Red Hat JBoss Web Server 5.0

Installation Guide

Install and Configure Red Hat JBoss Web Server 5.0

Last Updated: 2018-08-30

Red Hat JBoss Web Server 5.0 Installation Guide

Install and Configure Red Hat JBoss Web Server 5.0

Legal Notice

Abstract

This book contains information related to installation and basic configuration of Red Hat JBoss Web Server.

Table of Contents

CHAPTER 1. INTRODUCTION

1.1. ABOUT RED HAT JBOSS WEB SERVER

The JBoss Web Server is a fully integrated and certified set of components for hosting Java web applications. It consists of:

- **Apache Tomcat**: a servlet container in accordance with the Java Servlet Specification. JBoss Web Server contains Apache Tomcat 9.
- The **Apache Tomcat Native Library**: a Tomcat library, which improves Tomcat scalability, performance, and integration with native server technologies.
- The **tomcat-vault**: an extension for the JBoss Web Server used for securely storing passwords and other sensitive information used by a JBoss Web Server.
- The **mod_cluster** library: a library that allows communication between Apache Tomcat and the Apache HTTP Server's mod_proxy_cluster module. This allows the Apache HTTP Server to be used as a load balancer for JBoss Web Server. For information on the configuration of mod_cluster, or for information on the installation and configuration of the alternative load balancers mod_jk and mod_proxy, see the HTTP Connectors and Load Balancing Guide .

NOTE

- If you need clustering or session replication support for Java applications, Red Hat recommends that you use Red Hat JBoss Enterprise Application Platform (JBoss EAP).
- For a detailed list of component versions included in JBoss Web Server 5.0, see https://access.redhat.com/articles/111723.

This Installation Guide includes procedures for the installation, minor upgrade, and basic configuration of the Tomcat servers from JBoss Web Server on supported operating systems. Installation and configuration instructions for the Apache HTTP Server are covered in the JBoss Core Services Documentation.

1.2. SUPPORTED OPERATING SYSTEMS AND CONFIGURATIONS

For information on supported operating systems and configurations for JBoss Web Server, see https://access.redhat.com/articles/3497401/.

1.3. INSTALLATION METHODS

JBoss Web Server can be installed on supported Red Hat Enterprise Linux, Microsoft Windows, and Solaris systems using ZIP installation files available for each platform. JBoss Web Server can also be installed on supported Red Hat Enterprise Linux systems using RPM packages.

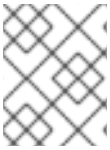
For ZIP installations, below is a summary of the components that are included in the ZIP files which form the core part of a JBoss Web Server installation.

- `jws-application-server-5.0.0.zip`
  - Tomcat 9

- mod_cluster
- tomcat-vault

- **`jws-application-server-5.0.0-<platform>-<architecture>.zip`**
  - Platform-specific utilities

1.4. UPGRADING JBOSS WEB SERVER

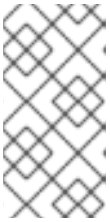


NOTE

The JBoss Web Server 5.0 can run on the same system as previous releases, however, this is not supported.

For systems using the JBoss Web Server 3.1, the recommended procedure for upgrading to the JBoss Web Server 5.0 is:

1. Shutdown any running instances of JBoss Web Server 3.1.
2. Backup the JBoss Web Server 3.1 installation and configuration files.
3. Change the ports used by the JBoss Web Server 3.1 connectors if the ports used are **`8080`** or **`8443`**.
4. Install JBoss Web Server 5.0 using one of the following guides:
   - Installing JBoss Web Server on Red Hat Enterprise Linux
   - Installing JBoss Web Server on Microsoft Windows
   - Installing JBoss Web Server on Solaris
5. Migrate your configuration from the JBoss Web Server 3.1



NOTE

The JBoss Web Server configuration files may have changed since the JBoss Web Server 3.1 release. It is recommended that you update the 5.0 version configuration files, rather than overwrite them with the configuration files from a different version (such as JBoss Web Server 3.1).

6. Remove the JBoss Web Server 3.1:
   - For systems where JBoss Web Server 3.1 was installed from RPM packages, uninstall using:

     ```
     yum group remove jws3
     ```

   - For systems where JBoss Web Server 3.1 was installed from .zip archives, uninstall by deleting the JBoss Web Server 3.1 root directory.

1.5. COMPONENT DOCUMENTATION BUNDLE

JBoss Web Server includes an additional documentation bundle that includes the original vendor documentation for each component. This documentation bundle, **`jws-docs-5.0.0.zip`**, is available at the Red Hat Customer Portal, and contains additional documentation for the following:

- tomcat
- tomcat-native
- tomcat-vault

CHAPTER 2. INSTALLING JBOSS WEB SERVER ON RED HAT ENTERPRISE LINUX

You can install JBoss Web Server on Red Hat Enterprise Linux using one of two methods:

- ZIP files
- RPM packages

Regardless of which method you choose, you must first install a supported Java Development Kit (JDK).

2.1. PREREQUISITES

2.1.1. Installing a Java Development Kit (JDK)

Before installing JBoss Web Server, you must first install a supported Java Development Kit (JDK).

For a list of supported JDKs for Red Hat JBoss Web Server 5.0, see: JBoss Web Server 5 Supported Configurations.

The installation of the OpenJDK or the IBM JDK are presented here. To install the Oracle JDK, follow the instructions provided by Oracle at: http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Installing a JDK using the YUM package manager

1. Subscribe your Red Hat Enterprise Linux system to the appropriate channel:
   - **OpenJDK:**
     - rhel-6-server-rpms
     - rhel-7-server-rpms
   - **IBM:**
     - rhel-6-server-supplementary-rpms
     - rhel-7-server-supplementary-rpms
2. As the root user, execute the command to install a 1.8 JDK:

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

   Replace **<*VENDOR*>** with **`ibm`** or **`openjdk`**.
3. Run the following commands as the root user to ensure the correct JDK is in use:

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

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

These commands return lists of available JDK versions with the selected version marked with a plus (+) sign. If the selected JDK is not the desired one, change to the desired JDK as instructed in the shell prompt.

IMPORTANT

All software that use the **`java`** and **`javac`** commands uses the JDK set by **`alternatives`**. Changing Java alternatives may impact on the running of other software.

Installing a JDK from a compressed archive (such as .zip or .tar)
If the JDK was downloaded from the vendor's website (Oracle, IBM or OpenJDK), use the installation instructions provided by the vendor and set the **`JAVA_HOME`** environment variable.

