATED INSTALLATION

11.1. 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 11.1. Installing with a Script

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

Result

The product is successfully installed and configured as specified by the script.

CHAPTER 12. RUNNING IN CLOUD ENVIRONMENTS

12.1. RUN RED HAT JBOSS DATA VIRTUALIZATION IN AN AMAZON AWS CLOUD INSTANCE

After following the standard installation procedure, undertake the following steps to use Red Hat JBoss Data Virtualization with AWS.

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

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`**.

CHAPTER 13. RUNNING RED HAT JBOSS DATA VIRTUALIZATION CARTRIDGE ON OPENSHIFT

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

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

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

    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

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, jbossdv1). 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.

4. Restart the Data Virtualization cartridge. This is required to initialize the MySQL datasource. Click the **Restart application** icon on the application overview page

5. 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.)

6. Log in to Salesforce interface with your user name an dpassword, then go to **`Personal Setup - Reset My Security Token`**. Reset the security token. You will get another email with the security token.

7. 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.)

8. 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).

9. From the **`Data Library`** screen, click the **`Create Data Services`** link. You will be directed to the **`Create Data Service`** page.

10. Click on the **`Manage Sources`** button. You will be directed to the **`Manage Data Sources`** page

11. 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).

12. Click the Add Source button. This will create a default H2 source called MyNewSource.

13. 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.

14. 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).

15. Click the back link to go back to the Create Data Service page

16. The next phase is to carete a data service which accesses the MySQL database table. Enter MySQLService for the service name. Enter a description for the service.

17. 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, the 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. INSTALLING PREREQUISITE COMPONENTS

A.1. INSTALL OPEN JDK ON RED HAT ENTERPRISE LINUX

Procedure A.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`**.

APPENDIX B. CONFIGURATION INFORMATION

B.1. SUPPORTED DATA SOURCES AND TRANSLATORS

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

Table B.1. Supported Data Sources and Translators

Data Source	Translator	Supported DV Version	Driver
Apache Hive 12	-	6.0+	-
Apache Solr	solr	6.1+	-
Cloudera Hadoop	-	6.1+	-
EDS 5.x	teiid	6.0+	-
Files – delimited, fixed length	file	6.0+	-
Generic Datasource-JDBC ansi	jdbc-ansi	6.0+	-
Generic Datasource-JDBC simple	jdbc-simple	6.0+	-
Google Spreadsheet	-	6.0+	-
Greenplum 4.x	postgresql	6.0+	-
Hortonworks Hadoop	-	6.1.+	-
HSQL (for test/examples only)	-	-	-
IBM DB2 10	db2	6.1+	Universal Driver v4.x
IBM DB2 9.7	db2	6.0+	Universal Driver v4.x
Ingres 10	ingres	6.0+	-
Intel Hadoop	-	6.1+	-
JBoss Data Grid 6.4 (remote client - hotrod)	infinispan-cache-dsl	6.2+	-
JBoss Data Grid 6.4 (library mode)	infinispan-cache	6.0 - post GA, 6.1+	-
LDAP/ActiveDirectory v3	ldap	6.0+	-
Mainframe (CICS,IMS,VSAM)	-	6.0+	-

Data Source	Translator	Supported DV Version	Driver
MariaDB	mysql5	6.1+	-
ModeShape/JCR 3.1	-	6.0+	-
MongoDB 2.2	mongodb	6.0	post GA, 6.1+ -
MS Access 2010	-	6.0+	-
MS Access 2013	-	6.0+	-
MS Excel 2010	excel	6.0+	-
MS Excel 2013	excel	6.0+	-
MS SQL Server 2008	sqlserver	6.0+	Microsoft SQL Server JDBC Driver 4
MS SQL Server 2012	sqlserver	6.0+	Microsoft SQL Server JDBC Driver 4
MySQL 5.1	mysql5	6.0+	V5.1
MySQL 5.5	mysql5	6.0+	V5.5
Netezza 6.0.2	netezza	6.0+	-
Oracle 10g R2	oracle	6.0+	Oracle JDBC Driver v10
Oracle 11g RAC	oracle	6.0+	Oracle JDBC Driver v11
Oracle 12c	oracle	6.0 - post GA, 6.1+	Oracle JDBC Driver v12
PostgreSQL 8.4	postgresql	6.0+	-
PostgreSQL 9.2	postgresql	6.0+	-
REST/JSON over HTTP	ws	-	-
RHEL 5.5/6 PostgreSQL config	-	6.0+	-
Salesforce.com API 22	salesforce	6.0+	-

Data Source	Translator	Supported DV Version	Driver
SAP Netweaver Gateway (OData)	sap-nw-gateway	6.1+	-
Support SAP Service Registry as a Data Source	-	6.2+	-
Sybase ASE 15	sybase	6.0+	jConnect JDBC3.0 v7
Teradata Express 12	teradata	6.0+	-
Webservices	ws	6.0+	-
XML Files	FILE	6.0+	-

NOTE

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

APPENDIX C. REVISION HISTORY

Revision 6.2.0-033	**Thu Dec 10 2015**	**David Le Sage**

Updates for 6.2.