JBoss Enterprise SOA Platform 4.3

Getting Started Guide

Your guide to getting started with the JBoss Enterprise SOA Platform

JBoss Enterprise SOA Platform 4.3 Getting Started Guide
Your guide to getting started with the JBoss Enterprise SOA Platform
Edition 1

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

1801 Varsity Drive
Raleigh, NC 27606-2072USAPhone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588Research Triangle Park, NC 27709USA

A guide to the inital installation & configuration of the JBoss Enterprise SOA Platform.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight key caps and key-combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**-**`Alt`**-**`F1`** to switch to the first virtual terminal. Press **`Ctrl`**-**`Alt`**-**`F7`** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`Mono-spaced Bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

[1] https://fedorahosted.org/liberation-fonts/

Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventions

Two, commonly multi-line, data types are set off visually from the surrounding text.

Output sent to a terminal is set in `Mono-spaced Roman` and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in `Mono-spaced Roman` but are presented and highlighted as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }

}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, 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 Bugzilla: *http://bugzilla.redhat.com/bugzilla/* against the product **JBoss_SOA_Platform.**

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

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.

1.1. Prerequisites

1.1.1. Hardware Requirement Recommendations

These recommendations do not represent an absolute minimum requirement for running the SOA Platform but rather present a baseline for reasonable performance. The SOA Platform will run on hardware that is below these recommendations although your performance will suffer accordingly.

Also note that performance is affected by your choice of hardware platform. When using 'non-server class' hardware such as a laptop for development & testing, performance may be less than when deployed in your production environment.

These guidelines also do not take into account the requirements of the server's operating system or other applications. It is the developers' responsiblity to provide performance requirements for their own services & applications to be deployed.

CPU

Minimum: Pentium4 class.

Recommended: Dual-core or greater.

MEMORY

1 Gigabyte of memory is a recommended minimum. Increasing memory is often the most effective means of increasing performance.

STORAGE SPACE

The full installation of the SOA Platform server requires 600 megabytes of harddrive space. The standalone ESB server only requires 220 megabytes of space. You will also require additional space for log files. 10 Gigabytes is considered adequate for a production environment with log rotation configured.

1.1.2. 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.

The following are the the JDK implementations which have been tested and certified for the SOA Platform. They are fully supported and compatiable implementations.

Supported Java Virtual Machine Versions

JVM	Versions
Sun JDK	1.5.x, 1.6.x
BEA JRockit JDK	1.5.x
HP-UX JDK	1.5.x
IBM JDK	1.5.x
Azul JDK	1.5.x

1.1.3. 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**

3. Use **up2date** (Red Hat Enterprise Linux 4) or **yum** (Red Hat Enterprise Linux 5) 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.

1.1.4. 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*. 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**.

1.2. Upgrading or Migrating Applications to 4.3GA

This release of the JBoss Enterprise SOA Platform is backwards compatible with previous releases with one exception. The inclusion of jBPM 3.2.3 in this release has introduced one additional configuration requirement. It is now required to set the jta.UserTransaction property in your **hibernate.cfg.xml** to **UserTransaction**. This is because of changes to transaction lookups in this new version of jBPM.

Important

Although every effort is made to ensure a smooth transition, every deployment environment is different. As a matter of best practices it is very highly recommended to test all your existing applications on this new version of the SOA Platform before deploying it into your production environment.

1.3. 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 EAP Server.

   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 EAP 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*[1]. 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.

Post-Installation: Configuring and starting the server

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

2.1. Post-installation configuration

2.1.1. Changing the Database used by the platform

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, including the Management and Monitoring consoles, to use the supported database of your choice.

There are a number of caveats:

- The scripts expect the platform to be in one of the configurations that the scripts provide. The scripts can be run again to change the configuration unless changes are made in addition to what the scripts provide.
- The ESB Management Console is a tech preview at this time.

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.

After the database reconfiguration the Management Console must be redeployed manually.

1. Edit the file **jboss-as/tools/console/management-esb/build.xml** and change the value for **org.jboss.esb.server.config** to **production** or **development** (if you have a **development** profile).
2. Run **ant deploy**

Important

Hypersonic SQL provides default database functionality for evaluation and development use only. It is *not* recommended or supported as a production-use database.

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

2.1.2. Creating a Development profile

For testing or development purposes you may wish to run the JBoss SOA Platform server without full production capabilities, such as clustering, enabled. The SOA Platform includes a configuration profile called *default* which is suitable for these situations.