If the JDK has was installed from a compressed archive, set the **`JAVA_HOME`** environment variable for Tomcat before running JBoss Web Server.

In the **`bin`** directory of Tomcat (***`JWS_HOME`***`/tomcat/bin`), create a file named **`setenv.sh`**, and insert the **`JAVA_HOME`** path definition.

For example:

```
$ cat JWS_HOME/tomcat/bin/setenv.sh

export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk.x86_64
```

2.1.2. Red Hat Enterprise Linux Package Prerequisites

Before installing JBoss Web Server on Red Hat Enterprise Linux, ensure the following prerequisites are met.

- A supported JDK is installed.
- You must remove the **`tomcatjss`** package before installing the **`tomcat-native`** package. The **`tomcatjss`** package uses an underlying NSS security model rather than the OpenSSL security model.

Removing the tomcatjss Package

1. As the root user, run the following command to remove **`tomcatjss`**:

```
# yum remove tomcatjss
```

2.2. ZIP INSTALLATION

Ensure that all of the prerequisites are met before installing JBoss Web Server.

2.2.1. Downloading and Extracting JBoss Web Server

To install JBoss Web Server, download and extract the installation ZIP files.

1. Open a browser and log in to the Red Hat Customer Portal .

2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:
   - The Red Hat JBoss Web Server 5.0 Application Server (**`jws-application-servers-5.0.0.zip`**).
   - The Red Hat JBoss Web Server 5.0 Native Components for RHEL (**`jws-application-servers-5.0.0-<platform>-<architecture>.zip`**).
6. Unzip the downloaded ZIP files to your installation directory.
   For example:

```
# unzip jws-application-server-5.0.0.zip -d /opt/
# unzip -o jws-application-server-
5.0.0-<platform>-<architecture>.zip -d /opt/
```

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as ***JWS_HOME***.

2.2.2. Managing JBoss Web Server on Red Hat Enterprise Linux

There is three supported methods for running and managing Red Hat JBoss Web Server on Red Hat Enterprise Linux:

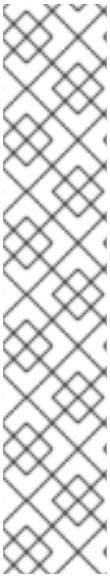
- using a system daemon
- on a command line
- using the Apache Common Daemon: Jsvc

The recommended method for managing the JBoss Web Server is using a system daemon.

2.2.2.1. Managing JBoss Web Server using a system daemon for .zip installations on Red Hat Enterprise Linux

Using the JBoss Web Server with a system daemon provides a method of starting the JBoss Web Server services at system boot. The system daemon also provides start, stop and status check functions.

The default system daemon for Red Hat Enterprise Linux 7 is systemd and for Red Hat Enterprise Linux 6 the default is SysV.



NOTE

To determine which system daemon is running, issue **`ps -p 1 -o comm=`**.

- For systemd:

```
$ ps -p 1 -o comm=

systemd
```

- For SysV:

```
$ ps -p 1 -o comm=

init
```

2.2.2.1.1. Setting up and using the JBoss Web Server with SysV

Prerequisites

- The **`redhat-lsb-core`** package. To install, run: **`yum install redhat-lsb-core`**

Setting up the JBoss Web Server for SysV
As the root user, execute the **`.postinstall.sysv`** script:

```
# cd JWS_HOME/tomcat
# sh .postinstall.sysv
```

Controlling the JBoss Web Server with SysV
SysV commands can only be issued by the root user.

- To enable the JBoss Web Server services to start at boot using SysV:

```
# chkconfig jws5-tomcat on
```

- To start the JBoss Web Server using SysV:

```
# service jws5-tomcat start
```

- To stop the JBoss Web Server using SysV:

```
# service jws5-tomcat stop
```

- To verify the status of the JBoss Web Server using SysV (the **`status`** operation can be executed by any user):

```
$ service jws5-tomcat status
```

For more information on using SysV, see: Red Hat Enterprise Linux 6 Deployment Guide: Running Services

2.2.2.1.2. Setting up and using the JBoss Web Server with systemd

Setting up the JBoss Web Server for systemd
As the root user, execute the **.postinstall.systemd** script:

```
# cd JWS_HOME/tomcat
# sh .postinstall.systemd
```

Controlling the JBoss Web Server with systemd
Systemd commands can only be issued by the root user.

- To enable the JBoss Web Server services to start at boot using systemd:

```
# systemctl enable jws5-tomcat.service
```

- To start the JBoss Web Server using systemd:

```
# systemctl start jws5-tomcat.service
```

- To stop the JBoss Web Server using systemd:

```
# systemctl stop jws5-tomcat.service
```

- To verify the status of the JBoss Web Server using systemd (the **status** operation can be executed by any user):

```
# systemctl status jws5-tomcat.service
```

For more information on using systemd, see: Red Hat Enterprise Linux 7 System Administrator's Guide: Managing System Services

2.2.2.2. Managing JBoss Web Server on a command line

2.2.2.2.1. Configuring the JBoss Web Server Installation

NOTE

The following configuration steps are performed by the **.postinstall.sysv** script and the **.postinstall.systemd** script described in Managing JBoss Web Server using a system daemon for .zip installations on Red Hat Enterprise Linux

Some configuration is required before running JBoss Web Server. This section includes the following configuration procedures:

- Setting the JAVA_HOME Environment Variable.
- Creating the tomcat user for simple and secure user management: Creating a Tomcat User.
- Grant the tomcat user access to the JBoss Web Server by moving the ownership of tomcat directory to the tomcat user.

Setting the JAVA_HOME Environment Variable
You must set the **JAVA_HOME** environment variable for Tomcat before running JBoss Web Server.

In the **bin** directory of Tomcat (***JWS_HOME*/tomcat/bin**), create a file named **setenv.sh**, and insert the **JAVA_HOME** path definition.

For example: **export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk.x86_64**

Creating a Tomcat User
Follow this procedure to create the **tomcat** user and its parent group:

1. In a shell prompt as the root user, change directory to ***JWS_HOME***.
2. Run the following command to create the **tomcat** user group:

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

3. Run the following command to create the **tomcat** user in the **tomcat** user group:

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

Move the ownership of tomcat directory to the tomcat user

1. From ***JWS_HOME***, run the following command to assign the ownership of the Tomcat directories to the **tomcat** user to allow the user to run the Tomcat service:

```
# chown -R tomcat:tomcat tomcat/
```

   You can use **ls -l** to verify that the **tomcat** user is the owner of the directory.

