

**Getting Started Guide**

# **JBoss Enterprise SOA Platform**

**4.2**



**ISBN:**

**Publication date: February, 2008**

## Getting Started Guide

---

Covering installation and initial configuration of JBoss Enterprise SOA Platform

---

# Getting Started Guide: JBoss Enterprise SOA Platform

Copyright © 2008 Red Hat, Inc.

## Legal Notice

1801 Varsity Drive  
Raleigh, NC27606-2072USAPhone: +1 919 754 3700  
Phone: 888 733 4281  
Fax: +1 919 754 3701  
PO Box 13588Research Triangle Park, NC27709USA

Copyright © 2007 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

---



---

|   |     |
|---|-----|
| Preface .....   | vii |
| 1. We Need Feedback .....   | vii |
| 1. Installing JBoss Enterprise SOA Platform .....                                   | 1   |
| 1. Prerequisites .....  | 1   |
| 1.1. Hardware and Operating System Requirements .....                               | 1   |
| 1.2. Configuring Your Java Environment .....  | 1   |
| 1.3. Installing the Sun JDK on Red Hat Enterprise Linux .....                       | 1   |
| 1.4. Installing and Configuring 32-bit Sun JDK 5.0 on Microsoft Windows .....       | 2   |
| 2. Upgrading or Migrating to JBoss Enterprise SOA Platform .....                    | 3   |
| 3. Installing the SOA Platform .....  | 3   |
| 2. Post-Installation: Configuring and starting the server .....                     | 5   |
| 1. Post-installation configuration .....  | 5   |
| 1.1. Changing the Database used by the platform .....                               | 5   |
| 1.2. Securing the server for production environments .....                          | 6   |
| 2. Starting JBoss Enterprise SOA Platform .....                                     | 8   |
| 2.1. Troubleshooting Start up .....   | 8   |
| 3. Running alongside an earlier version of JBoss Enterprise Application Platform .. | 9   |
| 3. Testing the Installation .....   | 11  |
| 1. Running the Helloworld Quickstart .....  | 11  |
| A. Revision History .....   | 13  |

---

---

## Preface

### 1. We Need Feedback

If you find a typographical error in the *SOAPGSG*, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in JIRA:

<http://jira.jboss.com/jira> against the Documentation component the *SOA Platform* project.

When submitting a bug report, be sure to mention the manual's identifier:

SOAP\_GSG

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.





# Installing JBoss Enterprise SOA Platform

In this chapter we cover installing JBoss Enterprise SOA Platform.

## 1. Prerequisites

### 1.1. Hardware and Operating System Requirements

For the latest information on supported Operating System / JVM combinations and supported Database platforms, please refer to <http://www.jboss.com/products/platforms/soa><sup>1</sup>.

You must have adequate disk space to install JDK and JBoss Enterprise SOA Platform. In addition to the JDK disk space requirement, approximately 800 MB is required for a full installation, and 220 MB is required for the standalone version. Before installing JBoss Enterprise SOA Platform you must also have a working installation of Java.

You must also have sufficient RAM in your machine for the JBoss Enterprise SOA Platform. You should have 768MB of physical RAM available for the server, in addition to the operating system and any other application requirements.

### 1.2. Configuring Your Java Environment

You need a working installation of **JDK 1.5** before you can install and run JBoss Enterprise SOA Platform. We will walk you through installing the 32-bit Sun JDK 5.0 on a Red Hat Enterprise Linux machine and Microsoft Windows 2003 machine.

