



Red Hat JBoss Data Virtualization 6.2

インストールガイド

インストール担当チーム向けのガイド

Red Hat JBoss Data Virtualization 6.2 インストールガイド

インストール担当チーム向けのガイド

Red Hat Customer Content Services

法律上の通知

Copyright © 2015 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). 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.

概要

本ガイドでは Red Hat JBoss Data Virtualization のインストールオプションについて説明します。

目次

第1章 注意事項	4
1.1. データのバックアップ	4
1.2. 変数名: EAP_HOME	4
1.3. 変数名: MODE	4
1.4. RED HAT のドキュメントサイト	4
第2章 本書で使用される主な用語	5
2.1. データフェデレーション	5
2.2. データロール	5
2.3. MAVEN について	6
2.4. チェックサムを検証	7
2.5. 仮想データベース	7
2.6. データソース	7
2.7. TEIID DESIGNER	7
2.8. コネクタフレームワーク	8
2.9. トランスレーター	8
2.10. サービス	8
2.11. OPEN DATABASE CONNECTIVITY (ODBC)	9
2.12. データベースツールの役割	9
第3章 プラットフォームの要件	10
3.1. アーキテクチャーと必要性の評価	10
第4章 製品のダウンロード	13
4.1. RED HAT カスタマーポータル	13
4.2. JBOSS DATA VIRTUALIZATION インストーラーのダウンロード	13
4.3. ダウンロードされたファイルの検証	14
第5章 MAVEN リポジトリ	15
5.1. 提供される MAVEN リポジトリ	15
5.2. ファイルシステムリポジトリを使用するよう MAVEN を設定	15
5.3. オンラインリポジトリを使用するよう MAVEN を設定	18
5.4. 依存関係管理	21
5.5. RED HAT JBOSS DATA VIRTUALIZATION に MAVEN 依存関係を使用	21
5.6. オフラインモード	22
第6章 事前に必要なコンポーネントのインストール	24
6.1. RED HAT ENTERPRISE LINUX に OPENJDK をインストール	24
6.2. MAVEN のインストール	24
第7章 製品のインストール	27
7.1. RED HAT JBOSS DATA VIRTUALIZATION のインストール方法	27
7.2. JBOSS DATA VIRTUALIZATION のインストール: 基本インストール	27
7.3. テキストベースのインストーラーを使用した JBOSS DATA VIRTUALIZATION のインストール	29
7.4. JBOSS DATA VIRTUALIZATION 開発ツールのインストール	29
第8章 製品の開始および停止	31
8.1. JBOSS DATA VIRTUALIZATION の起動	31
8.2. JBOSS EAP 6 を管理対象ドメインとして起動	31
8.3. JBOSS DATA VIRTUALIZATION の停止	32
第9章 JBOSS DATA GRID キャッシュのインストール	33
9.1. JBOSS DATA GRID アダプターの設定	33

第10章 ODBC サポート	34
10.1. はじめに	34
10.2. RED HAT ENTERPRISE LINUX に ODBC ドライバーをインストール	34
10.3. MICROSOFT WINDOWS に ODBC ドライバーをインストール	35
10.4. SOLARIS に ODBC ドライバーをインストール	36
10.5. ODBC の設定	37
10.6. DSN を使用しない接続	39
第11章 自動インストール	41
11.1. 自動インストール	41
第12章 クラウド環境での実行	42
12.1. AMAZON AWS CLOUD インスタンスでの RED HAT JBOSS DATA VIRTUALIZATION の実行	42
12.2. GOOGLE COMPUTE インスタンスでの RED HAT JBOSS DATA VIRTUALIZATION の実行	42
第13章 OPENSIFT で RED HAT JBOSS DATA VIRTUALIZATION CARTRIDGE を実行	44
13.1. カートリッジのインストール	44
13.2. OPENSIFT でデータ仮想化 WEB インターフェースを使用	45
付録A 事前に必要なコンポーネントのインストール	49
A.1. RED HAT ENTERPRISE LINUX に OPENJDK をインストール	49
付録B 設定情報	50
B.1. サポートされるデータソースおよびトランスレーター	50
付録C 改訂履歴	53

第1章 注意事項

1.1. データのバックアップ



警告

Red Hat は、システム設定をバックアップしてから本書に掲載されている設定タスクを実行することを推奨します。

[バグを報告する](#)

1.2. 変数名: EAP_HOME

EAP_HOME は、JBoss Data Virtualization がデプロイされた Red Hat JBoss Enterprise Application Platform インストールのルートディレクトリーを示しています。

[バグを報告する](#)

1.3. 変数名: MODE

MODE は、JBoss Data Virtualization が実行されているモードがスタンドアロンまたはドメインモードであるかによって **standalone** または **domain** のいずれかになります。本書に記載されているファイルパスに **MODE** がある場合は、これらのどちらかに置き換えてください (この変数は、ディレクトリー構造のどこに製品がインストールされているかに応じて、ご自身で設定する必要があります)。

[バグを報告する](#)

1.4. RED HAT のドキュメントサイト

Red Hat の公式ドキュメントサイトは <https://access.redhat.com/site/documentation/> になります。本書を含む最新バージョンのドキュメントをご覧になれます。

[バグを報告する](#)

第2章 本書で使用される主な用語

2.1. データフェデレーション

データフェデレーションの例は、複数のデータソースを統合して1つのクエリーが1つまたは複数のデータソースから結果を返すことができる **JBoss Data Virtualization** の機能について示しています。これにより、新しいアクセス詳細を晒さずに新しいソースを簡単に統合できるようになります。



注記

この例で説明したすべてのコードは **dynamicvdb-datafederation** クイックスタートで利用できます。クイックスタートの実行手順については、**EAP_HOME/quickstarts/dynamicvdb-datafederation/README.md** を参照してください。

この例では、次の2つの異なるデータソースが使用されます。

- H2 データベースに保存されるアカウント情報
- CSV テキストファイルに保存されるマーケットデータ (株価)

DynamicPortfolio VDB は、各データソースのモデルを定義します。VDB は **JBoss Data Virtualization** にデプロイされるため、**JDBC** を使用してユーザーアプリケーションにアクセスできます。**Accounts** モデルと **MarketData** モデルからデータを結合するテストクエリーが実行されると、**JBoss Data Virtualization** はリレーショナルソースと非リレーショナルソースの両方にアクセスし、ポートフォリオ値を計算して、結果を返します。

[バグを報告する](#)

2.2. データロール

データロールの例は、データロールを使用してデータへのアクセスを制御する方法を示しています。これには、読み取り専用と読み書きのアクセスロールが含まれます。



注記

この例で説明したすべてのコードは **dynamicvdb-dataroles** クイックスタートで利用できます。クイックスタートの実行手順については、**EAP_HOME/quickstarts/dynamicvdb-dataroles/README.md** を参照してください。

この例では、VDB (**portfolio-vdb.xml**) が次の2つの異なるデータアクセスロールで定義されています。

- read-only

read-only ロールは、VDB のアクセスを読み取り専用アクセス (つまり、選択) に制限します。このロールは、ログインクレデンシャルを持つすべてのユーザーに割り当てられます ('user' という名前のユーザーとパスワード 'user' を使用できます)。

- read-write

read-write ロールにより、読み書きアクセス (つまり、挿入/更新/削除) が提供されます。このロールは、'superuser' JAAS ロールを持つユーザーにのみ割り当てられます ('portfolio' ユーザーとパスワード 'portfolio' を使用できます)。

`portfolio-vdb.xml` の read-write data-role の例は以下のとおりです。

```
<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.proeperties 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>
```

JAAS に対してユーザーとロールを定義する方法については、`EAP_HOME/quickstarts/src/dynamicvdb-dataroles/security/teiid-security-roles.properties` ファイルと `EAP_HOME/quickstarts/src/dynamicvdb-dataroles/security/teiid-security-users.properties` ファイルを参照してください。`teiid-security-users.properties` ファイルは、ユーザークレデンシャル (つまり、ログイン ID およびパスワード) を確認するために JAAS によって使用され、`teiid-security-roles.properties` ファイルはユーザー名を `<mapped-role-name>` 要素にマップします。

[バグを報告する](#)

2.3. MAVEN について

Apache Maven は、ソフトウェアプロジェクトの構築および管理を行う Java アプリケーション開発で利用される分散型構築自動化ツールです。Maven は POM (Project Object Model) と呼ばれる設定 XML ファイルを利用して、プロジェクトプロパティの定義や構築プロセスの管理を行います。POM ファイルはプロジェクトのモジュールやコンポーネントの依存関係、構築の順番、生成されるプロジェクトパッケージと出力のターゲットを定義します。この結果、プロジェクトは正しく統一された状態で構築されます。

Maven はレポジトリを使用して Java ライブラリー、プラグイン、および他のビルドアーティファクトを格納します。リポジトリはローカルまたはリモートのいずれかになります。ローカルリポジトリは、ローカルマシンにキャッシュされたりリモートリポジトリからのアーティファクトのダウンロードです。リモートリポジトリは、<http://> (HTTP サーバーにある場合) や <file://> (ファイルサーバーにある場合) などの一般的なプロトコルを使用してアクセスされる他のリポジトリです。デフォルトのリポジトリは、パブリックリモートの [Maven 2 Central Repository](#) です。

Maven の設定は、`settings.xml` ファイルを変更して実行されます。グローバルな Maven 設定は、`M2_HOME/conf/settings.xml` ファイルまたは `USER_HOME/.m2/settings.xml` ファイルのユーザーレベル設定で指定できます。

Maven の詳細については、[Welcome to Apache Maven](#) を参照してください。

Maven リポジトリの詳細については、[Apache Maven Project - Introduction to Repositories](#) を参照してください。

Maven POM ファイルの詳細については [Apache Maven Project POM Reference](#) を参照してください。



注記

Red Hat JBoss 製品は `maven 3.0.x` でビルドされました。したがって、これは、独自の SwitchYard アプリケーションをビルドするのに推奨される `maven` バージョンです。

[バグを報告する](#)

2.4. チェックサムの検証

チェックサムの検証は、ダウンロードされたファイルが破損していないことを確認するために使用されます。チェックサムの検証には、デジタルデータの任意のブロックから固定サイズのデータ (またはチェックサム) を算出するアルゴリズムが使用されます。異なるユーザーが同じアルゴリズムを使用して特定ファイルのチェックサムを算出しても、結果は同じになります。そのため、サプライヤーと同じアルゴリズムを使用してダウンロードされたファイルのチェックサムを算出する場合、チェックサムが一致すればファイルの整合性が確認されます。チェックサムが一致しない場合は、ダウンロードの処理中にファイルが破損したことになります。

[バグを報告する](#)