2. Ensure that the **tomcat** user has execute permissions to all parent directories. For example:

```
# chmod -R u+X tomcat/
```

2.2.2.2.2. Starting JBoss Web Server

Run the following command as the **tomcat** user:

```
$ sh JWS_HOME/tomcat/bin/startup.sh
```

2.2.2.2.3. Stopping JBoss Web Server

To stop Tomcat, run the following command as the **tomcat** user:

```
$ sh JWS_HOME/tomcat/bin/shutdown.sh
```

2.3. RPM INSTALLATION

Installing JBoss Web Server from RPM packages installs Tomcat as service, and installs its resources into absolute paths. The RPM installation option is only available for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

RPM installation packages for JBoss Web Server are available from Red Hat Subscription Management.

2.3.1. Installing JBoss Web Server from RPM packages

Prerequisites

- Install a Java Development Kit (JDK) .
- Ensure that the tomcatjss package is removed .

Before downloading and installing the RPM packages, you must register your system with Red Hat Subscription Management and subscribe to the respective Content Delivery Network (CDN) repositories.

For information on registering Red Hat Enterprise Linux, see Configuring the Subscription Service for Red Hat Enterprise Linux 6 or The Subscription Manager for Red Hat Enterprise Linux 7 .

Attaching subscriptions to Red Hat Enterprise Linux (if required)

If the system does not have a subscription attached that provides JBoss Web Server:

1. Log in to the Red Hat Subscription Manager .
2. Click on the **Systems** tab.
3. Click on the **Name** of the system to add the subscription to.
4. Change from the **Details** tab to the **Subscriptions** tab, then click **`Attach Subscriptions`**.
5. Select the check box beside the subscription to attach, then click **`Attach Subscriptions`**.

NOTE

To verify that a subscription provides the required CDN repositories:

1. Log in to: https://access.redhat.com/management/subscriptions.
2. Click the **`Subscription Name`**.
3. Under **Products Provided**, you require:
   - JBoss Enterprise Web Server.
   - Red Hat JBoss Core Services.

Installing JBoss Web Server from RPM packages using YUM

1. On a command line, subscribe to the JBoss Web Server CDN repositories for your operating system version using **`subscription-manager`**:

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

   - For Red Hat Enterprise Linux 6:
     - jws-5-for-rhel-6-server-rpms
     - jb-coreservices-1-for-rhel-6-server-rpms
   - For Red Hat Enterprise Linux 7:

- jws-5-for-rhel-7-server-rpms
- jb-coreservices-1-for-rhel-7-server-rpms

2. Issue the following command as the root user to install JBoss Web Server:

```
# yum groupinstall jws5
```

NOTE

- Although not recommended, instead of using the group install, you can install each of the packages and their dependencies individually.
- The Red Hat JBoss Core Services repositories above are required for the installation of JBoss Web Server.

2.3.2. Starting JBoss Web Server

- In a shell prompt as the root user, start the Tomcat service.
  - For Red Hat Enterprise Linux 6:

```
# service jws5-tomcat start
```

  - For Red Hat Enterprise Linux 7:

```
# systemctl start jws5-tomcat.service
```

This is the only supported method of starting JBoss Web Server for an RPM installation.

- To verify that Tomcat is running, the output of the service **`status`** command should be reviewed. This can be executed as any user.
  - For Red Hat Enterprise Linux 6:

```
# service jws5-tomcat status
```

  - For Red Hat Enterprise Linux 7:

```
# systemctl status jws5-tomcat.service
```

2.3.3. Stopping JBoss Web Server

- In a shell prompt as the root user, stop the Tomcat service.
  - For Red Hat Enterprise Linux 6:

```
# service jws5-tomcat stop
```

  - For Red Hat Enterprise Linux 7:

```
# systemctl stop jws5-tomcat.service
```

- To verify that Tomcat is no longer running, the output of the service **status** command should be reviewed. This can be executed as any user.
  - For Red Hat Enterprise Linux 6:

```
# service jws5-tomcat status
```

  - For Red Hat Enterprise Linux 7:

```
# systemctl status jws5-tomcat.service
```

2.3.4. Configuring JBoss Web Server Services to Start at Boot

Use the following commands to enable the JBoss Web Server services to start at boot.

- For Red Hat Enterprise Linux 6:

```
# chkconfig jws5-tomcat on
```

- For Red Hat Enterprise Linux 7:

```
# systemctl enable jws5-tomcat.service
```

2.4. SELINUX POLICIES

2.4.1. SELinux Policy Information

The following table contains information about the SELinux policies provided in the jws5-tomcat-selinux packages.

Table 2.1. RPMs and Default SELinux Policies

Name	Port Information	Policy Information
jws5_tomcat	Four ports in **http_port_t** (TCP ports **8080**, **8005**, **8009**, and **8443**) to allow the tomcat process to use them.	The jws5_tomcat policy is installed, which sets the appropriate SELinux domain for the process when Tomcat executes. It also sets the appropriate contexts to allow tomcat to write to **/var/opt/rh/jws5/lib/tomcat**, **/var/opt/rh/jws5/log/tomcat**, **/var/opt/rh/jws5/cache/tomcat** and **/var/opt/rh/jws5/run/tomcat.pid**.

For more information about using SELinux and other Red Hat Enterprise Linux security information, see the *Red Hat Enterprise Linux Security Guide*

2.4.2. SELinux Policies for an RPM Installation

SELinux policies for JBoss Web Server are provided by the **jws5-tomcat-selinux** package. These packages are available in the JWS channel.

To enable SELinux policies for JBoss Web Server 5.0, install the **jws5-tomcat-selinux** package.

2.4.3. SELinux Policies for a ZIP Installation

In this release, SELinux policies are provided in the ZIP packages. The SELinux security model is enforced by the kernel and ensures applications have limited access to resources such as file system locations and ports. This helps ensure that the errant processes (either compromised or poorly configured) are restricted and in some cases prevented from running.

The **.postinstall.selinux** file is included in the **tomcat** folder of **jws-application-server-5.0.0-*<platform>*-*<architecture>*.zip**. If required, you can run the **.postinstall.selinux** script.