### 1.3. Installing the Sun JDK on Red Hat Enterprise Linux

#### Procedure 1.1. Install Sun JDK on Red Hat Enterprise Linux

1. The Sun Java SDK is provided by the Red Hat Enterprise Linux 4 Extras channel for your Linux variant and architecture. The channel names are as follows:

```
rhel-<arch>-<variant>-4-extras
```

where:

```
arch= i386 or x86_64, and variant = as, or es
```

2. For Red Hat Enterprise Linux 5 the Java SDK is provided by the supplementary channel:

```
rhel-i386-server-supplementary-5 or rhel-x86_64-server-supplementary-5
```

---

<sup>1</sup> <http://www.jboss.com/products/platforms/soa>

3. Use `up2date` to install the packages `java-1.5.0-sun` and `java-1.5.0-sun-devel`.
4. You may wish to use `alternatives` to set the system default `java`, `javac` and `java_sdk_1.5.0`.

This is only needed if you want to use the SysV service script and/or want this installed SDK to be the default `java` and `javac` in the system. This choice can often be overridden by setting the `JAVA_HOME` environment variable.

The `alternatives` system allows different versions of Java, from different sources to co-exist on your system. You should make sure the desired one is selected so that the service script uses the one you want.

As root, issue the following command:

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

and make sure the desired one is selected (marked with a '+'), or select it by entering its number as prompted.

Make sure you do the same for `javac` and `java_sdk_1.5.0`. All should point to the same manufacturer and version.

## 1.4. Installing and Configuring 32-bit Sun JDK 5.0 on Microsoft Windows

### Procedure 1.2. Install Sun JDK on Microsoft Windows

1. Download the Sun JDK 5.0 (Java 2 Development Kit) from Sun's website:  
[http://java.sun.com/javase/downloads/index\\_jdk5.jsp](http://java.sun.com/javase/downloads/index_jdk5.jsp). Choose "JDK 5.0 Update <x>" (where x is the latest update number) for download and then select your Windows Platform to perform the installation.
2. Create an environment variable called `JAVA_HOME` that points to the JDK installation directory, for example: `C:\Program Files\Java\jdk1.5.0_11\`. To do this, open the Control Panel from the Start Menu, switch to Classic View if necessary, open the System Control Panel applet, select the Advanced Tab, and click on the Environment Variables button.

In order to run `java` from the command line add the `jre\bin` directory to your path, for example: `C:\Program Files\Java\jdk1.5.0_11\jre\bin`.

## 2. Upgrading or Migrating to JBoss Enterprise SOA Platform

### Upgrading from JBoss ESB 4.0.

You will need to perform a clean install of JBoss Enterprise SOA Platform. Please refer to the steps detailed in [Section 3, “Installing the SOA Platform”](#). Once you have done this you can copy your ear, war, and jar files from your ESB installation into the appropriate SOA Platform folder: either `<install_location>/jboss_as/server/production` in the case of the embedded JBoss EAP version, or `<install_location>/jboss_as/server/default` in the case of the standalone version. You should then test your applications to see if they function. Some applications may work with no modification, but 100% code portability between the JBoss Enterprise SOA Platform and JBoss ESB is not guaranteed.

### Upgrading from JBoss ESB 4.2.

You should install the SOA Platform as per the instructions in [Section 3, “Installing the SOA Platform”](#). You will need to move your applications to the newly installed platform manually, where you should test them to see if they require modification. There have been significant changes from 4.2.1GA, especially in the area of JBPM integration. Applications developed for ESB 4.2.1CP1 may not require any modification to run on the JBoss Enterprise SOA Platform.

### Upgrading from JBoss Enterprise Application Platform 4.2.

The JBoss Enterprise SOA Platform contains version 4.3 of the JBoss Enterprise Application Platform. Earlier versions of the JBoss Enterprise Application Platform should be disabled or uninstalled.

If you have applications that depend on an earlier version of the JBoss Enterprise Application Platform you can run this and the JBoss Enterprise SOA Platform on the same machine. For information on running an earlier version of JBoss Enterprise Application Platform and JBoss Enterprise SOA Platform on the same machine please refer to [Section 3, “Running alongside an earlier version of JBoss Enterprise Application Platform”](#)

In either case, follow the instructions in [Section 3, “Installing the SOA Platform”](#) to install the product.

