



Red Hat JBoss Web Server 3.1

インストールガイド

Red Hat JBoss Web Server 3.1 のインストールおよび設定

Red Hat JBoss Web Server 3.1 インストールガイド

Red Hat JBoss Web Server 3.1 のインストールおよび設定

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installation_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, the Red Hat 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 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

本書には、Red Hat JBoss Web Server のインストールおよび設定に関連する情報が記載されています。

目次

第1章 はじめに	4
1.1. RED HAT JBOSS WEB SERVER	4
1.2. コンポーネント	4
1.3. サポート対象オペレーティングシステムおよび設定	4
1.4. インストール方法	5
1.5. JBOSS WEB SERVER のアップグレード	5
1.6. コンポーネントドキュメントバンドル	5
第2章 RED HAT ENTERPRISE LINUX への JBOSS WEB SERVER のインストール	6
2.1. 前提条件	6
2.1.1. Java Development Kit (JDK) のインストール	6
2.1.2. Red Hat Enterprise Linux パッケージの要件	7
2.2. ZIP インストール	7
2.2.1. JBoss Web Server のダウンロードおよび拡張	7
2.2.2. JBoss Web Server インストールの設定	8
2.2.3. JBoss Web Server の起動	9
2.2.4. JBoss Web Server の停止	10
2.3. RPM インストール	10
2.3.1. RPM パッケージからの JBoss Web Server のインストール	11
2.3.2. JBoss Web Server Plus グループのインストール	12
2.3.3. JBoss Web Server の起動	13
2.3.4. JBoss Web Server の停止	13
2.3.5. 起動時に開始する JBoss Web Server サービスの設定	14
2.4. SELINUX ポリシー	14
2.4.1. SELinux ポリシー情報	14
2.4.2. RPM インストール用の SELinux ポリシー	14
2.4.3. ZIP インストールの SELinux ポリシー	15
第3章 MICROSOFT WINDOWS への JBOSS WEB SERVER のインストール	17
3.1. JAVA DEVELOPMENT KIT (JDK) のインストール	17
3.2. JBOSS WEB SERVER のダウンロードおよび拡張	17
3.3. JBOSS WEB SERVER インストールの設定	18
3.4. JBOSS WEB SERVER の起動	19
3.5. JBOSS WEB SERVER の停止	20
第4章 SOLARIS への JBOSS WEB SERVER のインストール	21
4.1. JAVA DEVELOPMENT KIT (JDK) のインストール	21
4.2. JBOSS WEB SERVER のダウンロードおよび拡張	21
4.3. JBOSS WEB SERVER インストールの設定	22
4.4. JBOSS WEB SERVER の起動	23
4.5. JBOSS WEB SERVER の停止	23
第5章 JSVC を使用した TOMCAT の起動	24
5.1. JSVC を使用した TOMCAT の起動	24
5.2. JSVC を使用した TOMCAT の停止	24
5.3. JSVC パラメーター	25
第6章 JBOSS WEB SERVER の HIBERNATE	26
第7章 JBOSS OPERATIONS NETWORK(ON)を使用した JBOSS WEB SERVER の監視	28
7.1. JBOSS ON の WEB サーバープラグインパックのダウンロード	28
7.2. JBOSS ON MONITORING 用の TOMCAT の設定	28
7.2.1. RPM から Tomcat Installed の JBoss ON Monitoring の設定	30

7.2.2. Windows サービスとしてインストールされた Tomcat の JBoss ON Monitoring の設定	31
第8章 RED HAT JBOSS WEB SERVER 3 でのパスワード VAULT の使用	33
8.1. RED HAT JBOSS WEB SERVER 3 でのパスワード VAULT の使用	33
8.1.1. JBoss Web Server パスワード vault のインストール	33
8.1.1.1. RPM からの Red Hat Enterprise Linux への JBoss Web Server パスワード vault のインストール	33
8.1.1.2. .zip アーカイブからの Vault ファイルのダウンロードおよび拡張	33
8.1.2. Java キーストアの作成	34
8.1.3. JWS_HOME ディレクトリー外での tomcat-vault vault.properties ファイルの保存	34
8.1.4. パスワード vault の初期化	34
8.1.4.1. Apache Tomcat の Vault の対話的な初期化	34
パスワード vault を使用するように Tomcat を設定する	36
8.1.4.2. Apache Tomcat の Vault の非対話的な初期化 (サイレント設定)	36
8.1.5. パスワード Vault への Sensitive 文字列の格納	36
8.1.6. Tomcat 設定で保存された機密文字列の使用	37
第9章 WEBSOCKET を使用した JWS クライアントサーバー通信の設定	38
9.1. WEBSOCKET について	38
9.2. TOMCAT での WEBSOCKET の実装	38
9.2.1. 書き込みタイムアウトの設定	38
9.2.2. 着信バイナリーメッセージの設定	38
9.2.3. 着信テキストメッセージの設定	39
9.2.4. 追加のプログラムによるデプロイメントの設定	39
9.2.5. 非同期書き込みのコールバックの設定	39
9.2.6. IO 操作のタイムアウトの設定	39
付録A JAVA IPV4/IPV6 プロパティー	41
Java プロパティーの設定	41
Tomcat バインディングの設定	41

第1章 はじめに

1.1. RED HAT JBOSS WEB SERVER

Red Hat JBoss Web Server は、Java Web アプリケーションをホストするための完全に統合され認定されたコンポーネントセットです。これは、Web サーバー (Apache HTTP Server)、アプリケーションサーバー (Apache Tomcat Servlet コンテナ)、ロードバランサー (mod_jk および mod_cluster)、および Tomcat ネイティブライブラリーで構成されます。

このインストールガイドには、サポートされるオペレーティングシステム上で JBoss Web Server から Tomcat サーバーをインストール、マイナーアップグレード、および基本設定する手順が含まれています。Apache HTTP Server のインストールおよび設定手順は、[JBoss Core Services のドキュメント](#)を参照してください。

1.2. コンポーネント

JBoss Web Server は以下のコンポーネントで構成されています。

- **Apache Tomcat** は、Java Servlet Specification に準拠するサーブレットコンテナです。JBoss Web Server には、Apache Tomcat 7 と Apache Tomcat 8 が含まれています。
- **Apache Tomcat Native Library** は Tomcat ライブラリーで、Tomcat のスケーラビリティ、パフォーマンス、ネイティブサーバー技術との統合が改善されます。
- **Apache Tomcat Connectors (mod_jk, mod_cluster)** は、Apache HTTP Server と Apache Tomcat 間のコネクタです。デフォルトは mod_cluster であることに注意してください。これは JBoss ネイティブロードバランサーであり、mod_jk よりも効率的で信頼性があり、拡張可能であるためです。JBoss Web Server 3.1 より、mod_jk および mod_cluster コネクタは JBoss Core Services の一部として提供されます。これら 2 つのモジュールのインストールと設定の詳細は、『[HTTP Connectors and Load Balancing Guide](#)』を参照してください。
- **Apache HTTP Server** は、[Apache Software Foundation](#) により開発されたオープンソースの Web サーバーです。この実装は、現在の HTTP 標準に準拠します。JBoss Web Server 3.1 では、Apache HTTP Server は JBoss Core Services の一部として提供されます。http サーバーのインストールおよび設定に関する詳細は、[JBoss Core Services Documentation](#) を参照してください。
- **Hibernate** はオブジェクト関係マッピングフレームワークです。JBoss Web Server に含まれる Hibernate には、JPA 1.0 API を使用する Hibernate Core、Hibernate Annotations、および Hibernate EntityManager が含まれます。

JBoss Web Server 3.1 に含まれるコンポーネントバージョンの詳細な一覧は、<https://access.redhat.com/articles/111723> を参照してください。



注記

Tomcat クラスタリングは JBoss Web Server 3.1 から削除されました。Java アプリケーションのクラスタリングまたはセッションレプリケーションのサポートが必要な場合、Red Hat は Red Hat JBoss Enterprise Application Platform (JBoss EAP) の使用を推奨します。

1.3. サポート対象オペレーティングシステムおよび設定

JBoss Web Server のサポートされるオペレーティングシステムおよび設定に関する詳細は、<https://access.redhat.com/articles/1377603> を参照してください。

1.4. インストール方法

各プラットフォームで利用可能な ZIP インストールファイルを使用して、サポートされる Red Hat Enterprise Linux、Microsoft Windows、および Solaris システムに JBoss Web Server をインストールできます。RPM パッケージを使用して、サポートされる Red Hat Enterprise Linux システムに JBoss Web Server をインストールすることもできます。

ZIP インストールの場合、JBoss Web Server インストールのコア部分を形成する ZIP ファイルに含まれるコンポーネントの概要を以下に示します。

- **jws-application-servers-3.1.0-<platform>-<architecture>.zip**
 - Tomcat 7
 - Tomcat 8
 - プラットフォーム固有のユーティリティ

1.5. JBOSS WEB SERVER のアップグレード

JBoss Web Server のアップグレードはサポートされていません。以前のバージョンから JBoss Web Server 3.1 にアップグレードする場合は、新規インストールが推奨されます。



重要

RPM を使用して JBoss Web Server 1.x または 2.x をインストールする場合は、JBoss Web Server 1.x または 2.x を削除して、同じマシンに JBoss Web Server 3.1 をインストールすることはできません。古いパッケージの多くの設定ファイルはシステム上に残り、新しい RPM パッケージとの互換性の問題を作成します。同じ理由で、古い RPM を手動で削除し、新しい RPM のインストールもサポートされていません。

ZIP を使用して JBoss Web Server 1.x または 2.x がインストールされているシステムでは、以前のバージョンを削除して、同じシステムに JBoss Web Server 3.1 をインストールすることができます。