To install the SELinux policies using ZIP:

1. Install the **selinux-policy-devel** package:

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

2. Execute the **.postinstall.selinux** script:

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

3. Make and install the SELinux module:

```
cd selinux
make -f /usr/share/selinux/devel/Makefile
semodule -i jws5-tomcat.pp
```

4. Apply the SELinux contexts for JBoss Web Server:

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

5. Add access permissions to the required ports for JBoss Web Server. The JBoss Web Server has access to ports **8080**, **8009**, **8443** and **8005** on Red Hat Enterprise Linux 7 systems. When additional ports are required for JBoss Web Server, use the **semanage** command to provide the necessary permissions, replacing the port number with the port required:

```
semanage port -a -t http_port_t -p tcp <port>
```

NOTE

The JBoss Web Server on Red Hat Enterprise Linux 6 systems has access to the same ports as Red Hat Enterprise Linux 7 systems, with the exception of port **8005**. To grant the JBoss Web Server access to this port on a Red Hat Enterprise Linux 6 system, as the root user, issue:

```
semanage port -a -t http_port_t -p tcp 8005
```

6. Start the Tomcat service:

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

7. Check the context of the running process expecting **jws5_tomcat**:

```
ps -eo pid,user,label,args | grep jws5_tomcat | head -n1
```

8. To verify the contexts of the Tomcat directories, for example:

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

NOTE

By default, the SElinux policy provided is not active and the Tomcat processes run in the **unconfined_java_t** domain. This domain does not confine the processes, and it is recommended that you undertake the following security precautions if you chose not to enable the SElinux policy provided:

- Restrict file access for the **tomcat** user to only the files and directories that are necessary to the JBoss Web Server runtime.
- Do not run Tomcat as the **root** user.

CHAPTER 3. INSTALLING JBOSS WEB SERVER ON MICROSOFT WINDOWS

3.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)

Before installing JBoss Web Server on Microsoft Windows, you must first install a supported Java Development Kit (JDK).

For a list of supported configurations, see the Red Hat Customer Portal article: JBoss Web Server 5 Supported Configurations.

NOTE

For instructions on installing the IBM JDK, visit: https://www.ibm.com/developerworks/java/jdk/

To install the Oracle Java Development Kit:

1. Download the Oracle JDK for your operating system and architecture. You can download the JDK installation file from the Oracle website: http://www.oracle.com/technetwork/java/javase/downloads/index.html.
2. Double-click the downloaded file to start the installation.
3. Proceed as instructed in the installation window.

3.2. DOWNLOADING AND EXTRACTING JBOSS WEB SERVER

To install JBoss Web Server, download and extract the installation ZIP files.

1. Open a browser and log in to the Red Hat Customer Portal .
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:
   - The Red Hat JBoss Web Server 5.0 Application Server (**`jws-application-servers-5.0.0.zip`**).
   - The Red Hat JBoss Web Server 5.0 Native Components for Windows Server (**`jws-application-servers-5.0.0-<platform>-<architecture>.zip`**).
6. Unzip the downloaded ZIP files to your installation directory.

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as ***`JWS_HOME`***.

3.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION

Some configuration is required before running JBoss Web Server. This section includes the following configuration procedures:

- Setting Environment Variables
- Installing the Tomcat Service
- Configuring Folder Permissions for the JBoss Web Server Services

Setting Environment Variables

1. Log in to an account with local administrator permissions.
2. Go to **Control Panel** → **System**.
3. Click on the **Advanced** tab.
4. Click the **Environment Variables** button.
5. Click the **New** button for **System Variables**.
6. For **JAVA_HOME**, **TMP**, and **TEMP**, enter the appropriate name-value pairs for your system.
7. For the SSL Connector to work, you will also need to add ***JWS_HOME*\bin** to the **PATH** environment variable of the user that the services will run under. This user is **SYSTEM** by default.

Installing the Tomcat Service

1. Open a command prompt with administrator privileges and change to the **bin** folder for your Tomcat version:

```
cd /D "JWS_HOME\tomcat\bin"
```

2. Install the Tomcat service with the following command:

```
call service.bat install
```

Configuring Folder Permissions for the JBoss Web Server Services

Follow this procedure to ensure that the account used to run the services has full control over the ***JWS_HOME*** folder and all of its subfolders:

1. Right-click the ***JWS_HOME*** folder and click **Properties**.
2. Select the **Security** tab.
3. Click the **Edit** button.
4. Click the **Add** button.
5. In the text box, enter **LOCAL SERVICE**.
6. Select the **Full Control** check box for the **LOCAL SERVICE** account.
7. Click **OK**.

8. Click the **Advanced** button.
9. Inside the **Advanced Security Settings** dialog, select **LOCAL SERVICE** and click **Edit**.
10. Select the check box next to the **Replace all existing inheritable permissions on all descendants with inheritable permissions from this object** option.
11. Click **OK** through all the open folder property windows to apply the settings.

3.4. STARTING JBOSS WEB SERVER

You can start the JBoss Web Server from a command prompt, or with the Computer Management tool.

Starting JBoss Web Server from a Command Prompt

1. Open a command prompt with administrator privileges.
2. Start the Tomcat service:

```
net start tomcat9
```

Starting JBoss Web Server from the Computer Management Tool

1. Go to **Start** → **Administrative Tools** → **Services**.
2. In the **Services** list, right-click the name of the service (**Tomcat9**) and click **Start**.

NOTE

Some third-party applications add libraries to the system directory in Windows. These take precedence over Tomcat libraries when looked-up. This means that if those third-party libraries have the same name as the those used by Tomcat native libraries, they are loaded instead of the libraries distributed with JBoss Web Server.

In this situation, Tomcat may not start, and does not log any error messages in the Windows Event Log, or Tomcat log files. Errors can only be seen by using **catalina.bat run**.

If this behavior occurs, inspect the contents of the **C:\windows\System32** directory and other **PATH** directories, and ensure that there are no DLLs conflicting with those delivered with JBoss Web Server. In particular, look for **libeay32.dll**, **ssleay32.dll**, and **libssl32.dll**.

3.5. STOPPING JBOSS WEB SERVER

You can stop the JBoss Web Server from a command prompt, or with the Computer Management tool.

Stopping JBoss Web Server from a Command Prompt

1. Open a command prompt with administrator privileges.
2. Stop the Tomcat service:

```
net stop tomcat9
```

-

Stopping JBoss Web Server from the Computer Management Tool

1. Go to **Start** → **Administrative Tools** → **Services**.
2. In the **Services** list, right-click the name of the service (**`Tomcat9`**) and click **Stop**.

CHAPTER 4. INSTALLING JBOSS WEB SERVER ON SOLARIS

4.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)

Before installing JBoss Web Server on Solaris, you must first install a supported Java Development Kit (JDK).

For a list of supported configurations, see the Red Hat Customer Portal article: JBoss Web Server 5 Supported Configurations.