2.5. 仮想データベース

仮想データベース (VDB) は、複数の異なるデータソースからデータを統合するコンポーネントのコンテナであり、データが単一のデータベースにある (したがって、単一の統一 API を使用) ようにアプリケーションがデータにアクセスし、問い合わせることを可能にします。

[バグを報告する](#)

2.6. データソース

データソースはデータのレポジトリです。クエリ言語を使用すると、ユーザーはこれらのソースに格納されたデータを取得および操作できます。

[バグを報告する](#)

2.7. TEIID DESIGNER

Teiid Designer は、複数のデータソースをモデリング、分析、統合、およびテストする Eclipse ベースのグラフィカルモデリングツールです。Teiid Designer はリレーショナル、XML、および Web サービスビューを生成します。

Teiid Designer の機能は以下のとおりです。

- セマンティックな差異を解決します。
- 物理または論理レベルで仮想データ構造を作成します。
- 宣言型インターフェースを使用して、アプリケーションの互換性のためにデータを統合および集約し、ソースからターゲットの形式にデータを変換します。

[バグを報告する](#)

2.8. コネクタフレームワーク

Red Hat JBoss Data Virtualization には、VDB が依存する物理データソースへのアクセスを可能にするトランスレータとリソースアダプターのセットが含まれます。

Red Hat JBoss Data Virtualization によって直接サポートされないデータソースには、カスタムのトランスレータやリソースアダプターを開発することができます。

[バグを報告する](#)

2.9. トランスレータ

Red Hat JBoss Data Virtualization では、トランスレータはクエリーエンジンと物理データソース間の抽象レイヤーを提供します。このレイヤーはクエリーコマンドをソース固有コマンドに変換し、リソースアダプターを使用してこれらのコマンドを実行します。また、トランスレータは、物理ソースからの結果データをクエリーエンジンが必要な形式に変換します。

[バグを報告する](#)

2.10. サービス

Red Hat JBoss Data Virtualization Service は、ビジネスアプリケーションと1つまたは複数のデータソース間で機能し、実行時のビジネスアプリケーションによるアクセスに対してこれらのデータソースの統合を調整します。

Red Hat JBoss Data Virtualization サーバーは、以下のコンポーネントを管理します。

仮想データベース

仮想データベース (VDB) は、複数の物理リポジトリにあるデータの統一ビューを提供します。VDB はさまざまなデータモデルと、どのデータソースをどのように統合するかを定義する設定情報から構成されます。特に、ソースモデルは物理データソースの構造と特性を表すために使用され、ビューモデルは、アプリケーションに公開される統合データの構造と特性を表します。

アクセス層

アクセス層は、アプリケーションが JDBC、ODBC、または Web サービスを介してクエリー (リレーショナル、XML、XQuery、および手続き) を VDB に送信するために使用するインターフェースです。

クエリーエンジン

アプリケーションがアクセス層を介して VDB にクエリーを送信する場合、クエリーエンジンは、SQL 基準と VDB のソースおよびビューモデル間のマッピングによって決定される必須物理データソースに対して効率的なアクセスを提供する最適なクエリー計画を生成します。このクエリー計画に従って、オーダーが処理され、物理データソースが最も効率的にアクセスされます。

コネクタフレームワーク

トランスレーターとリソースアダプターは、クエリーエンジンと物理データソース間で透過的な接続を提供するために使用されます。トランスレーターは、クエリーをソース固有のコマンドに変換するために使用され、リソースアダプターはソースとの通信を提供します。

[バグを報告する](#)

2.11. OPEN DATABASE CONNECTIVITY (ODBC)

Open Database Connectivity (ODBC) は、データベース管理システム (DBMS) にアクセスするための標準的な C プログラミング言語のミドルウェア API です。データベースシステムやオペレーティングシステムに依存しないように設計されています。ODBC を使用して書かれたアプリケーションは、データアクセスコードに若干の変更を加えれば、クライアント側とサーバー側の両方で他のプラットフォームへのポートが可能です。

[バグを報告する](#)

2.12. データベースツールの役割

Red Hat JBoss Data Virtualization のデータベースは、ModeShape、Dashboard Builder、およびコマンド/監査ロギング機能の 3 つのアイテムにより必要とされます。Red Hat JBoss Data Virtualization とともに Red Hat JBoss Fuse Service Works を使用している場合は、BPEL および SwitchYard/jBPM 統合もこの機能を使用します。

[バグを報告する](#)

第3章 プラットフォームの要件

3.1. アーキテクチャーと必要性の評価

最低推奨サイズ

以下の最低条件を開始点としてください。これらの条件は期待される使用量に応じて調整する必要があります。

JBDS (Teiid Designer) - アプリケーションサーバーなし

- 2 GB の RAM が最低でも必要ですが、大型のモデルにはより多くの RAM が必要です。
- 最新のプロセッサ
- インストールされた製品ファイルには 500 MB のディスク領域が必要です。
- モデルプロジェクトや関連するアーティファクトには 2 GB 以上の領域が必要です。

以下の推奨サイズの目的は、サーバーの開始点 (最小サイズ) を提供することです。

以下は開始点で、推奨される最低条件になります。また、クライアントの情報が取得できないときも推奨サイズとして使用してください。

DV サーバーの最低条件は次のとおりです。

- 16 GB の JVM メモリーサイズ。
- 最新のマルチコア (デュアル以上) プロセッサまたは最新のマルチコアプロセッサを持つマルチソケットシステム
- JBoss のサーバー製品と DV コンポーネントには 20 GB 以上のディスク領域が必要です:
- インストールされた製品のファイルには 1 GB のディスクが必要です。
- ログファイルとデプロイされたアーティファクトには 5 GB 以上が必要です。
- BufferManager の maxBufferSize には 15 GB (デフォルト) が必要です。
- Modeshape (リポジトリ) が使用される場合、ファイル領域を最低でも 5 GB 増やす必要があります。
-

JVM フットプリントの最低条件を判断するには、平行性、データボリューム、および計画処理の 3 点を考慮します。

- 同時性: 最大セッション、トランスポートスレッドプール、エンジンスレッドプール/エンジン (特に最大アクティブ値) の設定、および接続プールサイズを考慮します。
- データボリューム: バッチサイズを基にデータソースから読み取られるデータの量を考慮します。デフォルトのプロセッサバッチサイズは 256 で、行ごとに約 2k バイトをターゲットとし、約 512kb のサイズでシステムを通過します。しかし、メモリー容量の多いマシンでは、バッチサイズを 512 に増加し、バッチごとにサイズを約 1mb にすることが推奨されます。
- 計画処理: クエリー計画を基に行われるデータの追加処理を考慮します。通常、これには追加のメモリーが必要です (ソートなど)。

サイズの決定に使用される仮定は次のとおりです。

- サーバーは調整されているため (スレッドプール、接続プールなど)、待機または最大のスループットなしで各クエリーは実行されます。
- 計画ではデータソースごとに1つのソースクエリーがあります (複雑なクエリーが増えると、さらに多くのメモリーが必要になります)。
- Teiid と同じ JVM で実行されている他のアプリケーションはありません (同じ JVM で他のアプリケーションが実行される場合は、追加のメモリー要件を考慮する必要があります)。
- 非トランザクションの直線的な読み取りを実行します (Teiid は、事前にバッチフェッチを行いますが、これはメモリーの要件が増えるため、バッチタイプが2倍になります)。
- デフォルトのプロセッサバッチサイズは 512 に設定されています (256 のデフォルトから変更)。これは、バッチのオーバーヘッドを低減するため、メモリー容量の多いマシンで推奨されます。

最小 JVM サイズを推定する式は (同時クエリー) * (4 * バッチバイト) + (2 * (計画ごとのソースクエリーの数 * ソースバイトの概算)) + オーバーヘッドになります。説明は次のとおりです。

- 同時クエリー
- バッチバイト: システムを通過するバッチを表します。256 バッチサイズをデフォルトで使用するとそのバイトサイズは約 512 kb になります。しかし、推奨される 512 バッチサイズを使用すると、各バッチは約 1mb になります。バッチバイトを2倍にするのは、別のバッチのプロセス中に部分的なバッチが読み出される場合にワークアイテムでバッチを格納するためです。
- 4: バッチバイトを2倍にすることと、計画操作を表します。計画操作 (ソート、結合など) を実現するために必要なバッチの追加コピー数で、2が使用されます。
- 計画ごとのソースクエリー: クエリーのデータソースの数ですが、前提を基に制限されます。
- ソースバイトの概算: $((32\text{bit} \times 5) + (4 * \text{列ごとの平均 raw バイト})) * \text{列の数}$ で、列ごとの平均 raw バイトには 10 を使用し、64 ビットのマシンを使用します。
- オーバーヘッド: AS (約 300mb) の調整、追加の Teiid オーバーヘッド (キャッシュ、計画など)、および接続プールのオーバーヘッドが含まれます。他の項目を算出するのは難しいため約 300mb のみを使用されますが、パフォーマンスを向上させるにはこれを検討する必要があります。

使用される正確な式は次のとおりです。

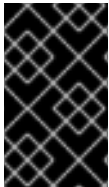
- $(\text{同時クエリー}) * (4 * \text{バッチバイト}) + (2 * \text{ソースバイト}) * \text{ソースクエリーの数} + 300\text{mb}$
- $(\text{同時クエリー}) * (4 * 1\text{mb}) + (2 * 512\text{kb}) * \text{ソースクエリーの数} + 300\text{mb}$
- $(\text{同時クエリー}) * (4\text{mb}) + (1\text{mb}) * \text{ソースクエリーの数} + 300\text{mb}$
- $\text{同時実行の数} * (5\text{mb}) * \#\text{ソースクエリー} + 300\text{mb}$

表3.1 表

同時実行 ソースクエリーの数	100	200
2	1.3 gb	2.3 gb
5	2.8 gb	5.3 gb
10	5.3 gb	10.3 gb

最大同時クエリーを基に、以下を実行して Teiid エンジン进行调整します。

- `maxActivePlans` を最大同時クエリーに設定します。
- `maxThreads` が `maxActivePlans` の 2 倍になるように設定します (トランザクションが使用される場合は 3 倍)。
- 各データソースの最大プールサイズが最大同時ソースクエリーと同じになるように設定します (最大同時 `maxThreads` が `maxActivePlans` の 5 倍クエリーが最小限になりますが、複数のソースクエリーが作成される原因となるサブクエリーがある場合、状況に応じて最大プールサイズを増加する必要があります)。
- すべての調整を行った後、サーバーのメモリー容量に余裕がある場合、`processBatchSize` および `connectorBatchSize` を増やし (例: それぞれ 512 および 1024)、データソースおよびエンジンからのスループットを増加することを検討してください。メモリー不足の場合は、JVM サイズを増やしてください。6GB 以下のメモリーを持つマシンは 512 のままにし、これよりも大きなメモリーを持つマシンにはより大きなサイズを使用してください。