1.6. コンポーネントドキュメントバンドル

JBoss Web Server には、各コンポーネントの元のベンダードキュメントが含まれる追加のドキュメントバンドルが含まれています。このドキュメントバンドル **jws-docs-3.1.0.zip** は Red Hat カスタマーポータルから入手できます。これには、以下の追加のドキュメントが含まれています。

- mod_cluster
- openssl
- tomcat7
- tomcat8
- tomcat-native

第2章 RED HAT ENTERPRISE LINUX への JBOSS WEB SERVER のインストール

JBoss Web Server は、以下のいずれかの方法を使用して Red Hat Enterprise Linux にインストールできます。

- [ZIP ファイル](#)
- [RPM パッケージ](#)

選択した方法に関係なく、最初に [サポートされる Java Development Kit\(JDK\)をインストール](#) する必要があります。

2.1. 前提条件

2.1.1. Java Development Kit (JDK) のインストール

JBoss Web Server をインストールする前に、サポートされる Java Development Kit (JDK) を最初にインストールする必要があります。

1. Red Hat Enterprise Linux システムを適切なチャンネルにサブスクライブします。

- **OpenJDK:**
 - `rhel-7-server-rpms`
- **Oracle:**
 - `rhel-7-server-thirdparty-oracle-java-rpms`
- **IBM:**
 - `rhel-7-server-supplementary-rpms`



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。

+ `.root` ユーザーとしてコマンドを実行し、1.7 または 1.8 JDK: をインストールします。JDK 1.7 の場合 :

+

```
# yum install java-1.7.0-<VENDOR>-devel
```

+ `<VENDOR& gt;` を `oracle`、`ibm`、または `openjdk` に置き換えます。JDK 1.8 の場合 :

+

```
# yum install java-1.8.0-<VENDOR>-devel
```

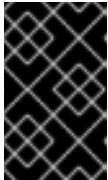
+ `<VENDOR>` を `oracle`、`ibm`、または `openjdk` に置き換えます。

1. root ユーザーとして以下のコマンドを実行し、正しい JDK が使用されていることを確認します。

```
# alternatives --config java
```

```
# alternatives --config javac
```

これらのコマンドは、利用可能な JDK バージョンと、プラス (+) 記号の付いた選択したバージョンの一覧を返します。選択した JDK が必要な JDK ではない場合は、シェルプロンプトで指示されたように、必要な JDK に変更します。



重要

java コマンドおよび **javac** コマンドを使用するすべてのソフトウェアは、**alternatives** によって設定された JDK を使用します。Java の代替の変更は、他のソフトウェアの実行に影響を及ぼす可能性があります。

2.1.2. Red Hat Enterprise Linux パッケージの要件

Red Hat Enterprise Linux に JBoss Web Server をインストールする前に、以下の前提条件を満たしていることを確認してください。

- [サポートされる JDK がインストールされている](#)。
- **tomcat-native** パッケージをインストールする前に、**tomcatjss** パッケージを削除する必要があります。**tomcatjss** パッケージは、OpenSSL セキュリティモデルではなく、基礎となる NSS セキュリティモデルを使用します。

tomcatjss パッケージの削除

1. root ユーザーとして以下のコマンドを実行し、**tomcatjss** を削除します。

```
# yum remove tomcatjss
```

2.2. ZIP インストール

JBoss Web Server をインストールする前に、すべての [前提条件を満たし](#) ていることを確認してください。

2.2.1. JBoss Web Server のダウンロードおよび拡張

JBoss Web Server をインストールするには、インストール ZIP ファイルをダウンロードし、展開します。

JBoss Web Server のダウンロード

1. ブラウザーを開き、[Red Hat カスタマーポータル](#) にログインします。
2. **ダウンロード** をクリックします。
3. **製品のダウンロード** リストの **Red Hat JBoss Web Server** をクリックします。
4. **Version** ドロップダウンメニューから正しい JBoss Web Server バージョンを選択します。

5. 以下の各ファイルの **Download** をクリックし、システムのプラットフォームおよびアーキテクチャが正しく選択されていることを確認します。
 - Red Hat JBoss Web Server 3.1 Application Server([jws-application-servers-3.1.0-
<platform> - <architecture& gt; .zip](#))

JBoss Web Server の抽出

1. ダウンロードした ZIP ファイルをインストールディレクトリーに展開します。
ZIP アーカイブを展開して作成したディレクトリーは、JBoss Web Server の最上位ディレクトリーになります。これは **JWS_HOME** と呼ばれます。

2.2.2. JBoss Web Server インストールの設定

JBoss Web Server を実行する前に一部の設定が必要になります。本セクションには、以下の設定手順が含まれています。

- [JAVA_HOME 環境変数の設定](#)
- 簡単でセキュアなユーザー管理のための tomcat ユーザーの作成: [Tomcat ユーザーの作成](#)
- [Tomcat の log4j ロギングの有効化](#)

JAVA_HOME 環境変数の設定

JBoss Web Server を実行する前に、Tomcat の **JAVA_HOME** 環境変数を設定する必要があります。

Tomcat の **bin** ディレクトリー（**JWS_HOME/tomcat7/bin** または **JWS_HOME/tomcat8/bin**）で、**setenv.sh** という名前のファイルを作成し、**JAVA_HOME** パス定義を挿入します。

例： **JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64**

Tomcat ユーザーの作成

以下の手順に従って、**tomcat** ユーザーと親グループを作成します。

1. root ユーザーとしてシェルプロンプトで、**JWS_HOME** ディレクトリーに移動します。
2. 以下のコマンドを実行して **tomcat** ユーザーグループを作成します。

```
# groupadd -g 53 -r tomcat
```

3. 以下のコマンドを実行して、**tomcat** ユーザーグループに **tomcat** ユーザーを作成します。

```
# useradd -c "Tomcat" -u 53 -g tomcat -s /bin/sh -r tomcat
```

4. **tomcat** ユーザーグループと **tomcat** ユーザーが正しく作成されていることを確認します。

```
$ grep tomcat /etc/group
tomcat:x:53

$ grep tomcat /etc/passwd
tomcat:x:53:53:Tomcat:/home/tomcat:/bin/sh
```

5. **JWS_HOME** から次のコマンドを実行して、Tomcat ディレクトリーの所有権をユーザーに割り当てて、**tomcat** ユーザーが Tomcat サービスを実行できるようにします。

```
# chown -R tomcat:tomcat tomcat<VERSION>
```

< ;VERSION> をそれぞれの Tomcat バージョン番号（7 または 8）に置き換えます。

ls -l を使用して、**tomcat** ユーザーがディレクトリーの所有者であることを確認できます。

6. **tomcat** ユーザーに、すべての親ディレクトリーへの実行権限があることを確認します。以下に例を示します。

```
# chmod -R u+X tomcat<VERSION>
```

Tomcat の Apache Log4j ロギングの有効化

Apache Log4j でのロギングを有効にするには、以下を実行します。

1. **JWS_HOME/extras/** ディレクトリーに移動します。

```
# cd JWS_HOME/extras/
```

2. **log4j-eap6.jar** および **log4j.properties** を **JWS_HOME/extras/** から **JWS_HOME/lib** にコピーします。

```
# cp log4j.properties log4j-eap6.jar ../tomcat<VERSION>/lib/
```

< ;VERSION> を Tomcat バージョン番号（7 または 8）に置き換えます。

3. **JWS_HOME/tomcat<VERSION>/extras/** ディレクトリー に移動します。

```
# cd ../tomcat<VERSION>/extras/
```

4. **tomcat-juli-adapters.jar** を **JWS_HOME/tomcat<VERSION>/extras** から **JWS_HOME/tomcat<VERSION>/lib** に コピーします。

```
# cp tomcat-juli-adapters.jar ../lib/
```

5. **JWS_HOME/tomcat<VERSION>/bin/tomcat-juli.jar** を **JWS_HOME/tomcat<VERSION>/extras/tomcat-juli.jar** に置き換えます。

```
# cp tomcat-juli.jar ../bin/
```

2.2.3. JBoss Web Server の起動

JBoss Web Server を起動するには、以下を開始する必要があります。

- Tomcat（7 または 8）

Tomcat を起動する前に、以下の前提条件を満たしていることを確認してください。

- Tomcat ユーザーが作成されている。
- JAVA_HOME が正しく設定されている。

Tomcat の起動

1. 各 Tomcat バージョン（7 または 8）で、**tomcat** ユーザーとして以下のコマンドを実行します。

```
$ sh JWS_HOME/tomcat<VERSION>/bin/startup.sh
```



重要

Tomcat を起動する方法は複数ありますが、**startup.sh** スクリプトを使用することが推奨されます。Jsvc を使用して Tomcat をサービスとして起動するには、Jsvc の [章](#) を参照してください。

2.2.4. JBoss Web Server の停止

JBoss Web Server を停止するには、以下を停止する必要があります。

- Tomcat（7 または 8）

Tomcat の停止

1. Tomcat を停止するには、それぞれの Tomcat バージョン（7 または 8）で root ユーザーとして以下のコマンドを実行します。

```
# sh JWS_HOME/tomcat<VERSION>/bin/shutdown.sh
```

2.3. RPM インストール

RPM パッケージから JBoss Web Server をインストールすると、Tomcat がサービスとしてインストールされ、そのリソースを絶対パスにインストールします。RPM インストールオプションは、Red Hat Enterprise Linux 7 でのみ利用できます。

JBoss Web Server の RPM インストールパッケージは、Red Hat Subscription Management から入手できます。



注記

Red Hat Satellite を使用して JBoss Web Server のインストールを管理するユーザーの場合：Red Hat Satellite 6 は JBoss Web Server 3.1 インストールの管理に推奨されますが、Satellite 5 ユーザー向けに以下の Red Hat Network(RHN)チャンネルも提供されています。