## 3. Installing the SOA Platform

Installation of JBoss Enterprise SOA Platform, once you have your Java environment configured, is quite straightforward.

### Procedure 1.3. Installation

1. Obtain the appropriate JBoss Enterprise SOA Platform zip file

There are two different zip files available. One contains the standalone version of JBoss Enterprise SOA Platform, the other contains the version of JBoss Enterprise SOA Platform with the embedded JBoss Enterprise Application Platform.

Which you use will depend on your needs. The standalone version is a lightweight version with core ESB functionality. The full version with embedded JBoss Enterprise Application Platform includes a fully fledged Java Application Server and Application Framework.

### 2. Verify the integrity of the zip file

Along with the zip file is a `.MD5` file. This file contains a checksum that can be used to verify the integrity of the zip file, alerting you if the file has been changed since it was uploaded, or corrupted on download.

On a Linux system you can use the command `md5sum <filename>`. On a Windows system you will need to download an md5sum generating program such as [MD5summer](http://www.md5summer.org/)<sup>2</sup>. This will generate an md5sum that you can compare with the sum in the `.MD5` file. If the two md5sums are not identical it means that the zip file you have has been corrupted on download, or has been changed since it was uploaded to the server.

### 3. Expand the zip file

To install JBoss Enterprise SOA Platform, simply unzip the appropriate zip file to a location of your choice.

JBoss Enterprise SOA Platform is now installed on your system. In the following chapter we will test the installation to make sure that everything is working properly.

---

<sup>2</sup> <http://www.md5summer.org/>

# Post-Installation: Configuring and starting the server

This chapter covers post-installation configuration and starting of JBoss Enterprise SOA Platform.

## 1. Post-installation configuration

### 1.1. Changing the Database used by the platform



#### Note

Hypersonic SQL provides default "out of the box" database functionality for evaluation and development use only. It is NOT recommended or supported as a production-use database. Technical support is not available for this component, and while we are happy to accept bugs filed against this component, we do not make any commitment to fix them within a specific timeframe.

In the `<install-directory>/jboss-as/tools/schema` directory you will find a set of scripts that will reconfigure the main components of the platform to use the supported database of your choice.

You can perform the reconfiguration manually by typing `ant` while in the `schema` directory, or automatically by editing `build.properties`. Instructions to configure `build.properties` are included as comments in the file itself.

Please refer to the release notes for any further notes relating to database configuration.

#### 1.1.1. Monitoring/Management Console Configuration

By default, the monitoring/management console uses `hsqldb` as a back-end database. It has also been tested with Oracle and MySQL, but could be extended to use any other JDBC/hibernate-supported database.

In the `management-esb` directory there is a `db.properties` file. In order to change the database from `hsqldb` to `mysql` or `oracle`, edit this file and change the `db` property to `mysql` or `oracle`.

Next, change the `/tools/management-esb/src/main/resources/<db>/management-ds.xml` file so that the settings match the datasource for your database.

JBoss ships with `hsqldb.jar` in the `server/<instance>/lib` directory. In order to use a different database as a back-end, please also copy the appropriate the JDBC driver into the `server/<instance>/lib` directory of your application server.

For MySQL, it may be necessary to create the database "statistics" before deploying if you have not edited the `management-ds.xml` file.

Please refer to `tools/console/MANAGEMENT-README.txt` for further information.

### 1.2. Securing the server for production environments



#### Warning

The following are important notes relating to the security of your system.

#### 1.2.1. Securing the JBPM Console

Two distinct `jbpm-console.war` files are shipped with the platform. One is a development version which allows unauthenticated access to deploy processes to the server, for use with a graphical process design tool such as JBoss Developer Studio while developing applications. The other is a production version which secures the console against remote deployment. *You should not run your server in a production environment with the unsecured development version of `jbpm-console.war` deployed.* Doing so poses a threat to the security of your server.