重要

`maxThreads` が `maxActivePlans` の 5 倍よりも大きい場合は、調整を検討してください。`maxThreads` が `maxActivePlans` の 10 倍である場合にサーバーが処理を調整する原因となることが確認されており、最初は 2 または 3 倍にすることが推奨されます。

[バグを報告する](#)

第4章 製品のダウンロード

4.1. RED HAT カスタマーポータル

The Red Hat カスタマーポータルは <https://access.redhat.com/home> にある Web サイトです。サブスクリプションの管理や維持、Red Hat ナレッジベースへのアクセス、および Red Hat やパートナーへの連絡を1カ所で行うことができます。

[バグを報告する](#)

4.2. JBOSS DATA VIRTUALIZATION インストーラーのダウンロード

JBoss Data Virtualization のインストーラーアーカイブは、<https://access.redhat.com/> の Red Hat カスタマーポータルから入手できます。.jar アーカイブを使用してグラフィカルインストーラーを実行できます。ここでは、アーカイブをダウンロードする手順を示します。

前提条件

- Red Hat カスタマーポータル <https://access.redhat.com/> でアカウントを設定する必要があります。
- Red Hat のサブスクリプションが最新の状態であることを確認してください。
- サポートされる設定を確認し、ご使用のシステムがサポートされるようにしてください。
- インストールディレクトリーの管理権限があることを確認してください。
- Microsoft Windows サーバーでショートカットを使用する場合、JAVA_HOME と PATH が環境プロパティに設定されている必要があります。
- Java 6 または 7 の JDK が必要です。

手順4.1 JBoss Data Virtualization インストーラーのダウンロード

1. Red Hat カスタマーポータルにログインします。
 - a. ブラウザーのアドレスバーに <https://access.redhat.com/> と入力します。
 - b. ログインをクリックし、Red Hat Login と Password を入力してカスタマーポータルにアクセスします。アカウントをお持ちでない場合はアカウントを登録する必要があります。
2. JBoss Data Virtualization インストーラーをダウンロードします。
 - a. ダウンロードをクリックし、Red Hat JBoss Data Virtualization をクリックします。
 - b. Red Hat JBoss Data Virtualization [バージョン] Installer オプションの横にある Download をクリックします。
 - c. ファイルを保存します。

[バグを報告する](#)

4.3. ダウンロードされたファイルの検証

手順4.2 Red Hat Enterprise Linux でのファイルチェックサムの検証

1. ダウンロードされたファイルのチェックサム値を取得
 - a. <https://access.redhat.com/jbossnetwork/> へ移動し、必要な場合はログインします。
 - b. **Product** で製品を選択し、**Version** でバージョンを指定します。
 - c. 検証するパッケージを選択すると、**Software Details** ページに移動します。
 - d. **MD5** と **SHA-256** のチェックサム値を書き留めておきます。
2. ファイルでチェックサムツールを実行
 - a. ターミナルウィンドウで、ダウンロードされたファイルが含まれるディレクトリへ移動します。
 - b. `md5 downloaded_file` を実行します。
 - c. `shasum downloaded_file` を実行します。

以下は出力例になります。

```
[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. チェックサム値を比較

- a. `md5` および `shasum` コマンドによって返されたチェックサム値と、**Software Details** ページに表示された対応する値を比較します。
- b. 2つのチェックサム値が同じでない場合は、再度ファイルをダウンロードします。チェックサム値が異なる場合、ダウンロード中にファイルが破損したか、サーバーへアップロードされた後に変更が加えられたことを意味します。数回ダウンロードしてもチェックサムの検証に失敗する場合は、Red Hat サポートまでご連絡ください。
- c. 2つのチェックサム値が同じであれば、ダウンロードされたファイルを安全に使用できます。

結果

ダウンロードされたファイルの整合性が検証されます。



注記

Microsoft Windows にはチェックサムツールが含まれていません。MD5summerfrom (<http://www.md5summer.org/>) などのサードパーティー製の MD5 アプリケーションをダウンロードしてください。

[バグを報告する](#)

第5章 MAVEN リポジトリー

5.1. 提供される MAVEN リポジトリー

本リリースでは、アプリケーションを構築するために必要なアーティファクトが含まれるリポジトリーのセットが提供されます。適切な構築機能を提供するには、これらのリポジトリーと Maven セントラリポジトリーを使用するよう Maven を設定する必要があります。

同じ機能を提供する、交換可能な 2 つのリポジトリーのセットが提供されます。1 つ目のセットはローカルファイルシステムでダウンロードおよび保存できます。2 つ目のリポジトリーはオンラインでホストされ、リモートリポジトリーとして使用されます。インストール時に Maven の `settings.xml` ファイルの場所を指定した場合、Maven はオンラインリポジトリーを使用するよう設定されています。



重要

オンラインリモートリポジトリーのセットは技術プレビューのコンポーネントのソースです。そのため、パッチの対象ではなく、開発環境の使用のみがサポートされます。本番環境でオンラインリポジトリーのセットを使用すると、セキュリティーの脆弱性の原因となる可能性があるためユースケースではサポートされません。詳細は <https://access.redhat.com/site/maven-repository> を参照してください。

[バグを報告する](#)

5.2. ファイルシステムリポジトリーを使用するよう MAVEN を設定

概要

Red Hat JBoss EAP DV の Maven リポジトリーはオンライン上にあるため、ローカルにダウンロードし、インストールする必要はありません。しかし、JBoss EAP の Maven リポジトリーをローカルでインストールする場合は、ローカルファイルシステム上のインストール、Apache Web Server 上のインストール、および Maven リポジトリーマネージャーを用いたインストールの 3 つの方法を使用できます。この例では、ローカルのファイルシステムへ JBoss EAP 6 の Maven リポジトリーをダウンロードする手順を取り上げます。この方法は、開発環境で Maven の知識を深めることができますが、チームによる本番環境での使用は推奨されません。

手順5.1

1. <https://access.redhat.com/jbossnetwork/restricted/listSoftware.html?product=data.services.platform> からリポジトリーをダウンロードします。
2. **ダウンロード** ボタンをクリックし、リポジトリーが含まれる **.zip** ファイルをダウンロードします。
3. ローカルファイルシステム上のファイルを希望のディレクトリーで展開します。
4. 展開されていないリポジトリーのエントリーを Maven の `settings.xml` ファイルに追加します。以下のコード例には、リポジトリーがあるプロファイルと、プロファイルのアクティブなエントリーが含まれます。

```
<?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>
```

結果

Maven リポジトリがダウンロードされ、ローカルファイルシステムで展開された後、Maven の `settings.xml` ファイルに登録されます。Maven の構築を実行するときに使用できます。

トラブルシューティング

問： アプリケーションを構築またはデプロイするときにエラーが表示されます。

答：問題

プロジェクトをビルドまたはデプロイするときに、以下のエラーの1つまたは両方が発生し、失

```
[ERROR] Failed to execute goal on project PROJECT_NAME
```

```
Could not find artifact ARTIFACT_NAME
```

原因

キャッシュされたローカル Maven リポジトリに古いアーティファクトが含まれている可能性があります。

解決策

この問題を解決するには、キャッシュされたローカルディレクトリを削除します。このディレクトリは Linux の場合は `~/.m2/repository/` ディレクトリ、Windows の場合は `%SystemDrive%\Users\USERNAME\.m2\repository\` ディレクトリになります。 `mvn clean install -U` を実行します。これにより、次回構築を行うと Maven が必要なアーティファクトの適切なバージョンをダウンロードするようになります。

問： JBoss Developer Studio が古い Maven 設定を使用します。

答：問題

Maven 設定を更新しても、この設定は JBoss Developer Studio には反映されません。

原因

Maven の `settings.xml` ファイルの編集時に JBoss Developer Studio が実行されていると、この設定は JBoss Developer Studio には反映されません。

解決策

IDE の Maven 設定を更新します。メニューから **Window** → **Preferences** を選択します。 **Preferences** ウィンドウで **Maven** を展開し、 **User Settings** を選択します。 **Update Settings** ボタンをクリックし、JBoss Developer Studio の Maven ユーザー設定を更新します。

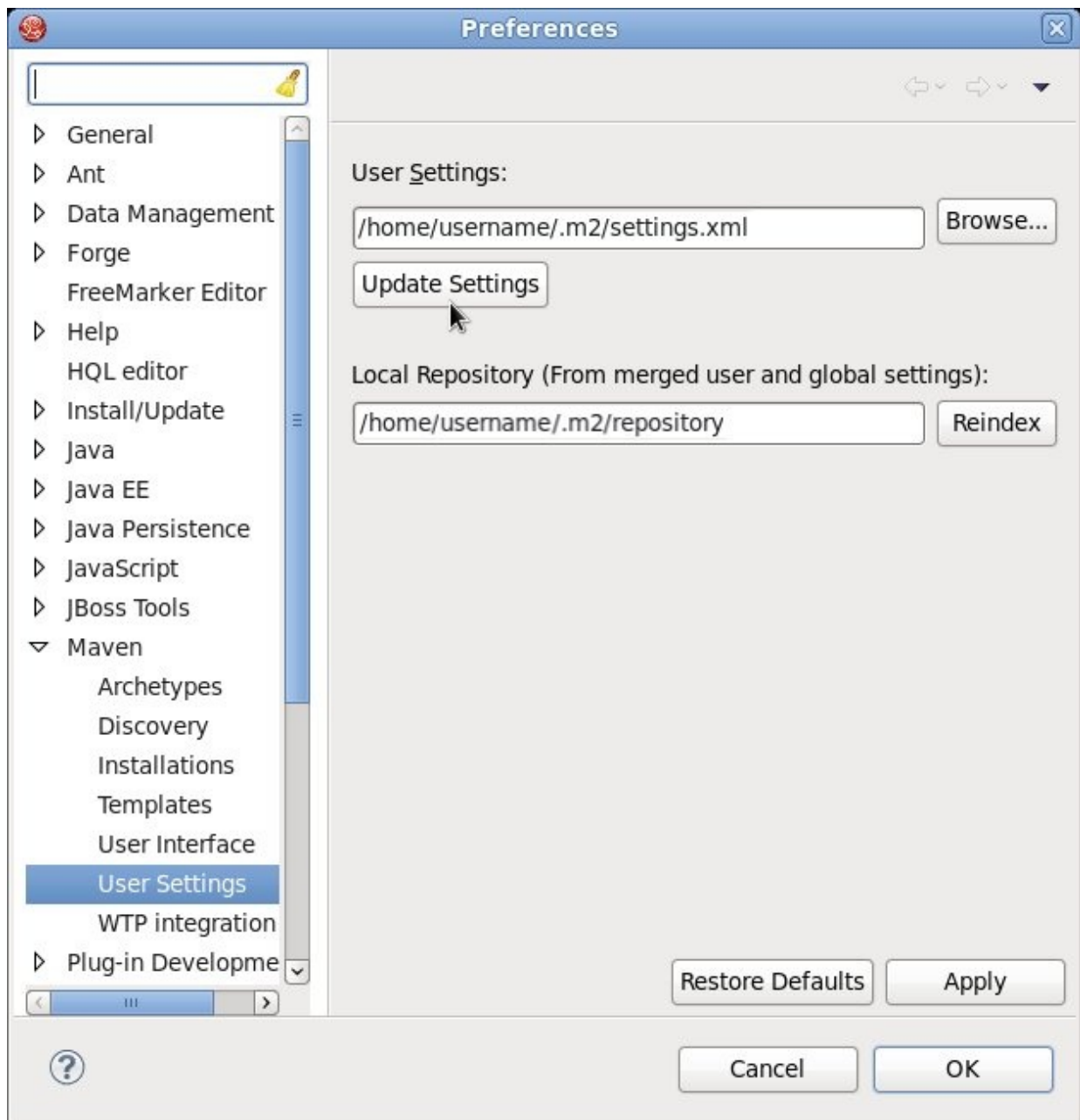


図5.1 Maven ユーザー設定の更新

[バグを報告する](#)

5.3. オンラインリポジトリを使用するよう MAVEN を設定

Red Hat JBoss Data Virtualization に必要なオンラインリポジトリは <http://maven.repository.redhat.com/techpreview/all/> にあります。

インストール中に Maven の `settings.xml` ファイルの場所を指定した場合、Maven はすでにオンラインリポジトリを使用するよう設定されています。

インストール中に Maven リポジトリを設定しなかった場合、以下の手順に従ってリポジトリを設定することができます (プロジェクトの POM ファイルを使用して設定することもできますが、推奨されません)。

手順5.2 オンラインリポジトリを使用するよう Maven を設定

1. 以下のコードサンプルのように、オンラインリポジトリのエントリを Maven の `settings.xml` ファイルに追加します。