- Red Hat Enterprise Linux 7 の場合:
 - jws-3-x86_64-server-7-rpm

Red Hat Satellite 6 ユーザーは、[Red Hat コンテンツ配信ネットワーク\(CDN\)リポジトリ](#)を使用できます。



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。



警告

RPM パッケージから JBoss Web Server をインストールすると、Java ライブラリーファイルが他のアプリケーションと共有されます。RPM パッケージを使用して JBoss Web Server 3 と JBoss EAP 6 の両方を同じマシンにインストールすると、ライブラリーバージョンの競合が発生します。この問題を回避するには、RPM インストール手法を使用して JBoss Web Server 3 または JBoss EAP 6 をインストールし、ZIP インストール方式を使用してもう1つをインストールできます。

2.3.1. RPM パッケージからの JBoss Web Server のインストール

前提条件

- [Java Development Kit \(JDK\)](#) をインストールします
- [tomcatjss](#) パッケージが削除されていることを確認します

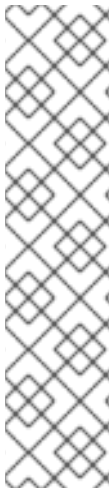
RPM パッケージをダウンロードしてインストールする前に、システムを Red Hat Subscription Management に登録し、該当する Content Delivery Network (CDN) リポジトリーをサブスクライブする必要があります。

Red Hat Enterprise Linux の登録に関する情報は、「[The Subscription Manager for Red Hat Enterprise Linux 7](#)」を参照してください。

Red Hat Enterprise Linux へのサブスクリプションの割り当て (必要な場合)

システムに JBoss Web Server を提供するサブスクリプションが割り当てられていない場合は、以下を行います。

1. [Red Hat Subscription Manager](#) にログインします。
2. **Systems** タブをクリックします。
3. システムの **Name** をクリックして、サブスクリプションを追加します。
4. **Details** タブから **Subscriptions** タブに移動し、**Attach Subscriptions** をクリックします。
5. アタッチするサブスクリプションの横にあるチェックボックスを選択してから、**Attach Subscriptions** をクリックします。



注記

サブスクリプションが必要な CDN リポジトリを提供していることを確認するには、以下を行います。

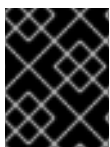
1. <https://access.redhat.com/management/subscriptions> にログインします。
2. **Subscription Name** をクリックします。
3. **Products Provided** には以下が必要です。
 - JBoss Enterprise Web Server
 - Red Hat JBoss Core Services

YUM を使用した RPM パッケージからの JBoss Web Server のインストール

1. コマンドラインで、**subscription-manager** を使用して、お使いのオペレーティングシステムバージョンの JBoss Web Server CDN リポジトリにサブスクライブします。

```
# subscription-manager repos --enable <repository>
```

- Red Hat Enterprise Linux 7 の場合:
 - jws-3-for-rhel-7-server-rpms
 - jb-coreservices-1-for-rhel-7-server-rpms



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。

1. root ユーザーとして以下のコマンドを実行し、JBoss Web Server をインストールします。

```
# yum groupinstall jws3
```



注記

- 推奨されませんが、group install を使用する代わりに、各パッケージと依存関係を個別にインストールできます。
- JBoss Web Server のインストールには、上記の Red Hat JBoss Core Services リポジトリが必要です。

2.3.2. JBoss Web Server Plus グループのインストール

JBoss Web Server Plus グループには、主に Hibernate とその依存関係の追加に関する追加のパッケージが含まれます。

JBoss Web Server Plus グループパッケージをインストールするには、root で以下のコマンドを実行します。

```
# yum groupinstall jws3plus
```


2.3.3. JBoss Web Server の起動

JBoss Web Server を起動するには、以下を開始する必要があります。

- Tomcat (7 または 8)

Tomcat の起動

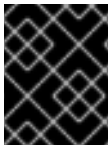
1. root ユーザーとしてシェルプロンプトで、Tomcat サービスを起動します。<code>VERSION</code>を必要な Tomcat バージョン (7 または 8) に置き換えます。
 - a. Red Hat Enterprise Linux 7 の場合:

```
# systemctl start tomcat<VERSION>.service
```

これは、RPM インストール用に Tomcat を起動するのにサポートされる唯一の方法です。

1. Tomcat が稼働していることを確認するには、service **status** コマンドの出力を確認する必要があります。これは、任意のユーザーとして実行できます。
 - a. Red Hat Enterprise Linux 7 の場合:

```
# systemctl status tomcat<VERSION>.service
```



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。

2.3.4. JBoss Web Server の停止

JBoss Web Server を停止するには、Tomcat サービスを停止します。

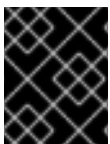
Tomcat の停止

1. root ユーザーとしてシェルプロンプトで、Tomcat サービスを停止します。<code>VERSION</code>を必要な Tomcat バージョン (7 または 8) に置き換えます。
 - a. Red Hat Enterprise Linux 7 の場合:

```
# systemctl stop tomcat<VERSION>.service
```

2. Tomcat が実行しなくなったことを確認するには、service **status** コマンドの出力を確認する必要があります。これは、任意のユーザーとして実行できます。
 - a. Red Hat Enterprise Linux 7 の場合:

```
# systemctl status tomcat<VERSION>.service
```



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。

2.3.5. 起動時に開始する JBoss Web Server サービスの設定

JBoss Web Server は起動時に起動するように設定できます。

以下のコマンドを使用して、JBoss Web Server サービスが起動時に開始できるようにします。<VERSION>を必要な Tomcat バージョン（7 または 8）に置き換えます。

- Red Hat Enterprise Linux 7 の場合:

```
# systemctl enable tomcat<VERSION>.service
```



重要

Red Hat Enterprise Linux 6 はサポートされなくなり、その後ドキュメントから削除されました。

2.4. SELINUX ポリシー

2.4.1. SELinux ポリシー情報

以下の表には、tomcat <version> -selinux パッケージで提供される SELinux ポリシーに関する情報をまとめています。

表2.1 RPM およびデフォルトの SELinux ポリシー

名前	ポート情報	ポリシー情報
tomcat<version>	http_port_t の 4 ポート (TCP ポート 8080 、 8005 、 8009 、および 8443) の 4 つのポートにより、tomcat プロセスがそのプロセスが使用できるようになります。	Tomcat <VERSION> ポリシーがインストールされ、Tomcat の実行時にプロセスに適した SELinux ドメインを設定します。また、適切なコンテキストを設定して、tomcat が /var/lib/tomcat <VERSION> 、 /var/log/tomcat <VERSION> 、 /var/cache/tomcat <VERSION> 、および /var/run/tomcat <VERSION> .pid に書き込みできるようにします。

SELinux およびその他の Red Hat Enterprise Linux セキュリティ情報の使用に関する詳細は『Red Hat Enterprise Linux Security Guide』を参照してください。

2.4.2. RPM インストール用の SELinux ポリシー

各 Tomcat の SELinux ポリシーは、独自の Tomcat サブパッケージから提供されます（**tomcat7-selinux** および **tomcat8-selinux**）。これらのパッケージは JWS チャネルで利用できます。

- Tomcat 7 で SELinux ポリシーを有効にするには、**tomcat7-selinux** パッケージをインストールします。
- Tomcat 8 で SELinux ポリシーを有効にするには、**tomcat8-selinux** パッケージをインストールします。

2.4.3. ZIP インストールの SELinux ポリシー

本リリースでは、SELinux ポリシーは ZIP パッケージで提供されます。SELinux セキュリティーモデルはカーネルにより適用され、アプリケーションがファイルシステムの場所やポートなどのリソースに限定的なアクセスを行えるようにします。これにより、誤ったプロセス (危険にさらされるか、設定されていないかのいずれか) が制限され、場合によっては実行できないようにすることが可能になります。**.postinstall.selinux** ファイルは各 **tomcat** フォルダに含まれます。必要に応じて、**.postinstall.selinux** スクリプトを実行できます。

ZIP を使用して SELinux ポリシーをインストールするには、以下を実行します。

1. **selinux-policy-devel** パッケージをインストールします。

```
yum install -y selinux-policy-devel
```

2. **.postinstall.selinux** スクリプトを実行します。

```
cd <JWS_home>/tomcat<version>  
sh .postinstall.selinux
```

ここで、**tomcat<version>** は **tomcat7** または **tomcat8** を参照します。

3. SELinux モジュールを作成してインストールします。

```
cd selinux  
make -f /usr/share/selinux/devel/Makefile  
semodule -i tomcat<version>.pp
```

4. JBoss Web Server の SELinux コンテキストを適用します。

```
restorecon -r <JWS_home>/tomcat<version>/
```

5. JBoss Web Server の必要なポートへのアクセスパーミッションを追加します。

```
semanage port -a -t http_port_t -p tcp 8005  
semanage port -a -t http_port_t -p tcp 8080  
semanage port -a -t http_port_t -p tcp 8009  
semanage port -a -t http_port_t -p tcp 8443
```

6. Tomcat サービスを起動します。

```
<JWS_home>/bin/startup.sh
```

7. **tomcat_<version>__t** が予想される実行中のプロセスのコンテキストを確認します。

```
ps -eZ | grep tomcat | head -n1
```

8. Tomcat ディレクトリーのコンテキストを確認するには、以下のように設定します。

```
ls -lZ <JWS_home>/tomcat<version>/logs/
```



注記

デフォルトでは、提供される selinux ポリシーがアクティブではなく、Tomcat プロセスが **unconfined_java_t** ドメインで実行されます。このドメインはプロセスに制限されず、提供される selinux ポリシーを有効にしない場合は、以下のセキュリティ対策を実行することが推奨されます。

- **tomcat** ユーザーのファイルアクセスを JBoss Web Server ランタイムに必要なファイルおよびディレクトリーのみに制限します。
- **root** ユーザーで Tomcat を実行しないでください。