#### Standalone version of JBoss Enterprise SOA Platform.

In the standalone version, we ship with the unsecured uploader console by default. *Initially, your server is configured for development.* The jBPM JPDl will be able to deploy processes. Before putting it into production you should secure the console.

#### Procedure 2.1. To secure the console in the standalone version

- Copy the file `/tools/resources/jbpm-console-production.war` to `/server/default/deploy/jbpm.esb/jbpm-console.war`.

#### Procedure 2.2. To enable remote deployment of processes in the standalone version

- Copy `/tools/resources/jbpm-console-development.war` to `/server/default/deploy/jbpm.esb/jbpm-console.war`.

In each case the file must be overwritten. You can not have two versions of the `war` in the

deployment directory.

### **Embedded JBoss Enterprise Application Platform version of JBoss Enterprise SOA Platform.**

In the embedded JBoss Enterprise Application Platform version, the "all" profile has the development version of the war, and the "production" profile has the production version. *By default your server is configured to operate in a secure mode.* To enable it for development mode you need to run in the unsecured mode of operation.

### **Procedure 2.3. To secure the console in the embedded EAP version**

- Start the server with no command line parameters or with the parameter "`-c production`"

### **Procedure 2.4. To enable remote deployment of processes in the embedded EAP version**

- Start the server using the parameter "`-c all`"

We *do not recommend* running the server on an unsecured network with the `jbpm-console-development.war` deployed or using the `all` profile without appropriate modification.

### **1.2.2. Preventing download of non-RMI classes on Port 8083 in the standalone version of the server**

If you use RMI (Remote Method Invocation) you will want to make Port 8083 of your server accessible to clients. The EAP version of the server is configured out of the box to restrict the classes that it serves on this port. The standalone server, however, is configured out of the box to serve all deployable classes via this port. This has been done to allow the quickstarts to function correctly by default.

To change this behaviour you need to modify the following line in `default/conf/jboss-service.xml`:

```
<!-- Should non-EJB .class files be downloadable -->
<attribute name="DownloadServerClasses">false</attribute>
```

The value for this attribute is set as `true` out of the box, and it should be set to `false` in actual production deployment to prevent the server from serving all deployable classes on Port 8083.

## 2. Starting JBoss Enterprise SOA Platform

Starting JBoss Enterprise SOA Platform is straightforward. See [Section 3, “Running alongside an earlier version of JBoss Enterprise Application Platform”](#) for instructions specific to using the server with an earlier version of the JBoss Enterprise Application Platform.

### "Default" server profile.

There are several different server profiles for the embedded EAP version of the platform, and only one for the standalone version. The default server profile is the server profile that will be used when the server is started without any command line parameters. This profile has ESB server functionality enabled.

In the case of the standalone version the default profile is named `default`.

In the case of the embedded EAP version there is a profile named `default`, but this is not the default profile used. The default profile used when the embedded EAP server is started with no parameters is the profile named `production`. Please refer to [Section 1.2, “Securing the server for production environments”](#) for more information about this profile.

### Procedure 2.5. Starting the Server

1. Change directory to the JBoss Enterprise SOA Platform installation directory.
2. Change into the `jboss-as/bin` directory.
3. If you are using Microsoft Windows then issue the command `run.bat`. On Linux or Unix-derived systems, use the command `./run.sh` to start the server.

JBoss Enterprise SOA Platform will now start up using the `production` profile, if you are using the version with embedded JBoss Enterprise Application Platform, or using the `default` profile if you are using the standalone version. In each case this is the profile with the ESB deployed. The server should now be accessible at: `http://localhost:8080`

To start the server with a different profile, simply use the startup command followed by `-c <profile name>`.

### 2.1. Troubleshooting Start up

Here are a few common problems with suggested resolutions.

#### JAVA\_HOME incorrectly set .

An incorrectly set environment variable named `JAVA_HOME` will cause the server to fail to start with a related error message. Set the `JAVA_HOME` variable to point to your JDK installation to solve this.