```
<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. JBoss Developer Studio の稼働中に `settings.xml` ファイルを変更した場合は、IDE の Maven 設定を更新する必要があります。メニューで **Window** → **Preferences** と選択します。**Preferences** ウィンドウで **Maven** を展開し、**User Settings** を選択します。**Update**

Settings ボタンをクリックし、JBoss Developer Studio の Maven ユーザー設定を更新します。

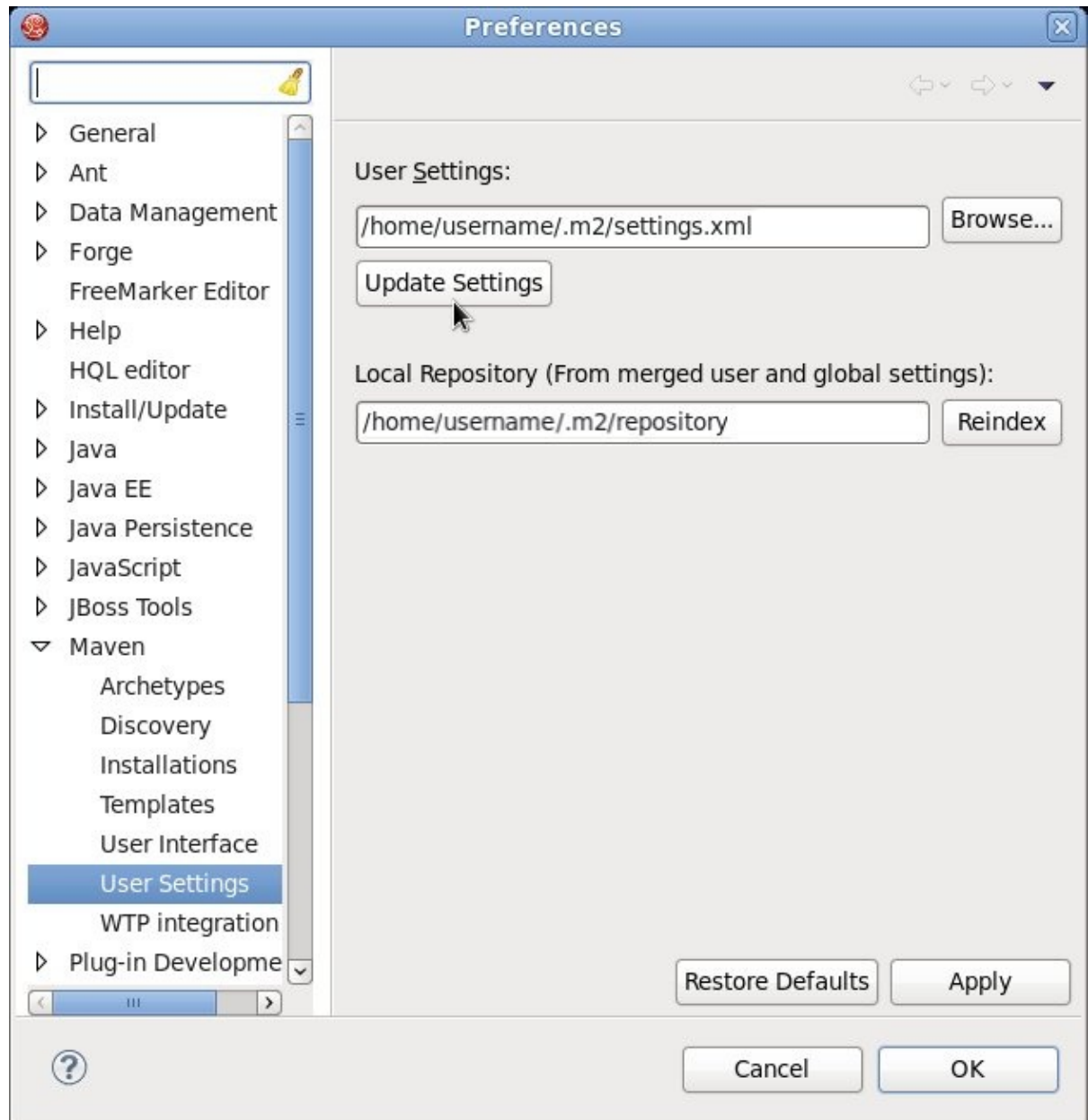


図5.2 Maven ユーザー設定の更新

結果

Red Hat JBoss 製品向けに提供されるオンラインリポジトリを使用するよう Maven が設定されます。

重要

キャッシュされたローカル Maven リポジトリに古いアーティファクトが含まれる場合は、プロジェクトをビルドまたはデプロイしたときに以下のいずれかの Maven エラーメッセージが発生することがあります。

- Missing artifact *ARTIFACT_NAME*
- [ERROR] Failed to execute goal on project *PROJECT_NAME*; Could not resolve dependencies for *PROJECT_NAME*

この問題を解決するには、キャッシュされたローカルディレクトリーを削除します。このディレクトリーは Linux の場合は `~/m2/repository/` ディレクトリー、Windows の場合は `%SystemDrive%\Users\USERNAME\.m2\repository\` ディレクトリーになります。これにより、次回の構築中に Maven が必要なアーティファクトの適切なバージョンをダウンロードするようになります。

[バグを報告する](#)

5.4. 依存関係管理

Red Hat JBoss Data Virtualization プロジェクトで適切な Maven 依存関係を使用するには、関係する Bill Of Materials (BOM) および親 POM ファイルをプロジェクトの `pom.xml` ファイルに追加する必要があります。BOM および親 POM ファイルを追加して、適切なバージョンのプラグインと提供された Maven リポジトリからの推移的依存関係が確実にプロジェクトに含まれるようにします。

Maven リポジトリは Maven セントラルリポジトリとの組み合わせのみで使用することを目的としており、他のリポジトリは必要ありません。

使用する親 POM ファイルは `org.jboss.dv.component.management:dv-parent-2.2.0.redhat-7.pom` です。

使用する BOM ファイルは `org.jboss.dv.component.management:dv-dependency-management-all-2.2.0.redhat-7.pom` です。

[バグを報告する](#)

5.5. RED HAT JBOSS DATA VIRTUALIZATION に MAVEN 依存関係を使用

Red Hat JBoss Data Virtualization に Maven 依存関係を使用するには、関係する Bill Of Materials (BOM) および親 POM ファイルをプロジェクト POM ファイルに追加する必要があります。BOM および親 POM ファイルを追加して、適切なバージョンのプラグインと推移的依存関係が確実にプロジェクトに含まれるようにします。

例を以下に示します。

```
<?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>
```

[バグを報告する](#)

5.6. オフラインモード

5.6.1. カスタムオフラインリポジトリのユースケース

プロジェクトの開発段階からデプロイメント段階に進む場合、通常はアーティファクトを必要時にインターネットからダウンロードせずに必要なアーティファクトをすべて事前インストールすると便利です。この場合、デプロイメントに必要なアーティファクトを格納するカスタムのオフラインリポジトリを作成するのが理想的です。しかし、アプリケーションのバンドルおよび機能に関連するすべての推移的依存関係を含める必要があるため、カスタムオフラインリポジトリを手動で作成するのは容易ではありません。

カスタムオフラインリポジトリの作成に最も適した方法は、Apache Karafの **features-maven-plugin** プラグインを利用してカスタムオフラインリポジトリを生成することです。

[バグを報告する](#)

5.6.2. Maven プロジェクトのリポジトリのダウンロード

Maven プロジェクトとそのランタイム依存関係をビルドするためにオフラインリポジトリを作成する必要がある場合は、[Maven 依存関係プラグイン](#)を使用できます。

たとえば、Maven プロジェクトのトップレベルディレクトリー (カレントディレクトリーに `pom.xml` ファイルがある) から以下の Maven コマンドを実行できるはずで

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

このコマンドを実行すると、`/tmp/cheese` ディレクトリーにプロジェクトをビルドするために必要な Maven 依存関係およびプラグインがダウンロードされます。

[バグを報告する](#)

5.6.3. オフラインリポジトリの生成

カスタムオフラインリポジトリを生成するには、新しいコマンドプロンプトを開き、`ProjectDir/custom-repo` ディレクトリーへ移動してから、以下の Maven コマンドを入力します。

```
mvn generate-resources
```

Maven のビルドが正常に完了すれば、カスタムオフラインリポジトリが以下の場所に生成されるはず

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

[バグを報告する](#)

第6章 事前に必要なコンポーネントのインストール

6.1. RED HAT ENTERPRISE LINUX に OPENJDK をインストール

手順6.1 Red Hat Enterprise Linux に OpenJDK をインストール

1. ベースチャンネルのサブスクリプション

RHN ベースチャンネルから OpenJDK を取得します (Red Hat Enterprise Linux のインストールはデフォルトでこのチャンネルにサブスクリプションされています)。