第3章 MICROSOFT WINDOWS への JBOSS WEB SERVER のインストール

3.1. JAVA DEVELOPMENT KIT (JDK) のインストール

JBoss Web Server を Microsoft Windows にインストールする前に、最初にサポートされる Java Development Kit (JDK) をインストールする必要があります。

サポートされる構成の一覧は、Red Hat カスタマーポータルの記事「[JBoss Web Server 3 Supported Configurations](#)」を参照してください。



注記

IBM JDK のインストール方法は、<https://www.ibm.com/developerworks/java/jdk/> を参照してください。

Oracle Java Development Kit をインストールするには、以下を実行します。

1. オペレーティングシステムおよびアーキテクチャー用の Oracle JDK 1.7 または 1.8 をダウンロードします。JDK インストールファイルは、Oracle の Web サイト <http://www.oracle.com/technetwork/java/javase/downloads/index.html> からダウンロードできます。
2. ダウンロードしたファイルをダブルクリックしてインストールを開始します。
3. インストールウィンドウで指示どおりに続行します。

3.2. JBOSS WEB SERVER のダウンロードおよび拡張

JBoss Web Server をインストールするには、インストール ZIP ファイルをダウンロードし、展開します。

JBoss Web Server のダウンロード

1. ブラウザーを開き、[Red Hat カスタマーポータル](#) にログインします。
2. **ダウンロード** をクリックします。
3. **製品のダウンロード** リストの **Red Hat JBoss Web Server** をクリックします。
4. **Version** ドロップダウンメニューから正しい JBoss Web Server バージョンを選択します。
5. 以下の各ファイルの **Download** をクリックし、システムのプラットフォームおよびアーキテクチャーが正しく選択されていることを確認します。
 - Red Hat JBoss Web Server 3.1 Application Server(**jws-application-servers-3.1.0-
<platform> - <architecture> .zip**)

JBoss Web Server の抽出

1. ダウンロードした ZIP ファイルをインストールディレクトリーに展開します。ZIP アーカイブを展開して作成したディレクトリーは、JBoss Web Server の最上位ディレクトリーになります。これは **JWS_HOME** と呼ばれます。

3.3. JBOSS WEB SERVER インストールの設定

JBoss Web Server を実行する前に一部の設定が必要になります。本セクションには、以下の設定手順が含まれています。

- [環境変数の設定](#)
- [インストール後のスクリプトの実行](#)
- [Tomcat サービスのインストール](#)
- [Tomcat の log4j ロギングの有効化](#)
- [JBoss Web Server サービスのフォルダーパーミッションの設定](#)

環境変数の設定

1. ローカル管理者権限を持つアカウントにログインします。
2. **コントロールパネル** → **システム** と選択します。
3. **Advanced** タブをクリックします。
4. **環境変数** ボタンをクリックします。
5. **システム変数** の **新規** ボタンをクリックします。
6. **JAVA_HOME**、**TMP**、**TEMP** に、システムに適切な名前と値のペアを入力します。
7. SSL コネクターが機能するには、サービスを実行するユーザーの **PATH** 環境変数に **JWS_HOME\bin** を追加する必要があります。このユーザーは、デフォルトで **SYSTEM** です。

インストール後のスクリプトの実行

1. 管理者権限でコマンドプロンプトを開き、JBoss Web Server インストールの **etc** ディレクトリーに移動します。

```
cd /D "JWS_HOME\etc"
```

2. 以下のコマンドを使用して、インストール後に Tomcat を実行します。

```
call postinstall.tomcat.bat
```

このスクリプトは、一時的なロギングと設定ディレクトリーに必要なシンボリックリンク（詳細ポイント）を作成します。

Tomcat サービスのインストール

1. 管理者権限でコマンドプロンプトを開き、Tomcat バージョンの **bin** フォルダーに移動します。

```
cd /D "JWS_HOME\share\tomcat<VERSION>\bin"
```

2. 以下のコマンドを使用して、Tomcat サービスをインストールします。

```
call service.bat install
```

Tomcat の log4j ロギングの有効化

1. 管理者権限でコマンドプロンプトを開き、**JWS_HOME\share\extras**に変更します。
2. log4j ファイルを Tomcat バージョンの **lib** フォルダにコピーします。

```
copy log4j-eap6.jar log4j.properties tomcat-juli-adapters.jar ..\tomcat<VERSION>\lib
```

3. Tomcat **bin** ディレクトリーの **tomcat-juli.jar** ファイルを **JWS_HOME\share\extras** の **tomcat-juli.jar** ファイルに置き換えます。

```
copy tomcat-juli.jar ..\tomcat<VERSION>\bin
```

JBoss Web Server サービスのフォルダパーミッションの設定

以下の手順に従って、サービスの実行に使用するアカウントが **JWS_HOME** フォルダとそのすべてのサブフォルダを完全に制御できるようにします。

1. **JWS_HOME** ディレクトリーを右クリックし、**Properties** をクリックします。
2. **Security** タブを選択します。
3. **Edit** ボタンをクリックします。
4. **Add** ボタンをクリックします。
5. テキストボックスに **LOCAL SERVICE** を入力します。
6. **LOCAL SERVICE** アカウントの **Full Control** チェックボックスを選択します。
7. **OK** をクリックします。
8. **Advanced** ボタンをクリックします。
9. **Advanced Security Settings** ダイアログ内で **LOCAL SERVICE** を選択し、**Edit** をクリックします。
10. **Replace all existing inheritable permissions on all descendants with inheritable permissions from this object** オプションの横にあるチェックボックスを選択します。
11. 開いているすべてのフォルダプロパティウィンドウで **OK** をクリックして設定を適用します。

3.4. JBOSS WEB SERVER の起動

JBoss Web Server を起動するには、以下を開始する必要があります。

- Tomcat (7 または 8)

コマンドプロンプトまたは Computer Management ツールを使用してサービスを起動できます。

コマンドプロンプトからの JBoss Web Server の起動

1. 管理者権限でコマンドプロンプトを開きます。
2. Tomcat サービスを起動します。

```
net start tomcat<VERSION>
```

Computer Management Tool からの JBoss Web Server の起動

1. Start → Administrative Tools → Services と選択します。
2. Services 一覧で、サービス名(**tomcat**)を右クリックし、**Start** をクリックします。

注記

一部のサードパーティーアプリケーションは、Windows のシステムディレクトリーにライブラリーを追加します。これは、検索時に Tomcat ライブラリーよりも優先されます。つまり、これらのサードパーティーライブラリーの名前が Tomcat ネイティブライブラリーで使用されるものと同じ場合は、JBoss Web Server と配布されるライブラリーの代わりにロードされます。

この場合、Tomcat は起動しない可能性があり、Windows イベントログまたは Tomcat ログファイルにエラーメッセージを記録しません。エラーが発生するのは、**catalina.bat run** の使用した場合のみです。

この動作が発生した場合は、**C:\windows\System32**ディレクトリーおよびその他の **PATH** ディレクトリーの内容を検査し、JBoss Web Server に同梱される DLL がないことを確認します。特に、**libeay32.dll**、**ssleay32.dll**、**libssl32.dll** を検索します。

3.5. JBOSS WEB SERVER の停止

JBoss Web Server を停止するには、以下を停止する必要があります。

- Tomcat

コマンドプロンプトまたは Computer Management ツールを使用してサービスを停止できます。

コマンドプロンプトからの JBoss Web Server の停止

1. 管理者権限でコマンドプロンプトを開きます。
2. Tomcat サービスを停止します。

```
net stop tomcat<VERSION>
```

Computer Management Tool からの JBoss Web Server の停止

1. Start → Administrative Tools → Services と選択します。
2. Services 一覧で、サービス名(**tomcat**)を右クリックし、**Stop** をクリックします。

第4章 SOLARIS への JBOSS WEB SERVER のインストール

4.1. JAVA DEVELOPMENT KIT (JDK) のインストール

Solaris に JBoss Web Server をインストールする前に、最初にサポートされる Java Development Kit(JDK)をインストールする必要があります。

サポートされる構成の一覧は、Red Hat カスタマーポータルの記事「[JBoss Web Server 3 Supported Configurations](#)」を参照してください。

Java Development Kit (JDK) のインストール

root ユーザーとしてコマンドラインで Oracle JDK をインストールします。

```
# pkg install jdk-<version>
```

<version> は、インストールする JDK のバージョンです（**jdk-8**など）。

別の方法：Solaris での Java 開発キットのダウンロードおよびインストール

1. オペレーティングシステムおよびアーキテクチャー用の Oracle JDK をダウンロードします。JDK インストールファイルは、Oracle の Web サイト <http://www.oracle.com/technetwork/java/javase/downloads/index.html> からダウンロードできます。
2. JDK インストールファイルを実行します。
3. シェルプロンプトで **usr** を開き、以下のコマンドを実行して現在の Java シンボリックリンクを表示します。

```
ls -lad java
```

4. リンクを削除します。

```
rm java
```

5. 新たにインストールした JDK への Java シンボリックリンクを新たに作成します。

```
ln -sf /usr/jdk/<JDK>
```

4.2. JBOSS WEB SERVER のダウンロードおよび拡張

JBoss Web Server をインストールするには、インストール ZIP ファイルをダウンロードし、展開します。

JBoss Web Server のダウンロード

1. ブラウザーを開き、[Red Hat カスタマーポータル](#) にログインします。
2. **ダウンロード**をクリックします。
3. **製品のダウンロード** リストの **Red Hat JBoss Web Server** をクリックします。

4. **Version** ドロップダウンメニューから正しい JBoss Web Server バージョンを選択します。
5. 以下の各ファイルの **Download** をクリックし、システムのプラットフォームおよびアーキテクチャが正しく選択されていることを確認します。
 - Red Hat JBoss Web Server 3.1 Application Server(**jws-application-servers-3.1.0-
<platform> - <architecture& gt; .zip**)