### **VM cannot allocate sufficient memory .**

This error occurs when there is not enough free memory available to the system to satisfy the memory requirements of JBoss Enterprise SOA Platform. You will need to quit applications to free up memory, allocate more virtual memory, or increase the amount of physical RAM installed in the system to make this memory available.

## **3. Running alongside an earlier version of JBoss Enterprise Application Platform**

JBoss Enterprise SOA Platform can run alongside an earlier version of JBoss Enterprise Application Platform. The JBoss Enterprise SOA Platform includes version 4.2 of JBoss Enterprise Application Platform, but you may have applications which depend on another version.

Either:

1. multi-home your network card so that it has multiple IP addresses (one for each instance)

- Launch each instance of JBoss AS with the command line option to bind this instance to a particular IP address:

```
run.bat/run.sh -b <ip-address or host>
```

2. or provide alternate ports for your server instance:

- This is easier to do and does not require a sysadmin. However, it increases firewall rule complexity.
- Modify the file `<serverroot>/jboss-as/server/production/conf/jboss-service.xml` if you are using the embedded JBoss Enterprise Application Platform; or the file `<serverroot>/default/conf/jboss-service.xml` if you are using the standalone version of the platform.
- Uncomment the `Service Binding` section and select a `ServerName` value from `docs/examples/binding-manager/sample-bindings.xml` (e.g. `ports-01` or `ports-02` or make your own named port configuration)



## Testing the Installation

## 1. Running the Helloworld Quickstart

The Helloworld quickstart, located in `samples/quickstarts/helloworld`, provides a handy way to make sure your installation is working.

### Prerequisites.

In order to build the quickstart you need the [ant](#)<sup>1</sup> java build tool installed on your machine. Users of Red Hat Enterprise Linux 5 can install it via the `ant` package found in RHEL. Ant is also packaged in rpm format by [jpackage.org](#)<sup>2</sup>.

Versions of `ant` installed via `rpm` do not require additional configuration. If you install a version manually from a zip file from [ant.apache.org](http://ant.apache.org)<sup>3</sup> you will need to set some environment variables. Please refer to the installation documentation for `ant` for details.

### Procedure 3.1. Building and running the Helloworld quickstart

1. Start the Server following the instructions in [Section 2, “Starting JBoss Enterprise SOA Platform”](#).
2. From a command terminal window, change directory into the `samples/quickstarts/helloworld` directory.
3. Type `ant deploy` to deploy the helloworld .esb package archive to the server.
4. Type `ant runtest`.
5. If you are using the standalone version of JBoss Enterprise SOA Platform, switch back to the server console. You should shortly see a message similar to the following:

```
12:14:51,044 INFO [STDOUT] &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
12:14:51,044 INFO [STDOUT] Body: Hello World
12:14:51,044 INFO [STDOUT] &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
12:14:51,054 INFO [STDOUT] Message structure:
12:14:51,054 INFO [STDOUT] [ message: [ JBOSS_XML ]
```

6. If you are using the version of JBoss Enterprise SOA Platform that includes the JBoss Enterprise Application Platform, you will need to monitor the server log file, located in `<install-location>/jboss-as/server/production/log/server.log`, for the message. The production profile (which is the default profile for the server) does not log much

<sup>1</sup> <http://ant.apache.org>

<sup>2</sup> <http://www.jpakage.org/>

<sup>3</sup> <http://ant.apache.org>

information to the console, for performance purposes. Instead information is recorded in the `server.log` file. Examine this file for the appearance of the "Hello World" message. If you see it there everything is working as it should.

---

# Appendix A. Revision History

## Revision History

Revision 7.3.0-1

Thu Jul 12 2007

David O'Brien

Changed file naming system to be more logical. Boilerplate is now Legal\_Notice.

Revision history moved to appendix.

Revision 7.3.0-0

Wed Jun 13 2007

David O'Brien

Reorganized book contents and added revision history.