2. パッケージのインストール

yum ユーティリティを使用して OpenJDK をインストールします:`yum install java-1.7.0-openjdk-devel`

3. OpenJDK がシステムデフォルトであることを確認

以下の手順に従って、正しい JDK がシステムデフォルトとして設定されていることを確認します。

a. root で `java` の `alternatives` コマンドを実行します。

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

b. `/usr/lib/jvm/jre-1.7.0-openjdk/bin/java` を選択します。

c. 同様に、`javac` の `alternative` コマンドを実行します。

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

d. `/usr/lib/jvm/java-1.7.0-openjdk/bin/javac` を選択します。

[バグを報告する](#)

6.2. MAVEN のインストール

前提条件

以下のソフトウェアがインストールされている必要があります。

- 圧縮ファイルの内容を抽出するアーカイブツール。
- Open JDK。

手順6.2 Maven のインストール

1. Maven のダウンロード

a. ブラウザーのアドレスバーに <http://maven.apache.org/download.cgi> を入力します。

b. `apache-maven-[latest-version]` ZIP ファイルをダウンロードし、ハードドライブに保存します。

2. Maven のインストールおよび設定

- Red Hat Enterprise Linux の場合

- a. Maven をインストールするディレクトリーに ZIP ファイルを展開します。
- b. ターミナルで `.bash_profile` ファイルを開きます: `vi ~/.bash_profile`
- c. `M2_HOME` 環境変数をファイルに追加します。

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

- d. さらに、`M2` 環境変数をファイルに追加します。

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

- e. ファイルに `PATH` 環境変数を追加します。

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

- f. `JAVA_HOME` が JDK の場所に設定されるようにします。例は次のとおりです。

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

- g. `$JAVA_HOME/bin` が `PATH` 環境変数に含まれるようにします。

- h. ファイルを保存して、テキストエディターを終了します。

- i. `source ~/.bash_profile` コマンドを実行して変更が反映されるようにします。

- j. 以下のコマンドを実行して、ご使用のマシンに Maven が正常にインストールされたことを確認します。

```
mvn --version
```

o Microsoft Windows の場合

- a. Maven をインストールするディレクトリーに ZIP ファイルを展開します。アーカイブからサブディレクトリー `apache-maven-3.3.3` が作成されます。
- b. **Start+Pause|Break** を押します。 **System Properties** ダイアログボックスが表示されます。
- c. **Advanced** タブをクリックし、 **Environment Variables** をクリックします。
- d. **System Variables** で **Path** を選択します。
- e. **Edit** をクリックし、区切り文字にセミコロンを使用して Maven のパスを 2 つ追加します。
 - `M2_HOME` 変数を追加し、 `C:\path\to\your\Maven` へのパスを設定します。
 - `M2` 変数を追加し、値を `%M2_HOME%\bin` に設定します。
- f. **Path** 環境変数を更新または作成します。
 - `%M2%` 変数を追加し、コマンドラインから Maven を実行できるようにします。

- 変数 `%JAVA_HOME%\bin` を追加し、適切な Java インストールへのパスを設定します。
- g. **OK** をクリックして、**System Properties** ダイアログボックスを含むすべてのダイアログボックスを閉じます。
- h. **Windows** のコマンドプロンプトを開き、以下のコマンドを実行してご使用のマシンに **Maven** が正常にインストールされたことを確認します。

```
mvn --version
```

結果

マシンに **Maven** が正常にインストールおよび設定されます。

[バグを報告する](#)

第7章 製品のインストール

7.1. RED HAT JBOSS DATA VIRTUALIZATION のインストール方法

Red Hat JBoss Data Virtualization をインストールする方法は 3 つあります。

- グラフィカルモード: グラフィカルモードでは、Red Hat JBoss Data Virtualization のインストールおよび設定をステップごとに説明するグラフィカルウィザードが実行されます。インストーラーを使用して、クイックスタートや Maven リポジトリなどの追加設定を行うこともできます。
- テキストモード: テキストモードでもインストーラーを実行できます。テキストモードでは、Red Hat JBoss Data Virtualization のインストールおよび設定をステップごとに説明します。
- 自動化スクリプトモード: 自動化スクリプトを使用すると Red Hat JBoss Data Virtualization の同一インスタンスを複数インストールできます。この自動化スクリプトは、最初のインストールインスタンスの後に生成されます。

[バグを報告する](#)

7.2. JBOSS DATA VIRTUALIZATION のインストール: 基本インストール

グラフィカルインストーラーを使用すると、GUI によるステップごとの手順に従って **JBoss Data Virtualization** をマシン上にインストールできます。ここでは、インストーラーを実行するために必要な手順について説明します。

前提条件

- Red Hat JBoss EAP 6.4 インスタンスが使用できる状態である必要があります。Red Hat の Web サイトからインストーラーバージョン (ZIP ではない) をダウンロードし、デフォルトに従います。EAP のスタートガイド https://access.redhat.com/site/documentation/ja-JP/JBoss_Enterprise_Application_Platform/6.4/html/Getting_Started_Guide/index.html を参照してください。インストール後に、6.4.3 以降にパッチする必要があります。
- [カスタマーポータル](#) から Red Hat JBoss Data Virtualization jar ファイルをダウンロードしておく必要があります。
- OpenJDK やオープンソースでない Azul Zing など、JVM をインストールしておく必要があります。

手順7.1 JBoss Data Virtualization のインストール

1. ターミナルウィンドウを開き、GUI インストーラーをダウンロードした場所に移動します。
2. 以下のコマンドを実行して GUI インストーラーを起動します。

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

3. ダイアログボックスが表示された後に、使用許諾契約書が表示されます。契約の内容に同意する場合は **I accept the terms of this license agreement** をクリックし、**Next** をクリックします。
4. Red Hat JBoss EAP がサーバーにインストールされている場所を Red Hat JBoss Data Virtualization が認識するよう指定します。ファイルパスを追加し、**Next** をクリックします。

5. Teiid インストールおよび Modeshape インストールにチェックマークが付いていることを確認し、**Next** をクリックします。
6. 新しい EAP 管理者、Dashboard 管理者、Teiid データアクセスユーザー、および ModeShape ユーザーを作成するよう要求され、OData のアクセスを有効にするかどうかを問われます。これらの管理者とユーザーが作成されると、ManagementRealm へ追加され、管理コンソールやその他のアプリケーションへアクセスするために使用できます。新しいユーザー名とパスワードを適切なフィールドに入力し、**Next** をクリックします。

必ずこれらのパスワードをすべて記憶するようにしてください。これらのパスワードはシステムの異なる部分へのアクセスを可能にします。EAP アカウントおよびパスワードは EAP サーバーの管理、Dashboard のパスワードは Web インターフェースに関連する管理機能、Teiid データアクセスユーザーおよび ModeShape ユーザーは標準的なユーザーアクセスを可能にします。

ユーザー名と同じパスワードは使用できません。また、パスワードは 8 文字以上である必要があり、英字、数字、および英数字以外の文字がそれぞれ 1 文字以上含まれる必要があります。

7. Red Hat JBoss Data Virtualization は、デフォルト設定または追加設定オプションでインストールできます。ここでは、デフォルトのみを使用するため、**Perform default configuration** を選択し、デフォルトのオプションで Red Hat JBoss Data Virtualization をインストールします。**Next** をクリックします。
8. **Configure password vault** の画面が表示されます。6 文字以上のパスワードを入力します。**Next** をクリックします。



注記

デフォルトの H2 データベースは本番環境のデータベースには適していません。テストおよび評価環境のみで使用してください。

9. インストールの概要が表示されます。**Next** をクリックしてインストールを開始します。インストールには時間がかかることがあります。すべてのコンポーネントがインストールされたら **Next** をクリックします。
10. 自動スクリプトを生成したい場合は **Generate an automatic installation script** をクリックします。これを選択すると、毎回ウィザードを使用しなくても初回のインストールで指定した設定を使用して、製品を迅速に再インストールしたり、大量にデプロイしたりすることができます。



注記

Red Hat JBoss Data Virtualization 6.0 用に作成された自動インストールスクリプトは Red Hat JBoss Data Virtualization 6.1 では動作しないため注意してください。

11. **Done** をクリックし、インストーラーを終了します。

結果

Red Hat JBoss Data Virtualization が正常にインストールされ、設定されます。



注記

JBoss Data Virtualization をインストール後に別の場所へ移動すると、FileNotFound 例外が発生することがあります。これは、一部のファイルパスは JBoss EAP サーバーによってハードコードされることが原因です。



警告

データが格納される JDK とは別の JDK で作成されたキーストアを用いて vault の使用を試みると、サーバーの起動に失敗します。vault にアクセスするには同じ JDK を一貫して使用する必要があります。

[バグを報告する](#)

7.3. テキストベースのインストーラーを使用した JBOSS DATA VIRTUALIZATION のインストール

テキストベースのインストーラーを使用して Red Hat JBoss Data Virtualization をインストールできます。このモードでは、グラフィカルウィザードを使用せずにインストールの手順を実行します。ディスプレイサーバーがない場合、GUI インストーラーは自動的にテキストモードで実行されます。

前提条件

[カスタマーポータル](#)から Red Hat JBoss Data Virtualization の jar ファイルをダウンロードしておく必要があります。

手順7.2 JBoss Data Virtualization のインストール

1. ターミナルウィンドウを開き、GUI インストーラーをダウンロードした場所に移動します。
2. 以下のコマンドを入力し、インストールプロセスを開始します。

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

3. ターミナルに表示されるインストールプロンプトに従います。デフォルト設定でインストールするか、追加の設定を行います。
4. 最後の手順で、自動インストールスクリプトを生成します。このスクリプトを使用すると、ヘッドレスインストールを実行したり、複数のインスタンスで同一インストールを実行したりすることができます。

結果

Red Hat JBoss Data Virtualization が正常にインストールされ、設定されます。

[バグを報告する](#)

7.4. JBOSS DATA VIRTUALIZATION 開発ツールのインストール

前提条件

以下のソフトウェアがインストールされている必要があります。

- Red Hat JBoss Developer Studio (『Red Hat JBoss Developer Studio Installation Guide』を参照してください)。
- 圧縮ファイルの内容を抽出するアーカイブツール。
- Open JDK (『Red Hat JBoss Data Virtualization インストールガイド』を参照) またはサポートされる他の Java 仮想マシン

手順7.3 Install JBoss Developer Studio Integration Stack のインストール

1. **Red Hat JBoss Developer Studio** を起動します。
2. **Red Hat JBoss Developer Studio** でメインメニューから **Help > Install New Software...** アクションをクリックします。
3. **Available Software** ページで **Add ...** ボタンをクリックします。
4. **Add Repository** ダイアログで以下の詳細を入力します。