JBoss Web Server の抽出

1. ダウンロードした ZIP ファイルをインストールディレクトリーに展開します。
ZIP アーカイブを展開して作成したディレクトリーは、JBoss Web Server の最上位ディレクトリーになります。これは **JWS_HOME** と呼ばれます。

4.3. JBOSS WEB SERVER インストールの設定

JBoss Web Server を実行する前に一部の設定が必要になります。本セクションには、以下の設定手順が含まれています。

- [インストール後のスクリプトの実行](#)
- [JAVA_HOME 環境変数の設定](#)
- [Tomcat の log4j ロギングの有効化](#)

インストール後のスクリプトの実行

1. シェルプロンプトを開き、**JWS_HOME/etc** ディレクトリーに移動します。
2. root ユーザーとして、インストール後のスクリプトを実行します。

```
# sh .postinstall.tomcat
```

インストール後のスクリプトは、Tomcat ユーザーおよびグループを作成します。

JAVA_HOME 環境変数の設定

JBoss Web Server を実行する前に、Tomcat の **JAVA_HOME** 環境変数を設定する必要があります。

1. Tomcat 設定ファイルを開きます。
 - Tomcat 7 の場合：**JWS_HOME/etc/sysconfig/tomcat7**
 - Tomcat 8 の場合：**JWS_HOME/etc/sysconfig/tomcat8**
2. 以下の行の最初にハッシュ(#)を削除します。

```
# JAVA_HOME="/usr/java"
```

Tomcat の log4j ロギングの有効化

必要な場合は、以下の手順にしたがって log4j ロギングを Tomcat に追加します。

1. シェルプロンプトを開き、**JWS_HOME/share/extras/** ディレクトリーに移動します。

2. **log4j-eap6.jar**、**log4j.properties**、および **tomcat-juli-adapters.jar** ファイルを Tomcat ディレクトリーの **lib** ディレクトリーにコピーします。
以下に例を示します。

```
# cp log4j.properties ../tomcat<VERSION>/lib/
```

```
# cp log4j-eap6.jar ../tomcat<VERSION>/lib/
```

```
# cp tomcat-juli-adapters.jar ../tomcat<VERSION>/lib/
```

< ;VERSION> をそれぞれの Tomcat バージョン番号 (7 または 8) に置き換えます。

3. Tomcat **bin** ディレクトリーの **tomcat-juli.jar** ファイルを **JWS_HOME/share/extras/** の **tomcat-juli.jar** ファイルに置き換えます。

```
# cp tomcat-juli.jar ../tomcat<VERSION>/bin/
```

4.4. JBOSS WEB SERVER の起動

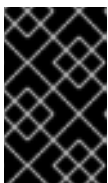
JBoss Web Server を起動するには、以下を開始する必要があります。

- Tomcat (7 または 8)

Tomcat の起動

1. root ユーザーとして、それぞれの Tomcat バージョン (7 または 8) で以下のコマンドを実行します。

```
# sh JWS_HOME/share/apache-tomcat-<VERSION>/bin/daemon.sh start
```



重要

Tomcat の起動方法は複数ありますが、**daemon.sh** スクリプトを使用することが推奨されます。Jsvc を使用して Tomcat をサービスとして起動するには、Jsvc の [章](#) を参照してください。

4.5. JBOSS WEB SERVER の停止

JBoss Web Server を停止するには、以下を停止する必要があります。

- Tomcat (7 または 8)

Tomcat の停止

1. Tomcat を停止するには、root ユーザーとしてシェルプロンプトで、それぞれの Tomcat バージョン (7 または 8) を指定して以下のコマンドを実行します。

```
# sh JWS_HOME/share/apache-tomcat-<VERSION>/bin/daemon.sh stop
```

第5章 JSVC を使用した TOMCAT の起動

Jsvc は、Linux、UNIX、および同様のオペレーティングシステムで Java アプリケーションの実行を容易にするライブラリーとアプリケーションのセットです。JBoss Web Server で Jsvc を使用すると、Tomcat はアイデンティティーを切り換えることができます。Jsvc を使用すると、Tomcat はルートレベルの操作を実行してから、非特権ユーザーに戻すことができます。Jsvc は主に Tomcat をサービスとして実行するために使用されます。

Jsvc ファイルは以下の場所にあります。

- Red Hat Enterprise Linux の場合 :
 - **JWS_HOME/extras/jsvc**
 - **JWS_HOME/tomcat<VERSION>/bin/jsvc**



注記

JWS_HOME/tomcat <VERSION> /bin/jsvc は **JWS_HOME/extras/jsvc** へのシンボリックリンクです。

- Solaris の場合 :
 - **JWS_HOME/sbin/jsvc**
 - **JWS_HOME/share/apache-tomcat-<VERSION>/bin/jsvc**



注記

JWS_HOME/share/apache-tomcat- <VERSION> /bin/jsvc は **JWS_HOME/sbin/jsvc** へのシンボリックリンクです。

5.1. JSVC を使用した TOMCAT の起動

Red Hat Enterprise Linux で Jsvc を使用した Tomcat の起動

以下のコマンドを実行して、Red Hat Enterprise Linux で Jsvc を使用して Tomcat を起動します。

```
JWS_HOME/tomcat<VERSION>/bin/daemon.sh start
```

Solaris での Jsvc を使用した Tomcat の起動

以下のコマンドを実行し、Solaris で Jsvc を使用して Tomcat を起動します。

```
JWS_HOME/share/tomcat<VERSION>/bin/daemon.sh start
```

5.2. JSVC を使用した TOMCAT の停止

Stop Tomcat Using Jsvc on Red Hat Enterprise Linux

以下のコマンドを実行して、Red Hat Enterprise Linux で Jsvc を使用して開始された Tomcat を停止します。

```
JWS_HOME/tomcat<VERSION>/bin/daemon.sh stop
```

Solaris での Jsvc を使用した Tomcat の停止

以下のコマンドを実行して、Solaris で Jsvc を使用して起動した Tomcat を停止します。

```
JWS_HOME/share/tomcat<VERSION>/bin/daemon.sh stop
```

5.3. JSVC パラメーター

`daemon.sh` スクリプトの実行時に以下のパラメーターを設定できます。

表5.1 `daemon.sh` 起動パラメーター

パラメーター名	Environment Variable	デフォルト値	詳細
<code>--java-home</code>	JAVA_HOME	PATH 変数の値に基づく。	Java ホームディレクトリーの場所。
<code>--catalina-home</code>	CATALINA_HOME	スクリプトの場所により決定されます。	Tomcat インストールディレクトリーの場所。
<code>--catalina-base</code>	CATALINA_BASE	PATH 変数の値に基づく。	複数のサーバーが同じインストールを使用している場合は、特定の設定および設定情報が含まれるディレクトリー。
<code>--catalina-pid</code>	-	\$CATALINA_BASE/logs/catalina-daemon.pid	Tomcat の実行中のインスタンスのプロセス ID(PID)が保存されるファイル。
<code>--tomcat-user</code>	-	tomcat	Tomcat が使用するユーザー。
<code>--service-start-wait-time</code>	-		--wait パラメーターのラッパーです。 --wait パラメーターは秒単位の値を受け入れます。

第6章 JBOSS WEB SERVER の HIBERNATE

Hibernate はオブジェクト関係マッピングフレームワークです。これは JBoss Web Server とは別にパッケージ化されます。このパッケージバージョンは、サポートされるすべてのプラットフォームで使用されます。

Hibernate は、通常の Tomcat インストールに使用されるのと同じ方法で使用されます。Hibernate JAR ファイルはデプロイメント WAR ファイルに追加されます。Tomcat は、**context.xml** で定義されるデフォルトの接続プールメカニズムを提供します。ただし、**persistence.xml** および **web.xml** も必要になります。以下の例は、Tomcat 接続プールメカニズムの設定を示しています。

- **/META-INF/context.xml** は Tomcat が作成する接続プールを定義します。

context.xml

```
<Context>
  <Resource
    name="jdbc/DsWebAppDB"
    auth="Container"
    type="javax.sql.DataSource"
    username="sa"
    password=""
    driverClassName="org.h2.Driver"
    url="jdbc:h2:mem:target/test/db/h2/hibernate"
    maxActive="8"
    maxIdle="4"/>
</Context>
```

- **/WEB-INF/classes/META-INF/persistence.xml** は JPA 設定ファイルです。これは、アプリケーションが Tomcat プールからの接続を消費するように Hibernate を設定する方法を定義します。Hibernate API を直接使用する場合は、**hibernate.cfg.xml** に表示される同様の設定を使用します。

persistence.xml

```
<persistence version="1.0"
  xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
  http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">

  <persistence-unit name="dswebapp">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect" />
      <property name="hibernate.connection.datasource"
value="java:comp/env/jdbc/DsWebAppDB"/>
    </properties>
  </persistence-unit>
</persistence>
```

- **/WEB-INF/web.xml** は通常の Web アプリケーションのデプロイメントファイルで、使用するデータソースを Tomcat に指示します。以下の例では、データソースは **jdbc/DsWebAppDB** です。