Installing a Java Development Kit (JDK)

Install the Oracle JDK on a command line as the root user:

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

Where <version> is the version of the JDK to install, such as **jdk-8**

Alternative: Download and Install a Java Development Kit on Solaris

1. Download the Oracle JDK for your operating system and architecture. You can download the JDK installation file from the Oracle website: http://www.oracle.com/technetwork/java/javase/downloads/index.html.
2. Run the JDK installation file.
3. Open **/usr** at a shell prompt, and run the following command to display the current Java symbolic link:

   ```
   ls -lad java
   ```

4. Remove the link:

   ```
   rm java
   ```

5. Create a new Java symbolic link to the newly installed JDK:

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

4.2. DOWNLOADING AND EXTRACTING JBOSS WEB SERVER

To install JBoss Web Server, download and extract the installation ZIP files.

1. Open a browser and log in to the Red Hat Customer Portal .
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:

- The Red Hat JBoss Web Server 5.0 Application Server (**jws-application-servers-5.0.0.zip**).
- The Red Hat JBoss Web Server 5.0 Native Components for Solaris (**jws-application-servers-5.0.0-*<platform>*-*<architecture>*.zip**).

6. Unzip the downloaded ZIP files to your installation directory.
   For example:

```
# unzip jws-application-server-5.0.0.zip -d /opt/
# unzip -o jws-application-server-
5.0.0-<platform>-<architecture>.zip -d /opt/
```

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as ***JWS_HOME***.

4.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION

Some configuration is required before running JBoss Web Server. This section includes the following configuration procedures:

- Running the Post-Installation Scripts

Running the Post-Installation Scripts

1. Open a shell prompt, and change directory to ***JWS_HOME*/tomcat/**.
2. As the root user, run the post-installation scripts:

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

   The post-installation script:

   - Sets the **JAVA_HOME** environment variable.
   - Creates the **tomcat** user.
   - Creates the **tomcat** user group.

4.4. MANAGING JBOSS WEB SERVER ON SOLARIS USING A SYSTEM DAEMON

Managing a JBoss Web Server with a system daemon allows the web server to restart automatically after a reboot or power outage without user intervention. The system daemon also ensures that the JBoss Web Server is running as the **tomcat** user and provides some basic logging.

SysV and the Service Management Facility (SMF) are the two system daemons found on Solaris.

For SysV users:

- To add JBoss Web Server 5.0 to SysV, see: Enabling JBoss Web Server 5.0 management using SysV on Solaris

- To review basic management commands for JBoss Web Server on SysV, see: Managing the JBoss Web Server using SysV

For SMF users:

- To add JBoss Web Server 5.0 to the SMF, see: Enabling JBoss Web Server 5.0 management using the Solaris Service Management Facility (SMF)
- To review basic management commands for JBoss Web Server on SysV, see: Managing the JBoss Web Server using the Service Management Facility (SMF)

4.4.1. Enabling JBoss Web Server 5.0 management using SysV on Solaris

Using the JBoss Web Server with SysV provides start, stop and status check functions for the tomcat service. This procedure shows how to add JBoss Web Server 5.0 to SysV.

Prerequisites

- Root user access.
- The Red Hat JBoss Web Server 5.0 installed on Oracle Solaris.
- SysV is the default system daemon.
- The postinstall script (`.postinstall.tomcat`) has been executed as the root user.

Procedure

1. On a command line, change to the ***`JWS_HOME`*`/tomcat/`** directory.
2. Copy the control script from the ***`JWS_HOME`*`/tomcat/services`** directory to **`/etc/init.d/`**:

```
# cp services/jws5-tomcat.init /etc/init.d/jws5-tomcat
```

3. Issue the following command to set the control script as executable:

```
# chmod +x /etc/init.d/jws5-tomcat
```

Results
The JBoss Web Server 5.0 instance of tomcat should now be present in SysV. For example:

```
$ service jws5-tomcat status

jws5-tomcat is stopped
```

Next Steps
For information on the basic commands for controlling JBoss Web Server 5.0 using SysV, see: Section 4.4.2, “Managing the JBoss Web Server using SysV” .

Additional Resources
For information on using SysV, see: Red Hat Enterprise Linux 6 Deployment Guide: Running Services

4.4.2. Managing the JBoss Web Server using SysV

SysV commands can only be issued by the root user.

- To enable the JBoss Web Server services to start at boot using SysV:

```
# chkconfig jws5-tomcat on
```

- To start the JBoss Web Server using SysV:

```
# service jws5-tomcat start
```

- To stop the JBoss Web Server using SysV:

```
# service jws5-tomcat stop
```

- To verify the status of the JBoss Web Server using SysV (the **status** operation can be executed by any user):

```
$ service jws5-tomcat status
```

For information on using SysV, see: Red Hat Enterprise Linux 6 Deployment Guide: Running Services

4.4.3. Enabling JBoss Web Server 5.0 management using the Solaris Service Management Facility (SMF)

Using the JBoss Web Server with Service Management Facility (SMF) provides start, stop and status check functions for the tomcat service. This procedure shows how to add JBoss Web Server 5.0 to the SMF.

NOTE

The service runlevel of **3** in the following procedure is a recommended value. For more information on Solaris runlevels, see: Oracle Solaris Administration: Common Tasks - Run Levels

Prerequisites

- Root user access.
- The Red Hat JBoss Web Server 5.0 installed on Oracle Solaris 10 or newer.
- The Oracle Solaris Service Management Facility (SMF) is the default system daemon.
- The postinstall script (**.postinstall.tomcat**) has been executed as the root user.

Procedure

1. On a command line, change to the ***JWS_HOME*/tomcat/** directory.
2. As the root user, import the control script into the SMF using the **svcbundle** command:

```
# svcbundle -i -s rc-script=JWS_HOME/tomcat/services/jws5-
tomcat.init:3 -s service-name=jws5-tomcat
```

Results
The JBoss Web Server 5.0 instance of tomcat should now be present in the SMF. For example:

```
# svcs -l jws5-tomcat

fmri         svc:/jws5-tomcat:default
enabled      true
state        online
next_state   none
state_time   July 25, 2018 01:59:29 AM AEST
logfile      /var/svc/log/jws5-tomcat:default.log
restarter    svc:/system/svc/restarter:default
manifest     /lib/svc/manifest/site/jws5-tomcat.xml
dependency   require_all/none svc:/milestone/multi-user (online)
```

Next Steps
For information on the basic commands for controlling JBoss Web Server 5.0 using the SMF, see: Section 4.4.4, “Managing the JBoss Web Server using the Service Management Facility (SMF)” .