Name フィールドに JBDSIS (または別の一意な名前) を入力します。 **Location** フィールドに <https://devstudio.jboss.com/updates/8.0/integration-stack/> を入力します (JBDS バージョン 7 のユーザーは URL を <https://devstudio.jboss.com/updates/7.0/integration-stack/> にし、**Check for Updates** 機能を使用します)。

5. **OK** をクリックします。
6. アップデートサイトのツリービューで **JBoss Data Virtualization Development** フォルダーとすべての子を選択します。



注記

JBDSIS がすでにインストールされている場合は、次の手順に進みます。 **What is already installed** リンクをクリックすると、JBDSIS がマシンにインストールされているかを確認できます。

7. **Next** をクリックします。
8. 追加の依存関係とライセンス契約に同意し、**Finish** をクリックして完了します。

インストールに完了すると、新しい機能が完全に動作するよう **Red Hat JBoss Developer Studio** の再起動を要求されます。

結果

Teiid Designer および ModeShape Tools が正常にインストールされます。

[バグを報告する](#)

第8章 製品の開始および停止

8.1. JBOSS DATA VIRTUALIZATION の起動

前提条件

以下のソフトウェアがインストールされている必要があります。

- Red Hat JBoss Data Virtualization

手順8.1 JBoss Data Virtualization の起動

- **JBoss EAP サーバーの起動**
JBoss EAP server を起動すると JBoss Data Virtualization を実行できます。以下のように JBoss EAP server を起動します。
 - **Red Hat Enterprise Linux の場合**
ターミナルを開き、以下のコマンドを入力します。

`$ EAP_HOME/bin/standalone.sh`
 - **Microsoft Windows の場合**
ターミナルを開き、以下のコマンドを入力します。

`$ EAP_HOME\bin\standalone.bat`



注記

エラーがないことを確認するには、サーバーログ `EAP_HOME/MODE/log/server.log` を確認します。管理コンソールを開き、登録された JBoss EAP 管理ユーザーのユーザー名とパスワードでログインします。管理コンソールの使用についての詳細は『Red Hat JBoss Enterprise Application Platform 管理および設定ガイド』を参照してください。



注記

詳細な起動オプションについては『Red Hat JBoss Enterprise Application Platform 管理および設定ガイド』を参照してください。

[バグを報告する](#)

8.2. JBOSS EAP 6 を管理対象ドメインとして起動

操作の順序

ドメイン内のサーバーグループのスレーブサーバーを起動する前にドメインコントローラーを起動する必要があります。最初に、この手順をドメインコントローラーで使用した後に、関連するホストコントローラーおよびドメインに関連する他のホストに対して使用してください。

手順8.2 プラットフォームサービスを管理対象ドメインとして起動

1. **Red Hat Enterprise Linux の場合**
コマンド `EAP_HOME/bin/domain.sh` を実行します。

2. Microsoft Windows Server の場合

コマンド `EAP_HOME\bin\domain.bat` を実行します。

3. 他のパラメーターを起動スクリプトに渡す (任意)

起動スクリプトに使用できるすべてのパラメーターを表示するには、`-h` パラメーターを指定します。

結果

JBoss EAP 6 管理対象ドメインインスタンスが起動します。

[バグを報告する](#)

8.3. JBOSS DATA VIRTUALIZATION の停止

JBoss Data Virtualization を停止するには JBoss EAP server を停止する必要があります。JBoss EAP を停止する方法は、JBoss EAP を起動した方法によって異なります。ターミナルで **CTRL+C** を押すと JBoss EAP を停止できます。



注記

他の方法で JBoss EAP サーバーを停止する場合は『Red Hat JBoss Enterprise Application Platform 管理および設定ガイド』を参照してください。

結果

サーバーがシャットダウンします。シャットダウンには数分かかることに注意してください。

[バグを報告する](#)

第9章 JBOSS DATA GRID キャッシュのインストール

9.1. JBOSS DATA GRID アダプターの設定

`infinispan-cache` および `infinispan-cache-dsl` トランスレーターはサーバーの起動時に動作するよう事前設定されていません。そのため、手動で設定する必要があります。

1. `docs/teiid/datasources/infinispan/` ディレクトリーへ移動します。
2. `add-infinispan-cache-translator.cli` または `add-infinispan-cache-dsl-translator.cli` のスクリプトを実行します。

[バグを報告する](#)

第10章 ODBC サポート

10.1. はじめに

10.1.1. JBoss Data Virtualization と ODBC

JBoss Data Virtualization は、PostgreSQL データベースサーバーをエミュレートして、デプロイされた VDB へ Open Database Connectivity (ODBC) のアクセスを提供します。

ODBC 準拠のクライアントアプリケーションは、サポートされる PostgreSQL ODBC ドライバーを使用して VDB へ接続できます。サポートされる PostgreSQL ODBC ドライバーは <http://access.redhat.com> からダウンロードできます。

アプリケーションによる ODBC の使用を可能にするには、最初にアプリケーションが実行されているマシンに ODBC ドライバーをインストールし、VDB の接続プロファイルを表すデータソース名 (DSN) を作成する必要があります。



警告

現在 JBoss Data Virtualization は、ODBC のプレーンテキストパスワード認証のみをサポートしています。クライアントとサーバーが SSL を使用するよう設定されていない場合は、ネットワーク上でプレーンテキストのパスワードが送信されます。SSL を使用せずに送信するパスワードをセキュア化する必要がある場合は、クライアントからの安全なパスワード値 (暗号化やハッシュ化など) を許可するセキュリティドメインのインストールを検討してください。

[バグを報告する](#)

10.2. RED HAT ENTERPRISE LINUX に ODBC ドライバーをインストール

10.2.1. Red Hat Enterprise Linux に ODBC ドライバーをインストール

前提条件

- 管理者権限が必要です。

手順10.1 Red Hat Enterprise Linux に ODBC ドライバーをインストール

1. ドライバーのダウンロード
<https://access.redhat.com/jbossnetwork/> から適切なドライバーパッケージ (`jboss-dv-psqlodbc-[バージョン]-X.rpm`) をダウンロードしてください。
2. パッケージのインストール
`Run sudo yum localinstall jboss-dv-psqlodbc-[version]-X.rpm.`



注記

他のオペレーティングシステムのインストールパッケージは <https://access.redhat.com/jbossnetwork/> からダウンロードできます。

[バグを報告する](#)

10.2.2. ODBC 環境の設定

- 環境の設定

Run the `/opt/redhat/jboss-dv/v6/psqlodbc/etc/setenv.sh` スクリプトを実行します。

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

このスクリプトは必要なディレクトリーを `LD_LIBRARY_PATH` および `PATH` 環境変数へ追加します。このスクリプトは、ドライバーを使用するたびに実行する必要があります。

[バグを報告する](#)

10.2.3. Linux インストールの DSN の設定

- `/opt/redhat/jboss-dv/v6/psqlodbc/etc/odbc.ini` ファイルを編集し、適切なユーザー名、パスワードおよびデータベースを適用して更新します。データベース名は VDB 名です。

JBoss Data Virtualization では、ODBC はデフォルトで **35432** ポートで有効になっています。

[バグを報告する](#)

10.3. MICROSOFT WINDOWS に ODBC ドライバーをインストール

10.3.1. Microsoft Windows に ODBC ドライバーをインストール

前提条件

- 管理者権限が必要です。

手順10.2 Microsoft Windows に ODBC ドライバーをインストール

1. <https://access.redhat.com/jbossnetwork/> から適切なドライバーパッケージ (`jboss-dv-psqlodbc-[バージョン]-X.zip`) をダウンロードしてください。
2. インストールパッケージを展開します。
3. `jboss-dv-psqlodbc-[version]-X.msi` ファイルをダブルクリックしてインストーラーを起動します。
4. インストーラーウィザードが表示されます。 **Next** をクリックします。
5. 使用許諾契約書が表示されます。契約書の内容に同意する場合は **I accept the terms in the License Agreement** をクリックし、 **Next** をクリックします。

6. 表示されたデフォルト以外のディレクトリーにインストールしたい場合は、**Browse** ボタンをクリックし、ディレクトリーを選択します。**Next** をクリックします。
7. 確認画面が表示されます。選択した内容を確認し、正しければ **Next** をクリックしてインストールを開始します。
8. インストールが終了したら、インストールプロセスの終了を知らせる画面が表示されま
す。**Finish** をクリックしてウィザードを終了します。



注記

他のオペレーティングシステムのインストールパッケージは <http://access.redhat.com> からダウンロードできます。

[バグを報告する](#)

10.3.2. Windows インストールの DSN の設定

手順10.3 Windows インストールの DSN の設定

1. ODBC ドライバーの基本オプションを設定します。
2. ODBC ドライバーのデータソースオプションを設定します。
3. ODBC ドライバーのグローバルオプションを設定します。

[バグを報告する](#)

10.4. SOLARIS に ODBC ドライバーをインストール

10.4.1. Solaris に ODBC ドライバーをインストール

前提条件

- 管理者権限が必要です。

手順10.4 Solaris に ODBC ドライバーをインストール

1. ドライバーのダウンロード
<https://access.redhat.com/jbossnetwork/> から適切なドライバーパッケージ (**jboss-dv-psqlodbc-6.0.0-X.zip**) をダウンロードしてください。
2. インストールパッケージの展開
/opt ディレクトリーでインストールパッケージを展開します。
3. PATH プロパティの設定
ドライバーを展開したディレクトリーから ODBC バイナリーが使用されるよう PATH プロパティを設定します。

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

4. 環境変数の設定
既存の `odbc.ini` に `ODBCINI` 環境変数を設定します。


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



注記

Solaris ターミナルとして Bourne Shell を使用している場合、上記 2 つのエクスポートコマンドを `~/.profile` ファイルに追加すると、毎回実行する必要がなくなります。同様に、Bash を使用している場合は `~/.bash_profile` ファイルに追加します。

[バグを報告する](#)

10.4.2. Solaris インストールの DSN の設定

- `/opt/redhat/jboss-dv/v6/psqlodbc/etc/odbc.ini` ファイルを編集し、適切なユーザー名、パスワードおよびデータベースを適用して更新します。データベース名は VDB 名です。

JBoss Data Virtualization では、ODBC はデフォルトで **35432** ポートで有効になっています。

[バグを報告する](#)

10.5. ODBC の設定

10.5.1. Microsoft Windows での ODBC オプションの設定

前提条件

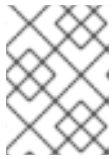
- 管理権限を使用してワークステーションにログインする必要があります。
- コントロールパネルのデータソース (ODBC) アプレットを使用して新しいデータソース名が追加されている必要があります。