You can either use this profile or create your own profile based on it.

```
cd /opt/jboss-soa-p.4.3.0/jboss-as/server/
cp -rf default development
cd /opt/jboss-soa-p.4.3.0/jboss-as/bin/
./run.sh -c development
```

Example 2.1. Creating and running a new profile based on the *default* profile on Linux/Unix

```
cd c:\jboss-soa-p.4.3.0\jboss-as\server\
xcopy default development /E
cd c:\jboss-soa-p.4.3.0\jboss-as\bin\
run.bat -c development
```

Example 2.2. Creating and running a new profile based on the *default* profile on Windows

2.1.3. Securing the server for production environments

2.1.3.1. 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. 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 EAP version of JBoss Enterprise SOA Platform

In the embedded JBoss EAP 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.

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

If you use Remote Method Invocation (RMI) you need to allow client access to port 8083 of your server. The default configuration of the *EAP embedded* SOA Platform server only permits download of EJB classes.The *standalone* SOA Platform server is configured to serve all deployable classes on this port [1].

This behaviour is controlled by the following line in **default/conf/jboss-service.xml**:

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

Figure 2.1. Configuration setting allowing download of non-EJB classes

In a production environment this value must be set to **false**.

2.2. Starting JBoss Enterprise SOA Platform

Starting JBoss Enterprise SOA Platform is straightforward. See *Section 2.3, "Running alongside an earlier version of JBoss EAP"* 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 profile used when the embedded EAP server is started with no parameters is the profile named **production**. Please refer to *Section 2.1.3, "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.

[1] This allows the quickstarts to function correctly by default.

JBoss Enterprise SOA Platform will now start up using the **production** profile, if you are using the version with embedded JBoss EAP Server 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 by accessible at: *http://localhost:8080*.

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

2.2.1. Enabling Access to the Server Consoles

Access to the server consoles is disabled in the default configuration.

To grant access you need to edit the files **soa-users.properties** and **soa-roles.properties**. These files are located in the **conf/props** directory of the server profile that you wish to allow access for.

soa-users.properties contains a list of users and their passwords in plain text. The format is **username=password**.

soa-roles.properties contains a list of users and the server roles that are assigned to them. The format is **username=role1,role2,role3** where there can be any number of roles.

These user and role details do not correspond to any other account details, such as a operating system user account. You can arbitrarily create user accounts here.

Procedure 2.6. Enabling Access to the Server Consoles

1. You need to add the required username and password to **soa-users.properties**, or enable the user **admin** by uncommenting that line. If you enable the **admin** user you should also change its password.

```
#admin=admin
harold=@dm1nU53r
```

Example 2.3. A new user added in **soa-users.properties**

2. You also must add that user to the **soa-roles.properties** file. The roles the user must be assigned to for Server Console access are **JBossAdmin**, **HttpInvoker**, **user** and **admin**.

```
#admin=JBossAdmin,HttpInvoker,user,admin
harold=JBossAdmin,HttpInvoker,user,admin
```

Example 2.4. Assigning user roles in **soa-roles.properties**

2.2.2. 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.

2.3. Running alongside an earlier version of JBoss EAP

JBoss Enterprise SOA Platform can run alongside an earlier version of JBoss EAP. You might want to do this if you have applications which depend on a different version of JBossEAP than the one included in the embedded version.

In this situation you have two options:

1. You can *multi-home* your network card so that it has multiple IP addresses. Once this is done you can launch each instance of JBoss EAP with the command line option to bind each instance to a particular IP address:

```
./run.sh -b  {ip-address or host}
```

Example 2.5. Using -b option to bind to an ip or hostname in Linux/Unix

```
run.bat -b  {ip-address or host}
```

Example 2.6. Using -b option to bind to an ip or hostname in Windows

2. Use the Service Bindings Manager to configure different ports for each server instance.

   This does not require changes to the hosting operating environment or hardware. However it will require changes to your firewall rules.

   Procedure 2.7. Enabling the Service Binding Manager

   1. Open **`jboss-service.xml`**.

      This file is located in **`jboss-as/server/production/conf/`** if you are using the embedded JBoss EAP or **`default/conf/`** if you are using the standalone version of the platform.

   2. 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

   3. Update your firewall configuration to take these new configurations into account.

QuickStarts

3.1. The HelloWorld QuickStart

This QuickStart allows you get up and running with the JBoss SOA Platform out of the box. It is located in the distribution under **samples/quickstarts/helloworld**.