Additional Resources

- For information on the Oracle Solaris Service Management Facility (SMF), see: Oracle: Introducing the Basics of Service Management Facility (SMF) on Oracle Solaris 11.
- For information on the basic usage of the SMF, see: Oracle Solaris Administration: Common Tasks - Managing SMF Services.
- For information on using the **svcbundle** command to migrate control scripts to the SMF, see: Oracle: Developing System Services in Oracle® Solaris 11.3 - Converting a Run Control Script to an SMF Service.

4.4.4. Managing the JBoss Web Server using the Service Management Facility (SMF)

The Service Management Facility (SMF) is the default system daemon for Oracle Solaris 10 or higher.

SMF commands can only be issued by the root user.

- To enable the JBoss Web Server services to start at boot using the SMF:

```
# svcadm enable jws5-tomcat
```

- To start the JBoss Web Server using the SMF:

```
# svcadm enable -t jws5-tomcat
```

- To stop the JBoss Web Server using the SMF:

```
# svcadm disable -t jws5-tomcat
```

- To verify the status of the JBoss Web Server using the SMF:

```
$ svcs -l jws5-tomcat
```

Additional Resources

- For information on the Oracle Solaris Service Management Facility (SMF), see: Oracle: Introducing the Basics of Service Management Facility (SMF) on Oracle Solaris 11.
- For information on the basic usage of the SMF, see: Oracle Solaris Administration: Common Tasks - Managing SMF Services.

4.5. MANUALLY MANAGING JBOSS WEB SERVER ON SOLARIS

To manually start or stop JBoss Web Server on Solaris using JSVC:

- Section 4.5.1, “Starting JBoss Web Server”
- Section 4.5.2, “Stopping JBoss Web Server”

4.5.1. Starting JBoss Web Server

To start JBoss Web Server: as the root user, run the following command:

```
# sh JWS_HOME/tomcat/bin/daemon.sh start
```

IMPORTANT

Although there are multiple methods of starting Tomcat, it is recommended that you use the **daemon.sh** script. To start Tomcat as a service using Jsvc, see the Jsvc chapter.

4.5.2. Stopping JBoss Web Server

To stop JBoss Web Server: as the root user, run the following command:

```
# sh JWS_HOME/tomcat/bin/daemon.sh stop
```

CHAPTER 5. USING JSVC TO START TOMCAT

Jsvc is a set of libraries and applications that facilitates running Java applications on Linux, UNIX, and similar operating systems. Using Jsvc with JBoss Web Server allows Tomcat to switch identities. Using Jsvc, Tomcat can perform root-level operations and then revert to a non-privileged user. Jsvc is primarily used for running Tomcat as a service.

Jsvc files are available at the following locations:

- ***JWS_HOME*/tomcat/bin/jsvc**
- ***JWS_HOME*/jbcs-jsvc-1.0/sbin/jsvc**

NOTE

***JWS_HOME*/bin/jsvc** is a symlink to ***JWS_HOME*/jbcs-jsvc-1.0/sbin/jsvc**.

5.1. STARTING TOMCAT USING JSVC

Start Tomcat Using Jsvc

Run the following command to start Tomcat using Jsvc:

```
JWS_HOME/tomcat/bin/daemon.sh start
```

5.2. STOPPING TOMCAT USING JSVC

Stop Tomcat Using Jsvc

Run the following command to stop Tomcat that was started using Jsvc:

```
JWS_HOME/tomcat/bin/daemon.sh stop
```

5.3. JSVC PARAMETERS

The following parameters can be configured when running the **daemon.sh** script:

Table 5.1. daemon.sh Startup Parameters

Parameter Name	Environment Variable	Default Value	Description
--java-home	**JAVA_HOME**	Based on the value of the **PATH** variable.	The Java home directory location.
--catalina-home	**CATALINA_HOME**	Determined by the location of the script.	The Tomcat installation directory location.

Parameter Name	Environment Variable	Default Value	Description
--catalina-base	**CATALINA_BASE**	Based on the value of the **PATH** variable.	The directory that contains the specific configuration and setup information if multiple servers are using the same installation.
--catalina-pid	-	**$CATALINA_BASE/logs/catalina-daemon.pid**	The file where the process ID (PID) for the running instance of Tomcat is stored.
--tomcat-user	-	**tomcat**	The user Tomcat uses.
--service-start-wait-time	-		This is a wrapper to the **--wait** parameter. The **--wait** parameter accepts values in seconds.

CHAPTER 6. HIBERNATE ON JBOSS WEB SERVER

Hibernate is an object-relational mapping framework. It is provided by the JBoss Web Server Maven Repository (**jboss-web-server-5.0.0-maven-repository.zip**). This packaged version is used on all supported platforms.

Hibernate is used in the same way it is used for a regular Tomcat installation: the Hibernate JAR files are added into a deployment WAR file. Tomcat provides a default connection pooling mechanism, which is defined in **context.xml**. However, **persistence.xml** and **web.xml** are also required. The example below shows a configuration with the Tomcat connection pooling mechanism.

- **/META-INF/context.xml** defines the connection pools Tomcat should create.

  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** is a JPA configuration file. It defines how the application configures Hibernate to consume connections from the Tomcat pool. If you are using the Hibernate API directly, use a similar configuration to that shown in **hibernate.cfg.xml**.

  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** is a regular web application deployment file, which instructs Tomcat which datasource to consume. In the example below, the datasource is **jdbc/DsWebAppDB**.

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>
```

For details, see the Hibernate documentation for JBoss Web Server.

CHAPTER 7. ENABLING HTTP/2 FOR THE RED HAT JBOSS WEB SERVER

The Hypertext Transfer Protocols are standard methods of transmitting data between applications (such as servers and browsers) over the internet. HTTP/2 improves on HTTP/1.1 by providing enhancements such as:

- header compression - reducing the size of the header transmitted by omitting implied information, and
- multiple requests and responses over a single connection - using binary framing to break down response messages, as opposed to textual framing.

Using HTTP/2 with the Red Hat JBoss Web Server:

- **is supported** for encrypted connections over TLS (**h2**).
- **is not supported** for unencrypted connections over TCP (**h2c**).

Prerequisites

- Root user access (Red Hat Enterprise Linux and Solaris systems), or
- Administrative access (Windows Server).
- Red Hat JBoss Web Server 5.0 or higher
- The following operating system native libraries (provided by **`jws-application-server-5.0.0-<platform>-<architecture>.zip`** where available).
  - Tomcat Native, for example:

    ```
    jws-5.0/tomcat/lib/libtcnative-1.so
    ```

  - Apache Portable Runtime (APR):

    ```
    jws-5.0/tomcat/lib/libapr-1.so.0.6.3
    ```

    Where the APR libraries are provided by **`jws-application-server-5.0.0-<platform>-<architecture>.zip`** for Red Hat Enterprise Linux, the libraries will be a symbolic link to:

    ```
    jws-5.0/jbcs-apr-1.6.3/lib64/libapr-1.so.0.6.3
    ```

  - OpenSSL, for example:

    ```
    jws-5.0/tomcat/lib/libcrypto.so.1.0.2n
    jws-5.0/tomcat/lib/libssl.so.1.0.2n
    ```

    Where the OpenSSL libraries are provided by **`jws-application-server-5.0.0-<platform>-<architecture>.zip`** for Red Hat Enterprise Linux, the libraries will be symbolic links to:

```
jws-5.0/jbcs-openssl-1.0.2n/openssl/lib64/libcrypto.so.1.0.2n
jws-5.0/jbcs-openssl-1.0.2n/openssl/lib64/libssl.so.1.0.2n
```

- A connector that supports the HTTP/2 protocol with SSL enabled. For JBoss Web Server 5.0, the connectors with HTTP/2 protocol support are:
  - The APR Native connector (APR)
  - The NIO connector with JSSE + OpenSSL (JSSE)
  - The NIO2 connector with JSSE + OpenSSL (JSSE)

Procedure

Enable HTTP/2 for a connector:

1. Add the HTTP/2 upgrade protocol (**`<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />`**) to the connector in the server configuration ***`JWS_HOME`*`/tomcat/conf/server.xml`**.
   For example:

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true">
    <UpgradeProtocol
className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
        <Certificate certificateKeystoreFile="/KeyStore.jks"
                     certificateKeystorePassword="changeit"
                     type="RSA" />
    </SSLHostConfig>
</Connector>
```

   `server.xml` contains an example connector definition for the APR protocol with the upgrade protocol to HTTP/2:

```
<Connector port="8443"
           protocol="org.apache.coyote.http11.Http11AprProtocol"
           maxThreads="150" SSLEnabled="true" >
    <UpgradeProtocol
className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
        <Certificate certificateKeyFile="conf/localhost-rsa-
key.pem"
                     certificateFile="conf/localhost-rsa-cert.pem"
                     certificateChainFile="conf/localhost-rsa-
chain.pem"
                     type="RSA" />
    </SSLHostConfig>
</Connector>
```

2. Restart the Red Hat JBoss Web Server as the root user, to apply the changed configuration.
   a. For SysV (Red Hat Enterprise Linux 6) users:

```
# service jws5-tomcat restart
```

b. For systemd (Red Hat Enterprise Linux 7) users:

```
# systemctl restart jws5-tomcat.service
```

c. For Red Hat Enterprise Linux users running Red Hat JBoss Web Server using **`startup.sh`**:

```
# JWS_HOME/sbin/shudown.sh
# JWS_HOME/sbin/startup.sh
```

d. For Solaris users:

```
# sh JWS_HOME/tomcat/bin/daemon.sh stop
# sh JWS_HOME/tomcat/bin/daemon.sh start
```

e. For Windows Server users:

```
# net restart tomcat9
```

Next Steps

Verify that HTTP/2 is enabled by reviewing the Red Hat JBoss Web Server logs or by using the **`curl`** command:

- Check the console output log (***`JWS_HOME`*`/tomcat/logs/catalina.out`**) to verify that the "connector has been configured to support negotiation to [h2]":

```
$ cat JWS_HOME/tomcat/logs/catalina.out | grep 'h2'

06-Apr-2018 04:49:26.201 INFO [main]
org.apache.coyote.http11.AbstractHttp11Protocol.configureUpgradeProt
ocol The ["https-openssl-apr-8443"] connector has been configured to
support negotiation to [h2] via ALPN
```

- Or verify using **`curl`** (for versions of **`curl`** that support **`HTTP2`**):

NOTE

To check **`curl`** for HTTP/2 support:

```
$ curl -V

curl 7.55.1 (x86_64-redhat-linux-gnu) ...
Release-Date: 2017-08-14
Protocols: dict file ftp ftps gopher http https ...
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos
SPNEGO NTLM NTLM_WB SSL libz TLS-SRP HTTP2 UnixSockets
HTTPS-proxy Metalink PSL
```

  - For example, when the HTTP/2 protocol is inactive:

```
$ curl -I http://<JBoss_Web_Server>:8080/
```

```
HTTP/1.1 200
...
```

- But if the HTTP/2 protocol is active, **curl** returns:

```
$ curl -I https://<JBoss_Web_Server>:8443/

HTTP/2 200
...
```

Where ***<JBoss_Web_Server>*** is the URI of the modified connector (such as **example.com**), and the port number is dependent on your configuration.

Additional Resources

- For additional information on using HTTP/2, see: Apache Tomcat 9 Configuration Reference: The HTTP Connector - HTTP/2 Support.
- For information on the HTTP/2 Upgrade Protocol and the supported attributes, see: Apache Tomcat 9 Configuration Reference: The HTTP2 Upgrade Protocol.
- The proposed internet standard for HTTP/2: IETF: RFC 7540 - Hypertext Transfer Protocol Version 2 (HTTP/2)

CHAPTER 8. USING A PASSWORD VAULT WITH RED HAT JBOSS WEB SERVER 5.0

8.1. USING A PASSWORD VAULT WITH RED HAT JBOSS WEB SERVER 5.0

A password vault is used to mask passwords and other sensitive strings, and store them in an encrypted Java keystore. This allows you to eliminate storing clear-text passwords in your Tomcat configuration files, as Tomcat can lookup passwords and other sensitive strings from a keystore using the vault.

Installing the JBoss Web Server password vault from .zip archive
The tomcat password vault is pre-installed by the **`jws-application-server-5.0.0.zip`** file. The password vault can be used once configured and is located at: ***`JWS_HOME`*`/tomcat/lib/tomcat-vault.jar`**.

Installing the JBoss Web Server password vault on Red Hat Enterprise Linux using the YUM package manager
Where the JBoss Web Server has been installed from RPMs on Red Hat Enterprise Linux, install the password vault as the root user by executing:

```
yum install jws5-tomcat-vault
```

The examples and commands below use ***`JWS_HOME`*** as the JBoss Web Server installation directory. Replace ***`JWS_HOME`*** with the path to your JBoss Web Server installation. Also, the paths below use **`/`** for directory separators.