設定する各データソース名は Teiid System 内の 1 つの VDB のみにアクセスできます。複数の VDB を利用できるようにするには、複数のデータソース名を設定する必要があります。

手順10.5 Microsoft Windows でのデータソース名 (DSN) の設定

1. Start メニューから **Settings - Control Panel** と選択します。
2. **Control Panel** が表示されます。 **Administrative Tools** をダブルクリックします。
3. **Data Sources (ODBC)** をダブルクリックします。
4. **ODBC Data Source Administrator** アプレットが表示されます。追加する DSN のタイプに関連するタブをクリックします。
5. **Create New Data Source** ダイアログボックスが表示されます。 **Select a driver for which you want to set up a data source** テーブルで **PostgreSQL Unicode** を選択します。
6. **Finish** をクリックします。
7. **Data Source Name** 編集ボックスで、このデータソースに割り当てる名前を入力します。

8. **Database** 編集ボックスで、このデータソースからアクセスする仮想データベースの名前を入力します。
9. **Server** 編集ボックスで、Teiid ランタイムのホスト名または IP アドレスを入力します。



注記

ファイアウォールまたは NAT アドレスを介して接続している場合は、ファイアウォールアドレスまたは NAT アドレスを入力する必要があります。

10. **Port** 編集ボックスで、システムが ODBC リクエストをリッスンするポート番号を入力します (Red Hat JBoss Data Virtualization のデフォルトでは、ODBC リクエストをポート 35432 でリッスンします)。
11. **User Name** および **Password** 編集ボックスで、Teiid ランタイムアクセスのユーザー名およびパスワードを入力します。
12. SSL モードは無効 (disabled) のままにしておきます (現在、SSL 接続はサポートされていません)。
13. データソースの詳細を **Description** フィールドに入力します。
14. **Datasource** ボタンをクリックし、オプションを設定します。Parse Statements、Recognize Unique Indexes、Maximum、Text as LongVarChar、および Bool as Char にチェックマークを入れます。MaxVarChar を 255、Max LongVarChar を 8190、Cache Size を 100、SysTable Prefixes を dd_ に設定します。

2 ページ目では、LF、Updatable Cursors、bytea as LO、default、7.4+ をクリックします。Extra Opts を 0x0 に設定します。

15. **Save** をクリックします。

Red Hat JBoss Data Virtualization が稼働している場合は、任意で **Test** をクリックすると接続を検証できます。

結果

Excel や Access などのアプリケーションを使用して VDB でデータをクエリーできるようになりました。

バグを報告する

10.5.2. Red Hat Enterprise Linux での ODBC オプションの設定

手順10.6 Red Hat Enterprise Linux での ODBC オプションの設定

1. `yum install unixODBC` コマンドを実行し、ドライバマネージャーをインストールします。
2. `odbcinst -q -d` コマンドを実行し、PostgreSQLdriver が正しくインストールされたことを確認します。
3. DSN を作成するには、`sudo vi /opt/redhat/odbc.ini` を実行して、テキストエディターで設定ファイルを開きます。



注記

このファイルを開くには、**sudo** を使用するか **root** でログインしている必要があります。

4. 以下の設定をファイルに追加します。

```
[<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. ファイルを保存して、テキストエディターを終了します。
6. 以下のコマンドを実行して DSN をテストします。

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



注記

DSN を使用せずに接続するには、以下の DSN を使用しない接続文字列を使用します。

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

[バグを報告する](#)

10.6. DSN を使用しない接続

DSN を明示的に作成せずに ODBC を使用して JBoss Data Virtualization VDB へ接続することもできます。しかし、このような場合にはアプリケーションに DSN を使用しない接続文字列が必要になります。接続文字列の例を以下に示します。

Windows の場合

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

*nix の場合

■

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

[バグを報告する](#)

第11章 自動インストール

11.1. 自動インストール

同じ設定で Red Hat JBoss 製品をインストールする必要がある場合、インストールスクリプトを使用すると時間を節約できます。事前定義された設定のインストールスクリプトを使用すると、インストールの手順を毎回行う必要がなく、1つのコマンドを実行してインストール全体を実行できます。インストールスクリプトを生成するには、インストーラー(グラフィカルまたはテキストモード)を実行し、希望の設定を指定した後、プロセスの最後の方で指示されたときにスクリプトの生成を選択します。

前提条件

- <https://access.redhat.com/jbossnetwork/> から関連するインストーラー JAR ファイルをダウンロードしておく必要があります。
- インストーラー(グラフィカルまたはテキストモード)を使用してインストールを行ったときにスクリプトを生成し、XML ファイルとして保存しておく必要があります。

手順11.1 スクリプトを用いたインストール

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

#### 結果

製品が正常にインストールされ、スクリプトの指定どおりに設定されます。

[バグを報告する](#)

## 第12章 クラウド環境での実行

### 12.1. AMAZON AWS CLOUD インスタンスでの RED HAT JBOSS DATA VIRTUALIZATION の実行

標準的なインストール手順を行った後、以下の手順に従って Red Hat Data Virtualization を AWS と使用します。

#### 手順12.1 Amazon AWS Cloud インスタンスでの Red Hat JBoss Data Virtualization の実行

1. セキュリティーグループを更新し、ポートを開きます (最低でも TCP、HTTP、および SSH ポートを開く必要があります)。
2. サーバーを起動するには、パラメーター `-Djboss.bind.address.management=0.0.0.0` および `-b 0.0.0.0 example ./standalone.sh -Djboss.bind.address.management=0.0.0.0 -b 0.0.0.0` を追加して、管理およびホストポートをバインドします。
3. Teiid Designer から AWS インスタンスにアクセスするには、JDBS の設定に移動し、**General, Network Connections SSH2** を選択します。

次に、**Key Management** タブの **Load Existing Key** を使用して Amazon によって生成されたキーを追加します。

4. サーバー接続を作成するには、**Server Configuration Overview Panel** の **Server Behavior** で **Remote System Deployment** を選択します。必ず **Server is externally managed...** にもチェックマークを入れるようにしてください。

**New Host** ボタンをクリックし、**SSH Only** オプションを選択した後、**Next** をクリックします。

Amazon パブリック IP アドレスと一致するようホスト名を設定し、接続名が同じになるようにします。

**Finish** をクリックします。

5. **Remote Systems** タブを開きます。

新しい接続を右クリックし、**connect** をクリックします。ユーザー ID を入力します (SSH キーが設定されている場合はパスワードを入力する必要はありません)。

6. **Server Configuration Overview Panel** に戻り、ホストのドロップダウンメニューに新たに作成したホストが選択されていることを確認します。
7. サーバーを起動します (起動したサーバーの状態を切り替えます)。

[バグを報告する](#)

### 12.2. GOOGLE COMPUTE インスタンスでの RED HAT JBOSS DATA VIRTUALIZATION の実行

#### 手順12.2 Google Compute インスタンスでの Red Hat JBoss Data Virtualization の実行

1. **Google Developers Console - Compute - Compute Engine - VM Instance - [インスタンスの名前] - Network** と選択し、必要なポートを開きます。
2. **Google Developers Console - Compute - Compute Engine - VM Instance - [インスタンスの名前] - SSH KEys** と選択し、パブリック SSH キーをアップロードします。
3. 管理ポート (`jboss.bind.address.management`) を外部インターフェースへバインドします (管理ポートのデフォルト値は `localhost` です)。

[バグを報告する](#)

## 第13章 OPENSIFT で RED HAT JBOSS DATA VIRTUALIZATION CARTRIDGE を実行

### 13.1. カートリッジのインストール

The Data Virtualization Cartridge は、OpenShift 環境向けの Teiid、ModeShape、および Dashboard Builder を提供します。

1. OpenShift ユーザーアカウントを作成します。
2. OpenShift Web コンソールを起動し、<https://openshift.redhat.com/app/console/applications> の Applications ページに移動します。
3. Application タブへ移動し、**Add Application...** を選択します。
4. xPaaS セクションへ移動し、JBoss Data Virtualization cartridge を選択します。
5. アプリケーションの名前 (例: jbossdv) を入力します。
6. アカウントが許可する場合は **Medium Gear** を選択します。
7. **Create Application** ボタンをクリックします。
8. Red Hat JBoss Data Virtualization Cartridge がデプロイされます。



#### 重要

今後使用するときのために、ユーザー名とパスワードの情報をコピーし、安全な場所に保管しておいてください。

9. **Continue to the application overview page** をクリックし、Data Virtualization カートリッジの概要を確認します。

状態が **Started** であることを確認します。

10. アプリケーションリンクをクリックすると、Data Virtualization の Welcome ページが表示されます。このページにはカートリッジ情報や役立つリンクが含まれています。
11. カートリッジがデプロイされた後に、[http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com](http://[MYAPP]-[MYDOMAIN].rhcloud.com) へ移動し、状態を確認します。

ユーザー、OData、および REST ロールとともにユーザーが自動的に生成されます。

12. [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/modeshape-webdav](http://[MYAPP]-[MYDOMAIN].rhcloud.com/modeshape-webdav) および [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/modeshape-rest](http://[MYAPP]-[MYDOMAIN].rhcloud.com/modeshape-rest) の ModeShape へアクセスします。

msuser と msadmin の 2 人の ModeShape ユーザーがインストールとともに生成されます。

13. [http://\[MYAPP\]-\[MYDOMAIN\].rhcloud.com/dashboard](http://[MYAPP]-[MYDOMAIN].rhcloud.com/dashboard) の Dashboard Builder へアクセスします。

Dashboard 管理者である dbadmin がインストールとともに生成されます (Teiid ユーザーは Dashboard の読み取り専用アクセスのみ許可されます)。





## 重要

パスワードを取得するため、パブリック OpenShift ssh キーをアプリケーションのメインページに追加し、ssh 経由でマシンに接続するようにしてください。Want to log in to your application? をクリックし、マシンへ接続したらターミナルを開き、`env | grep PASSWORD` を実行してパスワードを取得します。

[バグを報告する](#)

## 13.2. OPENSIFT でデータ仮想化 WEB インターフェースを使用

### 手順13.1 Box Grinder プラグインのインストール

- 最初に、MySQL データベースを OpenShift Data Virtualization インスタンスに割り当てます。ご使用のデータ仮想化アプリケーションの概要ページに移動し、**Add MySQL 5.5** リンクをクリックします (Databases セクションの下にあります)。

MySQL カートリッジがデプロイされます。完了すると、正常にデプロイされたことを伝える画面が表示されます。



## 重要

後で参照できるよう、必ずクレデンシャルの情報を保存してください。

- Tools and Support** の下にある **Add phpMyAdmin 4.0** リンクをクリックします。これにより、Web インターフェースが追加され、MySQL データベースの管理が容易になります。

これで、MySQL データベースと管理インターフェースのデプロイメントが完了します。

- phpMyAdmin インターフェースを使用して、一部のデータをデータベースにロードします。<https://myApp-myDomain.rhcloud.com/phpmyadmin> で phpMyAdmin カートリッジをインストールしたときに表示されたリンクを使用します。

カートリッジがインストールされたときに取得した root ユーザーと root パスワードを使用します。

左側のツリーパネルで、DV アプリケーション (例: jbossdv1) の名前と一致するデータベースをクリックします。**SQL** タブを選択します。SQL タブ上でこの DDL をコピーし、テキストエリアに張り付けます。