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <resource-env-ref>
    <resource-env-ref-name>jdbc/DsWebAppDB</resource-env-ref-name>
    <resource-env-ref-type>javax.sql.DataSource</resource-env-ref-type>
  </resource-env-ref>
</web-app>
```

詳細は、[JBoss Web Server の Hibernate ドキュメント](#) を参照してください。

第7章 JBOSS OPERATIONS NETWORK(ON)を使用した JBOSS WEB SERVER の監視

JBoss Web Server は Red Hat JBoss Operations Network(JBoss ON)によって監視および管理できます。

JBoss ON を使用して JBoss Web Server を監視するには、JBoss ON サーバーに **Web Server Plugin Pack** をインストールし、JBoss Web Server で Tomcat を監視するように設定する必要があります。

7.1. JBOSS ON の WEB サーバープラグインパックのダウンロード

JBoss ON を使用して JBoss Web Server を監視するには、まず **Web Server プラグインパック** を JBoss ON サーバーにダウンロードしてインストールする必要があります。

1. Web ブラウザーを開き、Red Hat カスタマーポータル(<http://access.redhat.com>)にログインします。
2. **ダウンロード**をクリックします。
3. **製品のダウンロード**リストの **Red Hat JBoss Web Server**をクリックします。
4. **Product** ドロップダウンメニューで **JBoss ON for Web Server**を選択します。
5. リストで **Red Hat JBoss Operations Network の Web Server プラグインパック**を見つけ、**Download** をクリックします。

JBoss ON サーバーにプラグインをインストールする方法については、JBoss ON ドキュメントを参照してください。

7.2. JBOSS ON MONITORING 用の TOMCAT の設定

JBoss ON が JBoss Web Server で Tomcat を監視できるようにするには、JBoss ON 検出を許可するように Tomcat を設定し、JBoss ON に必要なアクセスを提供する必要があります。

Linux オペレーティングシステムでのユーザーパーミッションの設定



注記

Microsoft Windows の場合は、この手順を省略し、[JBoss ON Monitoring の Tomcat の設定](#)に進みます。

JBoss ON エージェントには、Tomcat ディレクトリーへの読み取りおよび書き込みパーミッションが必要です。root 権限を持つユーザーとして以下のコマンドを実行し、JBoss ON Agent を実行するユーザーを **tomcat** グループに追加します。

```
# usermod -aG tomcat <JBOSSON_AGENT_USER>
```

JBoss ON Monitoring 用の Tomcat の設定

JBoss Web Server インスタンスは、Linux プラットフォームおよび Unix プラットフォームで自動検出されます。ただし、認証と正確な Tomcat 監視を適切に処理できるように、インスタンスの JMX を設定する必要があります。

認証を処理するように JMX を設定するには、以下を行います。

1. 編集する JBoss Web Server インスタンスの **startup** または **setenv** ファイルを開きます。
 - ZIP ファイルからインストールされた Red Hat Enterprise Linux で、**JWS_HOME/tomcat <VERSION> /bin/setenv.sh** を開きます。
 - RPM からインストールした Red Hat Enterprise Linux で、**/usr/sbin/tomcat <VERSION>** を開きます。
 - Microsoft Windows で **JWS_HOME\share\tomcat <VERSION> \bin\setenv.bat** を開いてください。
 - **daemon.sh** を使用して Tomcat を起動する Solaris で **JWS_HOME/tomcat <VERSION> /bin/setenv.sh** を開きます。
2. JMX 監視に使用できるポートを定義します。ポートがファイアウォールによってブロックされていないことを確認します。

- Red Hat Enterprise Linux および Solaris の場合：

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.port=PORT_NUMBER -
Djava.rmi.server.hostname=IP_ADDRESS"
```

- Microsoft Windows の場合：

```
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.port=PORT_NUMBER -
Djava.rmi.server.hostname=IP_ADDRESS"
```

3. 実稼働環境では、以下の行を起動ファイルの JAVA_OPTS 変数に追加し、SSL で JMX をセキュアにし、ファイアウォールでアクセスを制限します。

- Red Hat Enterprise Linux および Solaris の場合：

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.access.file=JWS_HOME/jmxremote.access"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.password.file=JWS_HOME/jmxremote.password"
```

- Microsoft Windows の場合：

```
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.access.file=JWS_HOME\jmxremote.access"
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.password.file=JWS_HOME\jmxremote.password"
```



注記

開発目的で認証および SSL を無効にする場合は、スタートアップファイルの JAVA_OPTS 変数に以下の行を追加します。

```
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.authenticate=false"
```

4. Tomcat サーバーリソースを検出し、JBoss ON インベントリーにインポートしたら、新しいリソースの接続設定を更新する必要がある場合があります。
 - a. JBoss ON インターフェースでは、新たにインポートされた Tomcat サーバーリソースの **Connection Settings**。
 - b. **Manager URL** プロパティの値を RMI URL に確認し、Tomcat サーバー起動ファイルで定義されている正しい JMX ホスト名とポート番号を使用していることを確認します。この値の例を以下に示します。

```
service:jmx:rmi:///jndi/rmi://$IP_ADDRESS:$PORT/jmxrmi
```

7.2.1. RPM から Tomcat Installed の JBoss ON Monitoring の設定

1. シェルプロンプトでは、root ユーザーになります。
2. start セクションおよび **start -security** セクションの `/usr/sbin/tomcat <VERSION>` ファイルで JMX JAVA_OPTS プロパティを設定します。

```
if [ "$1" = "start" ]; then
  JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.port=8100 -
  Dcom.sun.management.jmxremote.ssl=false -
  Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxr
  emote.access.file="/etc/tomcat<VERSION>/jmxremote.access" -
  Dcom.sun.management.jmxremote.password.file="/etc/tomcat<VERSION>/jmxremote.passw
  ord""

  ${JAVACMD} $JAVA_OPTS $LOGGING_CONFIG $CATALINA_OPTS \
  -classpath "$CLASSPATH" \
  -Dcatalina.base="$CATALINA_BASE" \
  -Dcatalina.home="$CATALINA_HOME" \
  -Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" \
  -Djava.io.tmpdir="$CATALINA_TMPDIR" \
  org.apache.catalina.startup.Bootstrap start \
  >> ${CATALINA_BASE}/logs/catalina.out 2>&1 &
  if [ ! -z "$CATALINA_PID" ]; then
    echo $! > $CATALINA_PID
  fi
elif [ "$1" = "start-security" ]; then
  JAVA_OPTS="{JAVA_OPTS} -Dcom.sun.management.jmxremote.port=8100 -
  Dcom.sun.management.jmxremote.ssl=false -
  Dcom.sun.management.jmxremote.authenticate=true -Dcom.sun.management.jmxr
  emote.access.file="/etc/tomcat<VERSION>/jmxremote.access" -
  Dcom.sun.management.jmxremote.password.file="/etc/tomcat<VERSION>/jmxremote.passw
  ord""

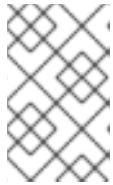
  ${JAVACMD} $JAVA_OPTS $LOGGING_CONFIG $CATALINA_OPTS \
  -classpath "$CLASSPATH" \
  -Dcatalina.base="$CATALINA_BASE" \
```

3. 以下のコマンドを実行して Tomcat を起動します。

```
service tomcat<VERSION> start
```

4. JBoss ON エージェントを起動します。

- JBoss ON Web UI で、JBoss ON エージェントおよび Tomcat をインポートします。
- JBoss ON Web UI で、Tomcat 接続設定（プリンシパルおよび認証情報）を設定します。
- JBoss ON Web UI で、Tomcat コントロール メソッド設定を RPM System V init スクリプトに設定します。



注記

Tomcat プラグインは RPM System V init スクリプト設定に対してサービス **tomcat <VERSION> start/stop** コマンドを常に実行されるため、Start スクリプトおよび Shutdown スクリプト 設定は設定されません。

7.2.2. Windows サービスとしてインストールされた Tomcat の JBoss ON Monitoring の設定

- C:\jmx** ディレクトリーに **controlRole readwrite** で **jmxremote.access** ファイルを作成します。
- C:\jmx** ディレクトリーに **controlRole pwd** を指定して **jmxremote.password** ファイルを作成します。



注記

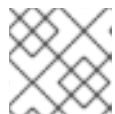
jmxremote.access および **jmxremote.password** の所有者を SYSTEM に設定し、**jmxremote.password** のアクセスを SYSTEM だけに制限します。SYSTEM ユーザーは読み取りアクセスのみを持っている必要があります。

- Tomcat Windows サービスで JMX を有効にします。

```
JWS_HOME\sbin\tomcat<VERSION>.exe //US//Tomcat<VERSION> ++JvmOptions="-
Dcom.sun.management.jmxremote.port=8100;-
Dcom.sun.management.jmxremote.access.file="C:\jmx\jmxremote.access";-
Dcom.sun.management.jmxremote.password.file="C:\jmx\jmxremote.password";-
Dcom.sun.management.jmxremote.ssl=false;-
Dcom.sun.management.jmxremote.authenticate=true"
```

- Tomcat Windows サービスを起動し、それが稼働していることを確認します。
- JBoss ON エージェントをインストールして設定します。エージェントプロンプトで **discovery** と入力して、Tomcat Windows サービスを検出します。
- JBoss ON Web UI で **Inventory** をクリックし、**Discovery Queue** をクリックし、**import Tomcat および RHQ エージェント** を選択します。
- JBoss ON Web UI で、プラットフォームに移動し、エージェント名を検索します。エージェントをクリックします。
- Agent** ページで、Tomcat Server が一覧表示されます。Tomcat サーバーをクリックして選択します。
- JBoss ON Web UI で **Inventory** タブをクリックしてから、**Connection Settings** で Tomcat Server を設定します。

10. Principal および Credentials 情報を入力します。**jmxremote** ファイルに設定されている **controlRole** および password を使用します。
11. 制御メソッドを **RPM System V init** スクリプトに設定します。



注記

Start および Shutdown スクリプトフィールドは設定できません。

12. **保存** をクリックします。
13. Tomcat Server JVM の接続設定を更新し、Principal および Credentials を設定します。

第8章 RED HAT JBOSS WEB SERVER 3 でのパスワード VAULT の使用

8.1. RED HAT JBOSS WEB SERVER 3 でのパスワード VAULT の使用

パスワード vault は、パスワードやその他の機密文字列をマスクし、暗号化された Java キーストアに保存するために使用されます。Tomcat は vault を使用してキーストアからパスワードやその他の機密文字列を検索することができるため、Tomcat 設定ファイルにクリアテキストのパスワードの保存をなくすることができます。

以下の例とコマンドは、**JWS_HOME** を JBoss Web Server インストールディレクトリーとして使用します。**JWS_HOME** を JBoss Web Server インストールへのパスに置き換えます。また、以下のパスではディレクトリー区切り文字として / が使用されます。

8.1.1. JBoss Web Server パスワード vault のインストール

JBoss Web Server パスワード vault をインストールする方法は 2 つあります。

- [RPM から Red Hat Enterprise Linux に JBoss Web Server パスワード vault をインストールする](#) ([yum](#) を使用して JBoss Web Server がインストールされている Red Hat Enterprise Linux システムの場合)。
- [.zip アーカイブからの Vault ファイルのダウンロードおよび 拡張](#) (すべての JBoss Web Server インストールに利用可能)

8.1.1.1. RPM からの Red Hat Enterprise Linux への JBoss Web Server パスワード vault のインストール

JBoss Web Server が Red Hat Enterprise Linux の RPM からインストールされている場合は、以下のコマンドを実行してパスワード vault を root ユーザーとしてインストールします。