To run this QuickStart:

1. Ensure that the **samples/quickstart/conf/quickstarts.properties** file has the correct configuration and home directory settings for your server. This configuration is used by all the QuickStarts.

```
# Location of your JBoss Application Server installation.
org.jboss.esb.server.home=/opt/jboss-soa-p.4.3.0/jboss-as

# JBossAS server name.
org.jboss.esb.server.config=default
```

   Example 3.1. QuickStart deployment properties

2. Manually start the server using the startup scripts as described in *Section 2.2, "Starting JBoss Enterprise SOA Platform"*.

   If you are running the server on Windows do not run it as a Windows Service for the examples in this guide. You need to be able to view the console output.

3. From a command terminal window, change directory into the **samples/quickstarts/helloworld** directory.

4. Run the command **ant deploy** to deploy the helloworld **.esb** archive to your application server.

5. Run the command **ant runtest**.

6. You should now see a "Hello World" message appear in your application server's console.

```
17:25:12,614 INFO  [STDOUT] &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
17:25:12,615 INFO  [STDOUT] Body: Hello World
17:25:12,615 INFO  [STDOUT] &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
17:25:12,615 INFO  [STDOUT] Message structure:
17:25:12,615 INFO  [STDOUT] [ message: [ JBOSS_XML ]
```

   Figure 3.1. Helloworld quickstart console output

That's it! The QuickStart ran successfully. Your environment is properly configured.

You can find more detailed directions about each QuickStart example by running **ant help-quickstarts** in the specific QuickStart directories or by consulting the QuickStart's **readme.txt** file. You can also find information on how to run a particular QuickStart under different deployment scenarios by running **ant help** in the directory of specific QuickStart.

3.2. Components of the QuickStart

The following diagram illustrates the sequence of events that take place in this QuickStart. It touches on a number of the key concepts within the JBoss SOA Platform.