8.1.1. Enabling the Password Vault

1. Stop Tomcat if it is running.
2. Edit ***`JWS_HOME`*`/tomcat/conf/catalina.properties`**, and add the following line:

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

8.1.2. Creating a Java Keystore

To use a password vault, you must first create a Java keystore. You can do this using the **`keytool -genseckey`** command. For example:

```
$ 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
```

IMPORTANT

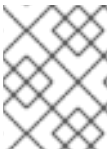
The values above are examples only. Replace them with values specific to your environment.

For an explanation of the parameters, use the **keytool -genseckey -help** command.

8.1.3. Storing the tomcat-vault vault.properties file outside of the JWS_HOME directory

The **vault.properties** file for the **tomcat-vault** can be stored outside of ***JWS_HOME*/tomcat/conf/** in a ***CATALINA_BASE*/conf/** directory (if set).

To set the *CATALINA_BASE* directory, follow the instructions in the section '**Advanced Configuration - Multiple Tomcat Instances**' in the Running The Apache Tomcat 9.0 Servlet/JSP Container document found on the Apache Tomcat Website.



NOTE

The default location for *CATALINA_BASE* is ***JWS_HOME*/tomcat/** also known as *CATALINA_HOME*.

For more information on setting *CATALINA_BASE*, see:

- Apache Tomcat 9: Introduction - Directories and Files
- Running The Apache Tomcat 9.0 Servlet/JSP Container: Advanced Configuration - Multiple Tomcat Instances

8.1.4. Initializing the Password Vault

The vault must be initialized before it can be used to store sensitive strings. This is done using the ***JWS_HOME*/tomcat/bin/tomcat-vault.sh** vault script. For Microsoft Windows, the script is **tomcat-vault.bat**.

The script can be run interactively or non-interactively. Below is an example of an interactive execution of the script to initialize a password vault, with the values shown below using the example keystore from the previous step.

8.1.4.1. Initializing the Vault for Apache Tomcat interactively

IMPORTANT

The values below are examples only. Replace them with values appropriate for your environment.

```
# JWS_HOME/tomcat/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

  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/
Enter Keystore URL: JWS_HOME/tomcat/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.keystore
KEYSTORE_PASSWORD=MASK-3CuP21KMHn7G6iH/A3YpM/
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=JWS_HOME/tomcat/
...
********************************************
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
```

Note the output for the Tomcat properties file, as you will need this to configure Tomcat to use the vault.

Configuring Tomcat to Use the Password Vault

In ***JWS_HOME*/tomcat/conf/**, create a file named **vault.properties** containing the vault configuration produced when initializing the vault. The values provided below use the example vault initialized in the previous steps.

NOTE

For **KEYSTORE_PASSWORD**, you must use the masked value that was generated when initializing the vault.

```
KEYSTORE_URL=JWS_HOME/tomcat/vault.keystore
KEYSTORE_PASSWORD=MASK-3CuP21KMHn7G6iH/A3YpM/
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=JWS_HOME/tomcat/
```

8.1.4.2. Initializing the Vault for Apache Tomcat non-interactively (silent setup)

The Vault for Apache Tomcat can be created non-interactively by providing the required input as arguments to the **`tomcat-vault.sh`** script. The **`vault.properties`** file is also created as output of the **`tomcat-vault.sh`** script when the **`-g, --generate-config`** option is used.

IMPORTANT

The values below are examples only. Replace them with values appropriate for your environment.

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

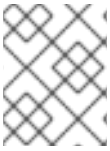
8.1.5. Storing a Sensitive String in the Password Vault

The vault script used in the previous steps is also used to store sensitive strings in the password vault. The script can be run interactively or non-interactively.

When adding a string to a password vault, the sensitive string needs a name that it will be referred by. For a password vault, this name is called an **`attribute name`**, and the password itself is called a **`secured attribute`**.

The example below demonstrates using the vault script non-interactively to store a password. It uses the vault that was initialized in the previous steps, and stores the sensitive string **P@SSW0#D** with the attribute name **`manager_password`**.

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



NOTE

You can optionally specify a vault block to store the password in. If you don't specify a block, one will be automatically created for you. In the above example, my_block is used.

8.1.6. Using a Stored Sensitive String in Your Tomcat Configuration

After storing a sensitive string in the password vault, you can refer to it in your configuration files by entering the stored string's attribute as **`${VAULT::block_name::attribute_name::}`**.

For example, to use the password stored in the previous steps, replace:

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

with:

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

As a result, only a reference to the password is visible in the Tomcat configuration file, and the actual password is only stored in the password vault.

APPENDIX A. JAVA IPV4/IPV6 PROPERTIES

Configuring Java Properties

In Java there are 2 properties that are used to configure IPv4 and IPv6. These are **`java.net.preferIPv4Stack`** and **`java.net.preferIPv6Addresses`**.

java.net.preferIPv4Stack (default: false)

If IPv6 is available then the underlying native socket, by default, is an IPv6 socket. This socket lets applications connect and accept connections from IPv4 and IPv6 hosts. If application use only IPv4 sockets, then set this property to **`true`**. However, it will not be possible for the application to communicate with IPv6 only hosts.

java.net.preferIPv6Addresses (default: false)

If a host has both IPv4 and IPv6 addresses, and IPv6 is available, then the default behavior is to use IPv4 addresses over IPv6. This allows backward compatibility. If applications that depend on an IPv4 address representation, for example: 192.168.1.1. Then, set this property to **`true`** to change the preference and use IPv6 addresses over IPv4 where possible.

To pass these properties to Tomcat, set **`CATALINA_OPTS`** in the ***`JWS_HOME`*`/tomcat/bin/setenv.*`** file.

NOTE

If the ***`JWS_HOME`*`/tomcat/bin/setenv.sh`** or ***`JWS_HOME`*`/tomcat/bin/setenv.bat`** file does not exist, then you need to create one.

On Linux:

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

On Windows:

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

Configuring Tomcat Bindings

The Tomcat bindings can be set in ***`JWS_HOME`*`/tomcat/conf/server.xml`** with the IPv6 address:

- Specify the Tomcat binding address:
  `<Server … address="`*`TOMCAT_BINDING_ADDRESS`*`">`
- Specify the HTTP connector address:
  `<Connector protocol="HTTP/1.1" … address="`*`HTTP_CONNECTOR_ADDRESS`*`">`
- Specify the AJP connector address:
  `<Connector protocol="AJP/1.3" … address="`*`AJP_CONNECTOR_ADDRESS`*`">`