```
yum install tomcat-vault tomcat-vault-tomcat<VERSION>
```

ここで、<VERSION> は tomcat 7 ([tomcat-vault-tomcat7](#)) または tomcat 8 ([tomcat-vault-tomcat8](#)) の場合は 8 になります。



注記

tomcat-vault RPM インストールでは、vault jar は `/usr/share/java/vault-tomcat-<VERSION>-jar-with-dependencies.jar` にあります。この jar は JWS zip インストールで使用できます (`JWS_HOME/tomcat-<VERSION>/lib/` の場合は JWS RPM インストールの場合: `/usr/share/tomcat<VERSION>/lib`)。

8.1.1.2. .zip アーカイブからの Vault ファイルのダウンロードおよび拡張

1. Tomcat が実行されている場合は停止します。
2. vault zip の内容を **JWS_HOME** ディレクトリーに展開します。このトピックでは、**JWS_HOME/tomcat-vault** は展開した vault ディレクトリーを参照します。
3. **JWS_HOME/tomcat-vault/modules/system/layers/base/tomcat-vault/main/tomcat-vault.jar** を **JWS_HOME/tomcat<VERSION>/lib/** にコピーします。**JWS_HOME/tomcat<VERSION>/conf/catalina.properties** を編集し、以下の行を追加し

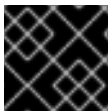
ます。

```
org.apache.tomcat.util.digester.PROPERTY_SOURCE=org.apache.tomcat.vault.util.PropertySourceVault
```

8.1.2. Java キーストアの作成

パスワード vault を使用するには、最初に Java キーストアを作成する必要があります。これは、**keytool -genseckey** コマンドを使用して実行できます。以下に例を示します。

```
$ keytool -genseckey -keystore JWS_HOME/tomcat/vault.keystore -alias my_vault -storetype jceks -keyalg AES -keysize 128 -storepass <vault_password> -keypass <vault_password> -validity 730
```



重要

上記の値は例です。ご使用の環境に固有の値に置き換えてください。

パラメーターの説明は、**keytool -genseckey -help** コマンドを使用します。

8.1.3. JWS_HOME ディレクトリー外での tomcat-vault vault.properties ファイルの保存

この機能は JBoss Web Server 3.1 Service Pack 2 によって導入されました。

tomcat-vault の vault.properties ファイルは、**CATALINA_BASE/conf/** ディレクトリーの **JWS_HOME/tomcat<VERSION>/conf/** の外に保存できます（設定されている場合）。

CATALINA_BASE ディレクトリーを設定するには、Apache Tomcat Web サイトにある『[Running The Apache Tomcat 8.0 Servlet/JSP Container](#)』の「[Advanced Configuration - Multiple Tomcat Instances](#)」セクションの説明に従います。



注記

CATALINA_BASE のデフォルトの場所は **JWS_HOME/tomcat<VERSION>/**（別名 CATALINA_HOME）です。

CATALINA_BASE の設定に関する詳細は、以下を参照してください。

- [Apache Tomcat 8: Introduction - directories and Files](#)
- [Running The Apache Tomcat 8.0 Servlet/JSP Container: Advanced Configuration - Multiple Tomcat Instances](#)

8.1.4. パスワード vault の初期化

vault は機密文字列の保存に使用する前に初期化する必要があります。これは、**JWS_HOME/tomcat-vault/bin/tomcat-vault.sh** vault スクリプトを使用して行われます。Microsoft Windows の場合、スクリプトは **tomcat-vault.bat** です。

スクリプトは対話的または非対話的に実行できます。以下は、前の手順でキーストアの例を使用して、パスワード vault を初期化するスクリプトのインタラクティブな実行例です。

8.1.4.1. Apache Tomcat の Vault の対話的な初期化



重要

以下の値は例としてのみ使用されます。ご使用の環境に適切な値に置き換えてください。

```
# JWS_HOME/tomcat-vault/bin/tomcat-vault.sh
```

```
WARNING JBOSS_HOME may be pointing to a different installation - unpredictable results may occur.
```

```
=====
```

```
JBoss Vault
```

```
JBOSS_HOME: JWS_HOME/tomcat-vault
```

```
JAVA: java
```

```
=====
```

```
*****
****  JBoss Vault  *****
*****
```

```
Please enter a Digit::
```

```
0: Start Interactive Session
1: Remove Interactive Session
2: Exit
```

```
0
```

```
Starting an interactive session
```

```
Enter directory to store encrypted files: JWS_HOME/tomcat-vault/
```

```
Enter Keystore URL: JWS_HOME/tomcat-vault/vault.keystore
```

```
Enter Keystore password: <vault_password>
```

```
Enter Keystore password again: <vault_password>
```

```
Values match
```

```
Enter 8 character salt: 1234abcd
```

```
Enter iteration count as a number (Eg: 44): 120
```

```
Enter Keystore Alias: my_vault
```

```
Initializing Vault
```

```
Jun 16, 2018 10:24:27 AM org.apache.tomcat.vault.security.vault.PicketBoxSecurityVault init
```

```
INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready
```

```
Vault Configuration in tomcat properties file:
```

```
*****
```

```
...
```

```
KEYSTORE_URL=JWS_HOME/tomcat-vault/vault.keystore
```

```
KEYSTORE_PASSWORD=MASK-3CuP21KMHn7G6iH/A3YpM/
```

```
KEYSTORE_ALIAS=my_vault
```

```
SALT=1234abcd
```

```
ITERATION_COUNT=120
```

```
ENC_FILE_DIR=JWS_HOME/tomcat-vault/
```

```
...
```

```
**
```

```
Vault is initialized and ready for use
```

```
Handshake with Vault complete
```

```
Please enter a Digit::
0: Store a secured attribute
1: Check whether a secured attribute exists
2: Exit
```

```
2
```

Tomcat プロパティファイルの出力に注意してください。これは、Tomcat が vault を使用するよう
に設定する必要があるためです。

パスワード vault を使用するように Tomcat を設定する
JWS_HOME/tomcat<VERSION>/conf/ で、vault の初期化時に生成された vault 設定を含む
vault.properties という名前のファイルを作成します。以下の値は、前の手順で初期化された vault の
例を使用します。



注記

KEYSTORE_PASSWORD には vault の初期化時に生成されたマスクされた値を使用す
る必要があります。

```
KEYSTORE_URL=JWS_HOME/tomcat-vault/vault.keystore
KEYSTORE_PASSWORD=MASK-3CuP21KMhN7G6iH/A3YpM/
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=JWS_HOME/tomcat-vault/
```

8.1.4.2. Apache Tomcat の Vault の非対話的な初期化 (サイレント設定)

Apache Tomcat の Vault は、tomcat-vault.sh スクリプトに必要な入力を引数として提供して非対話的
に作成できます。vault.properties ファイルは、-g, --generate-config オプションが使用される場合に
tomcat-vault.sh スクリプトの出力としても作成されます。



重要

以下の値は例としてのみ使用されます。ご使用の環境に適切な値に置き換えてくださ
い。

```
$ JWS_HOME/tomcat-vault/bin/tomcat-vault.sh \
--keystore JWS_HOME/tomcat-vault/vault.keystore \
--keystore-password <vault_password> \
--alias my_vault \
--enc-dir JWS_HOME/tomcat-vault/ \
--iteration 120 \
--salt 1234abcd \
--generate-config JWS_HOME/tomcat<VERSION>/conf/vault.properties
```

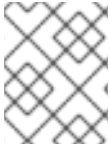
8.1.5. パスワード Vault への Sensitive 文字列の格納

前の手順で使用した vault スクリプトは、機密文字列をパスワード vault に保存するためにも使用され
ます。スクリプトは対話的または非対話的に実行できます。

文字列をパスワード vault に追加する場合、機密文字列には参照される名前が必要です。パスワード vault では、この名前は `attribute name` と呼ばれ、パスワード自体は `secured attribute` と呼ばれます。

以下の例は、vault スクリプトを非対話的に使用してパスワードを保存する方法を示しています。前の手順で初期化された vault を使用し、機密文字列 `P@SSW0#D` を属性名 `manager_password` で保存します。

```
$ JWS_HOME/tomcat-vault/bin/tomcat-vault.sh --keystore JWS_HOME/tomcat-vault/vault.keystore -  
-keystore-password <vault_password> --alias my_vault --enc-dir JWS_HOME/tomcat-vault/ --  
iteration 120 --salt 1234abcd --vault-block my_block --attribute manager_password --sec-attr  
P@SSW0#D
```



注記

オプションで、パスワードを保存する vault ブロックを指定できます。ブロックを指定しないと、自動的に作成されます。上記の例では、`my_block` が使用されます。

8.1.6. Tomcat 設定で保存された機密文字列の使用

パスワード vault に機密文字列を保存したら、保存された文字列の属性を `${VAULT::block_name::attribute_name::}` として入力して設定ファイルで参照できます。

たとえば、前の手順で保存されたパスワードを使用するには、以下を置き換えます。