```
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);
```

最後に、管理インターフェースの右端にある **GO** ボタンをクリックします。

**PricesTable** テーブルが作成され、事前設定されます。内容をクリックするとその内容を確認できます。

4. **Data Virtualization** カートリッジを再起動します。これは、MySQL データソースを初期化するために必要になります。アプリケーション概要ページの **Restart application** アイコンをクリックします。
5. <http://www.developerforce.com/events/regular/registration.php?d=70130000000EjHb> へアクセスし、Salesforce アカウントを登録します (アカウントへログインするリンクが含まれる電子メールを受け取ります)。
6. ユーザー名とパスワードを入力して Salesforce インターフェースへログインし、**Personal Setup - Reset My Security Token** へ移動します。セキュリティトークンをリセットします。セキュリティトークンが含まれる別のメールを受け取ります。
7. 以下の例で Salesforce インスタンスへの接続を確立したら、Salesforce のユーザー名を使用します (パスワードは Salesforce パスワードとセキュリティトークンの組み合わせになります)。
8. <http://APP-NAME-DOMAIN-NAME.rhcloud.com/dv-ui> の OpenShift にデプロイされた Web インターフェースに移動します (APP-NAME はアプリケーションに付けた名前に置き換え、DOMAIN-NAME は OpenShift ドメイン名に置き換えます)。
9. **Data Library** 画面から **Create Data Services** リンクをクリックします。 **Create Data Service** ページへ移動されます。
10. **Manage Sources** ボタンをクリックします。 **Manage Data Sources** ページへ移動されます。
11. **MySQLDS** ソースをクリックします。これは設定されていますがアクティブでない状態です。ソースの **mysql5** トランスレーターを選択し、**Save Changes** ボタンをクリックします。デプロイされると、ソースはアクティブになるはずですが (緑色のチェックマークが表示されます)。
12. **Add Source** ボタンをクリックします。 **MyNewSource** というデフォルトの H2 ソースが作成されます。
13. **MyNewSource** をクリックします。

表示されたプロパティーで **Source Type** の **Name** に **SalesforceDS** を入力し、**Salesforce** ボタンをクリックします。 **Ok** ボタンをクリックすると、ソースが **Salesforce** タイプに変更され、トランスレーターが **Salesforce** に設定されます。 **Connection Properties** で **Salesforce** アカウ

- ントのユーザー名を入力します。パスワードとトークンの組み合わせをパスワードとして入力します。
14. **Save Changes** ボタンをクリックします。各ダイアログボックスに対して **Ok** をクリックします。Salesforce ソースがデプロイされます (終了するまで数分かかります)。
  15. バックリンクをクリックし、**Create Data Service** ページに戻ります。
  16. 次に、MySQL データベーステーブルにアクセスするデータサービスを作成します。サービス名に対して **MySQLService** を入力します。サービスの説明を入力します。
  17. **Service Helpers Active Sources** タブの **MySQLDS** をクリックします。
  18. **Tables** で **dv61.PricesTable** をクリックし、チェックボックスを使って両方の列を選択します。
  19. **Service View Definition** を作成するには **Create Service View** ボタンをクリックします。**Service View Definition** テキストエリアが事前設定されます。
  20. **Test Service** ボタンをクリックしてサービスをテストします。
  21. **Create Data Service** をクリックします。エントリーが許可され、サービスが作成されます。**Data Service Details** ページへリダイレクトされます。
  22. Salesforce 専用サービスの作成手順も同様になります。**Data Library** へ戻り、**Create Data Service** ボタンをクリックして新しいサービスを作成します。
  23. サービス名に **SalesforceProductService** を入力します。サービスの説明を入力します。
  24. **Service Helpers Active Sources** タブで **SalesforceDS** をクリックします。
  25. **Tables** で **Product2** をクリックし、チェックボックスを使って **Name** および **ProductCode** 列を選択します。
  26. **Service View Definition** を作成するには **Create Service View** ボタンをクリックします。**Service View Definition** テキストエリアが事前設定されます。
  27. 前回と同様にサービスをテストしてサンプルデータを確認し、**Create Service** をクリックしてサービスを作成します。
  28. マッシュアップサービスの作成手順も同様になります。**Data Library** へ戻り、**Create Data Service** ボタンをクリックして新しいサービスを作成します。
  29. サービス名に **MashupService** を入力します。サービスの説明を入力します。
  30. **Service Helpers** セクションで **Join Definition** タブをクリックします (ここで結合を定義します)。
  31. **Available Sources** ドロップダウンメニューで **MySQLDS** を選択します。**Source Tables** で **PricesTable** をクリックします。左側のボタンをクリックし、**Left (左)** テーブルとして指定します。
  32. **Available Sources** ドロップダウンメニューで **SalesforceDS** を選択します。**Source Tables** で **Product2** テーブルをクリックします。右側のボタンをクリックし、**Right (右)** テーブルとして指定します。
  33. 左側の **Prices** テーブルで **ProdCode** と **SalesPrice** 列を確認します。右側の **Product2** テーブルで **Name** 列をチェックします。

34. JoinType は Inner Join のままにしておきます。
35. 左側の Join Criteria 列で ProdCode を選択します。
36. 右側の Join Criteria 列で ProductCode を選択します。
37. Apply ボタンをクリックし、Service Definition DDL を生成します。
38. Test Service ボタンをクリックしてサンプルデータを確認します。
39. Create Service ボタンをクリックし、マッシュアップサービスを作成します。Data Service Details ページへリダイレクトされます。Data Service Details ページにはデータのサンプルが表示され、異なる接続オプションの接続手順と URL も提供されます。
40. Back to Library リンクをクリックし、Data Library に戻ります。作成した 3 つのサービスが Data Library ページに表示されます。



### 注記

各サービスに使用できるアクションは他にもあります。

- サービスの編集: Edit Data Service ページへリダイレクトします。
- サービスの複製: 選択したサービスのコピーを作成します。
- サービスのテスト: Data Service Details ページへリダイレクトします。
- サービスの削除: 選択したサービスを削除します。
- ファイルへの保存: サービスは Teiid の「動的 VDB」によってサポートされます。このアクションによって動的な VDB xml がファイルへ保存されません。

[バグを報告する](#)

## 付録A 事前に必要なコンポーネントのインストール

### A.1. RED HAT ENTERPRISE LINUX に OPENJDK をインストール

#### 手順A.1 Red Hat Enterprise Linux に OpenJDK をインストール

1. ベースチャンネルのサブスクリプション

RHN ベースチャンネルから OpenJDK を取得します (Red Hat Enterprise Linux のインストールはデフォルトでこのチャンネルにサブスクリプションされています)。

2. パッケージのインストール

yum ユーティリティを使用して OpenJDK をインストールします:`yum install java-1.7.0-openjdk-devel`

3. OpenJDK がシステムデフォルトであることを確認

以下の手順に従って、正しい JDK がシステムデフォルトとして設定されていることを確認します。

a. root で `java` の `alternatives` コマンドを実行します。

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

b. `/usr/lib/jvm/jre-1.7.0-openjdk/bin/java` を選択します。

c. 同様に、`javac` の `alternative` コマンドを実行します。

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

d. `/usr/lib/jvm/java-1.7.0-openjdk/bin/javac` を選択します。

[バグを報告する](#)

## 付録B 設定情報

### B.1. サポートされるデータソースおよびトランスレーター

以下の表は Red Hat によってサポートされるデータソースおよびトランスレーターの一覧になります。

表B.1 サポートされるデータソースおよびトランスレーター

| データソース                                    | トランスレーター             | サポートされる DV バージョン | ドライバー                 |
|-------------------------------------------|----------------------|------------------|-----------------------|
| Apache Hive 12                            | -                    | 6.0+             | -                     |
| Apache Solr                               | solr                 | 6.1+             | -                     |
| Cloudera Hadoop                           | -                    | 6.1+             | -                     |
| EDS 5.x                                   | teiid                | 6.0+             | -                     |
| ファイル - 区切り、固定長                            | ファイル                 | 6.0+             | -                     |
| 汎用データソース - JDBC ansi                      | jdbc-ansi            | 6.0+             | -                     |
| 汎用データソース - JDBC simple (postgresql84)     | jdbc-simple          | 6.0+             | -                     |
| Google Spreadsheet                        | -                    | 6.0+             | -                     |
| Greenplum 4.x                             | postgresql           | 6.0+             | -                     |
| Hortonworks Hadoop                        | -                    | 6.1+             | -                     |
| HSQL (テスト/例のみ)                            | -                    | -                | -                     |
| 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 (リモートクライアント - hotrod) | infinispan-cache-dsl | 6.2+             | -                     |

| データソース                          | トランスレーター         | サポートされる DV バージョン     | ドライバー                  |
|---------------------------------|------------------|----------------------|------------------------|
| JBoss Data Grid 6.4 (ライブラリーモード) | infinispan-cache | 6.0 - post GA、6.1+   | -                      |
| LDAP/ActiveDirectory v3         | ldap             | 6.0+                 | -                      |
| Mainframe (CICS,IMS,VSAM)       | -                | 6.0+                 | -                      |
| 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+                 | -                      |

| データソース                              | トランスレーター       | サポートされる DV バージョン | ドライバー               |
|-------------------------------------|----------------|------------------|---------------------|
| HTTP 上の REST/JSON                   | ws             | -                | -                   |
| RHEL 5.5/6 PostgreSQL config        | -              | 6.0+             | -                   |
| Salesforce.com API 22               | salesforce     | 6.0+             | -                   |
| SAP Netweaver Gateway (OData)       | sap-nw-gateway | 6.1+             | -                   |
| SAP Service Registry をデータソースとしてサポート | -              | 6.2+             | -                   |
| Sybase ASE 15                       | sybase         | 6.0+             | jConnect JDBC3.0 v7 |
| Teradata Express 12                 | teradata       | 6.0+             | -                   |
| Webservices                         | ws             | 6.0+             | -                   |
| XML ファイル                            | FILE           | 6.0+             | -                   |



**注記**

MS Excel は書き込みプロシージャがあるためサポートされます。

[バグを報告する](#)



## 付録C 改訂履歴

改訂 6.2.0-031  
6.2 用に更新

Fri Aug 14 2015

David Le Sage