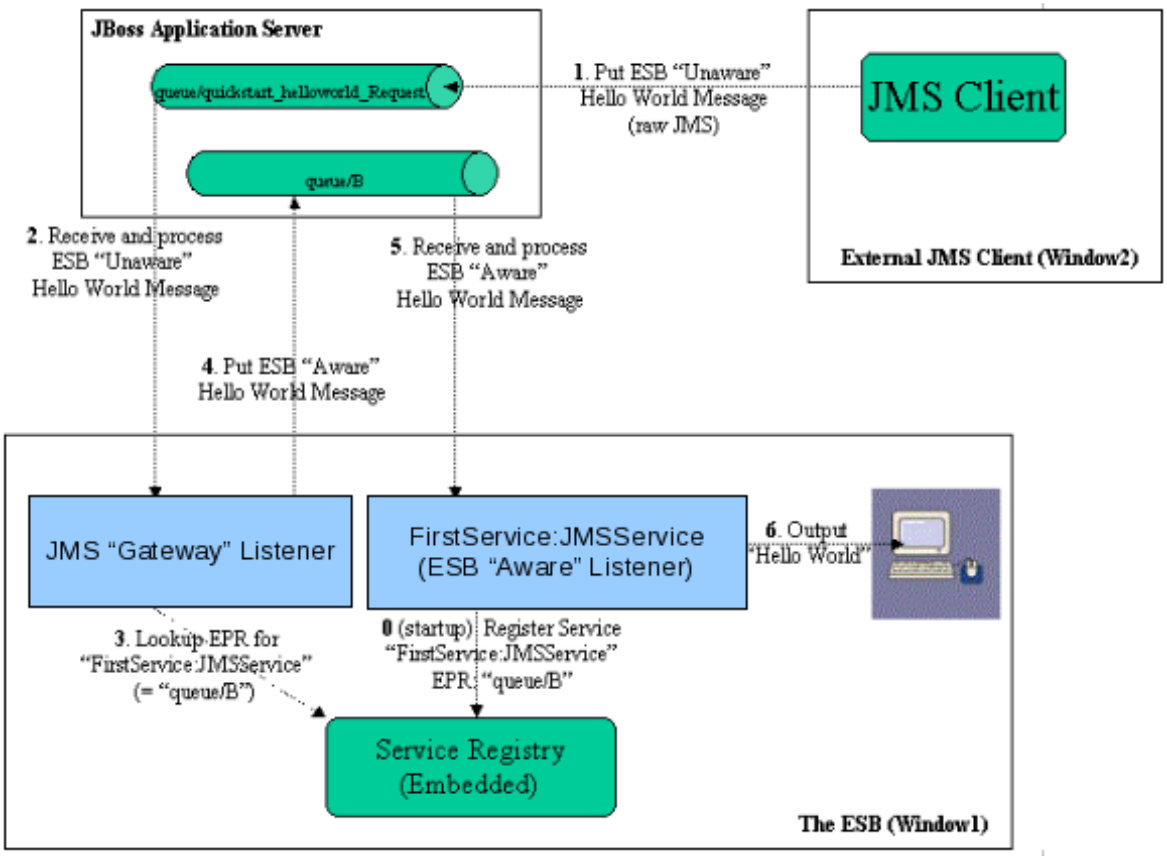

Figure 3.2. QuickStart Components & event sequence

Service Registry

This is a JAXR Registry implementation. In this QuickStart, the registry uses RMI based communication. The JBoss SOA Platform Services Guide contains more details on the Registry Service.

JMS Gateway Listener

A "Gateway Listener" is one of the key architectural components within the JBoss SOA Platform. This listener type provides an interface between the SOA Platform and external service endpoints. In this case, we're using a JMS Gateway.

The ESB Aware Service Listener

The **`FirstService:SimpleListener`** ESB Aware Service Listener listens for *ESB Aware* messages on queue/quickstart_helloworld_Request_gw.

3.2.1. ESB Aware and Unaware Messages

Messages passed between components within the JBoss SOA Platform Enterprise Message Bus are expected to comply with the definition of a message in **`xml/message.xsd`**. This well defined concept of a message is often refered to as an *ESB Message*[1].

This allows easy and effective communication between services within the SOA Platform domain.

However service endpoints outside of the ESB are not expected to be aware of these requirements or comply with them. These are referred to as *ESB unaware* or *non-ESB* Endpoints and Messages. A *gateway* listener provides a layer of translation for messages passing between ESB aware and non-aware components.

3.3. Helloworld QuickStart Sequence of Events

After starting the JBoss SOA Platform server in Window1 and before any "Hello World" messages are put on the ESB, the "FirstService:SimpleJMSService" Service is registered with the Registry Service.

The sequence of events in the Hello World QuickStart are as follows:

- ESB Unaware JMS Client endpoint puts an ESB Unaware "Hello World" Message (plain String Object) into JMS Queue "queue/quickstart_helloworld_Request".
- The JMS Gateway Listener receives the ESB Unaware message. The Gateways Job is to adapt this message by making it an ESB Aware Message for consumption by an ESB Aware Endpoint.
- The JMS Gateway Listener uses the registry to lookup the Endpoint Reference (EPR) for "FirstService:SimpleJMSService" Service. This works out to be JMS Queue "queue/quickstart_helloworld_Request_gw".
- The JMS Gateway Listener "adapts" the message into an ESB Aware message and places it into JMS Queue "queue/=quickstart_helloworld_Request_gw".
- "FirstService:SimpleJMSService" Service receives the message.
- "FirstService:SimpleJMSService" Service extracts the payload from the message and prints it to the console.

3.4. Running other Quickstarts

Once you have successfully run the Helloworld QuickStart and understand the concepts involved, there are many other Quickstarts to try.

Important

Please note that the Quickstarts each have different requirements which are documented in their respective **`readme.txt`**. Not all the Quickstarts will run on every server configuration.

There are several dozen Quickstarts included. Below is a suggested list of Quickstarts which it is recommended to look at first.

[1] This topic is more fully discussed in the SOA Platform Programmer's Guide.

- helloworld
- helloworld_action
- custom_action
- helloworld_file_action
- helloworld_ftp_action
- simple_cbr
- fun_cbr
- business_service
- business_rules_service
- scripting_groovy
- transform_CSV2XML
- transform_XML2POJO
- transform_XML2XML_simple
- transform_XML2XML_date_manipulation
- aggregator
- bpm_orchestration1
- bpm_orchestration2
- webservice_consumer1
- webservice_producer

Warning

The **`groovy_gateway`** Quickstart does not work when the server is running in 'headless' mode. The SOA Platform runs in this mode by default due to the requirements of other components. More details regarding this can be found in the known issues section of the SOA Platform Release Notes and online at *http://jira.jboss.org/jira/browse/SOA-906* .

Appendix A. Revision History

Revision History
Revision 1.0 Wed 10 Sep 2008 DarrinMison*dmison@redhat.com*
Created

Index

F