```
<user username="manager" password="P@SSW0#D" roles="manager-gui"/>
```

上記の行を、以下のように置き換えます。

```
<user username="manager" password="${VAULT::my_block::manager_password::}"  
roles="manager-gui"/>
```

その結果、Tomcat 設定ファイルに、パスワードへの参照のみが表示され、実際のパスワードはパスワード vault にのみ保存されます。

第9章 WEBSOCKET を使用した JWS クライアントサーバー通信の設定

9.1. WEBSOCKET について

WebSocket は、双方向で完全な duplex メッセージを提供し、1つの TCP ソケット接続上でクライアントとサーバー間で即時に分散される Web 技術です。完全な duplex 通信は、双方向通信を同時に許可します。

コンテナは WebSockets 1.0 JSR 356 API の実装を提供します。API を使用するには、Java 7 以降を実行し、Web コンテナの APR または NIO2 HTTP/1.1 コネクターを設定する必要があります。

JSR 356 は、Java 向け WebSocket API の標準です。開発者は JSR 356 API を使用して、実装とは関係なく WebSocket アプリケーションを作成できます。WebSocket API は純粋にイベント駆動型です。

開発者は、WebSocket 用に JSR 356 Java API を使用して、サーバー側およびクライアント側で WebSocket をアプリケーションに統合できます。Tomcat 7 および 8 は、JSR-356 標準仕様に準拠する WebSocket プロトコルを実装します。

Java クライアントは JSR 356 準拠のクライアント実装を使用して WebSocket サーバーに接続します。Web クライアントでは、WebSocket JavaScript API を使用して WebSocket サーバーと通信できます。WebSocket クライアントとサーバー間の唯一の違いは、WebSocket クライアントと WebSocket サーバーが接続されている方法です。WebSocket クライアントは、ピアへの接続元となる WebSocket ポイントです。WebSocket サーバーはすでにパブリッシュされ、ピアからの接続を待機する WebSocket エンドポイントです。

WebSocket が使用できる例には、銀行、チャット、マルチプレイヤー、ソーシャルネットワークアプリケーションなどがあります。

9.2. TOMCAT での WEBSOCKET の実装

Tomcat で WebSocket を設定するには、以下の個別の設定が必要です。

- [書き込みタイムアウト](#)
- [受信バイナリーメッセージ](#)
- [受信テキストメッセージ](#)
- [追加のプログラムによるデプロイメント](#)
- [非同期書き込みのコールバック](#)
- [接続の確立中の IO 操作のタイムアウト](#)

9.2.1. 書き込みタイムアウトの設定

`org.apache.tomcat.websocket.BLOCKING_SEND_TIMEOUT` プロパティを使用すると、ブロックモードで書き込みタイムアウトを変更できます。このプロパティはミリ秒単位の値を受け入れます。デフォルト値は 20000 (20 秒) です。

9.2.2. 着信バイナリーメッセージの設定

受信バイナリーメッセージを設定するには、`MessageHandler.Partial` を定義する必要があります。`MessageHandler.Partial` が定義されていない場合は、メッセージ全体が `MessageHandler.Whole` への単一呼び出しで配信されるように、受信バイナリーメッセージをバッファする必要があります。

バイナリーメッセージのデフォルトのバッファサイズは 8192 バイトです。web アプリケーションのバッファサイズは、サーブレットコンテキストの初期化パラメーター `org.apache.tomcat.websocket.binaryBufferSize` の値を変更することで変更できます。

9.2.3. 着信テキストメッセージの設定

受信テキストメッセージを設定するには、`MessageHandler.Partial` を定義する必要があります。`MessageHandler.Partial` が定義されていない場合、受信メッセージ全体が `MessageHandler.Whole` への単一呼び出しで配信されるようにバッファする必要があります。

テキストメッセージのデフォルトのバッファサイズは 8192 バイトです。web アプリケーションのバッファサイズは、サーブレットコンテキストの初期化パラメーター `org.apache.tomcat.websocket.textBufferSize` の値を変更することで変更できます。

9.2.4. 追加のプログラムによるデプロイメントの設定

Java WebSocket 1.0 仕様では、最初のエンドポイントが WebSocket ハンドシェイクを開始した後にプログラムによるデプロイメントが許可されません。ただし、デフォルトで Tomcat では、追加のプログラムによるデプロイメントが可能です。追加のプログラムによるデプロイメントは、サーブレットコンテキスト初期化パラメーター `org.apache.tomcat.websocket.noAddAfterHandshake` を使用して実行できます。

システムプロパティ `org.apache.tomcat.websocket.STRICT_SPEC_COMPLIANCE` を `true` に設定して、デフォルト設定を変更します。

9.2.5. 非同期書き込みのコールバックの設定

非同期書き込みのコールバックは、書き込みを開始したスレッドに対して別のスレッドで実行する必要があります。コンテナスレッドプールは Servlet API 経由で公開されません。そのため、WebSocket の実装は独自のスレッドプールを提供する必要があります。

以下のサーブレットコンテキストの初期化パラメーターは、スレッドプールを制御します。

`org.apache.tomcat.websocket.executorCoreSize`

エグゼキュータースレッドプールのコアサイズ。設定されていない場合、デフォルトの 0 (ゼロ) が使用されます。

`org.apache.tomcat.websocket.executorMaxSize`

エグゼキュータースレッドプールの最大許容サイズ。設定されていない場合は、デフォルトの 10 が使用されます。

`org.apache.tomcat.websocket.executorKeepAliveTimeSeconds`

アイドルスレッドが終了までエグゼキュータースレッドプールに留まる最大時間。指定のない場合は、デフォルトの 60 秒が使用されます。

9.2.6. IO 操作のタイムアウトの設定

接続の確立中の IO 操作のタイムアウトは、提供された `javax.websocket.ClientEndpointConfig` の `userProperties` によって制御されます。`org.apache.tomcat.websocket.IO_TIMEOUT_MS` プロパティを変更すると、タイムアウトを変更できます。このプロパティは値をミリ秒単位で受け入れます。デフォルト値は 5000 (5 秒) です。

WebSocket クライアントに接続してサーバーエンドポイントのセキュリティーを保護するには、クライアント SSL 設定は、提供された `javax.websocket.ClientEndpointConfig` の `userProperties` によって制御されます。

以下のユーザープロパティーがサポートされます。

- `org.apache.tomcat.websocket.SSL_CONTEXT`
- `org.apache.tomcat.websocket.SSL_PROTOCOLS`
- `org.apache.tomcat.websocket.SSL_TRUSTSTORE`
- `org.apache.tomcat.websocket.SSL_TRUSTSTORE_PWD`

デフォルトのトラストストアのパスワードは `changeit` です。`org.apache.tomcat.websocket.SSL_TRUSTSTORE` および `org.apache.tomcat.websocket.SSL_TRUSTSTORE_PWD` プロパティーは、`org.apache.tomcat.websocket.SSL_CONTEXT` プロパティーが設定されている場合は無視されません。

付録A JAVA IPV4/IPV6 プロパティ

Java プロパティの設定

Java では、`r` は IPv4 と IPv6 の設定に使用される 2 つのプロパティです。これらは `java.net.preferIPv4Stack` および `java.net.preferIPv6Addresses` です。

`java.net.preferIPv4Stack` (default: false)

IPv6 が利用できる場合は、デフォルトで基礎となるネイティブソケットが IPv6 ソケットになります。このソケットにより、アプリケーションが IPv4 および IPv6 ホストからの接続を許可できるようになります。アプリケーションが IPv4 ソケットのみを使用する場合は、このプロパティを `true` に設定します。ただし、アプリケーションが IPv6 のみのホストと通信することはできません。

`java.net.preferIPv6Addresses` (デフォルト: false)

ホストに IPv4 アドレスと IPv6 アドレスの両方があり、IPv6 が利用できる場合には、デフォルトの動作では IPv6 上で IPv4 アドレスを使用します。これにより、後方互換性を維持できます。192.168.1.1 など、IPv4 アドレス表現に依存するアプリケーションの場合、可能な場合は IPv4 で IPv6 アドレスを使用するように、このプロパティを `true` に設定します。

これらのプロパティを Tomcat に渡すには、`CATALINA_HOME/bin/setenv.*` ファイルに `CATALINA_OPTS` を設定します。



注記

`CATALINA_HOME/bin/setenv.sh` または `CATALINA_HOME/bin/setenv.bat` ファイルが存在しない場合は、作成する必要があります。

Linux の場合:

```
export "CATALINA_OPTS=-Djava.net.preferIPv4Stack=YOUR_VALUE -
Djava.net.preferIPv6Addresses=YOUR_VALUE"
```

Windows の場合:

```
set "CATALINA_OPTS=-Djava.net.preferIPv4Stack=YOUR_VALUE -
Djava.net.preferIPv6Addresses=YOUR_VALUE"
```

Tomcat バインディングの設定

Tomcat バインディングは、IPv6 アドレスを使用して `CATALINA_HOME/conf/server.xml` で設定できます。

- Tomcat バインディングアドレスを指定します。
<Server ... address="TOMCAT_BINDING_ADDRESS">
- HTTP コネクターアドレスを指定します。
<connector protocol="HTTP/1.1" ... address="HTTP_CONNECTOR_ADDRESS">
- AJP コネクターアドレスを指定します。
<Connector protocol="AJP/1.3" ... address="AJP_CONNECTOR_ADDRESS">