JBoss ESB 4.7

Getting Started With JBoss ESB

JBESB-GS-11/6/09

Legal Notices

The information contained in this documentation is subject to change without notice.

JBoss Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. JBoss Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Java™ and J2EE is a U.S. trademark of Sun Microsystems, Inc. Microsoft® and Windows NT® are registered trademarks of Microsoft Corporation. Oracle® is a registered U.S. trademark and Oracle9™, Oracle9 Server™ Oracle9 Enterprise Edition™ are trademarks of Oracle Corporation. Unix is used here as a generic term covering all versions of the UNIX® operating system. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Copyright

Software Version

JBoss ESB 4.7

Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions as set forth in contract subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause 52.227-FAR14.

Contents

Contents.. iii

About This Guide... 4

What This Guide Contains.................................... 4

Audience.. 4

Prerequisites... 4

Organization... 4

Documentation Conventions.................................5

Additional Documentation....................................5

Contacting Us.. 5

Getting Started... 6

Downloading.. 6

Installation to the JBoss Application Server (not required for ESB Server)................................... 6

Installation to the JBoss Application Server 5.1.0.GA... 7

Windows script update for JBoss AS 5.............7
Scoped Deployments in JBoss AS 5.................8
Java 6... 8

ESB archive deployment strategies......................8

The Hello World QuickStart................................. 9

Components of the QuickStart........................11
ESB Aware and Unaware Messages...............12
QuickStart Sequence of Events......................12

Running other Quickstarts................................. 14

About This Guide

What This Guide Contains

The goal of this document is to assist you in getting test applications up and running on JBossESB as quickly as possible.

Audience

This guide is for anyone who is wants to quickly understand how to install and use JBoss ESB 4.7.

Prerequisites

None.

Organization

This guide contains the following chapter:

1. **Chapter 1, Installation:** This chapter reviews prerequisites (software needed to operate JBossESB), downloading JBossESB, and building JBossESB.

Documentation Conventions

The following conventions are used in this guide:

Convention	Description
Italic	In paragraph text, italic identifies the titles of documents that are being referenced. When used in conjunction with the Code text described below, italics identify a variable that should be replaced by the user with an actual value.
Bold	Emphasizes items of particular importance.
`Code`	Text that represents programming code.
Function | Function	A path to a function or dialog box within an interface. For example, "Select File | Open." indicates that you should select the Open function from the File menu.
( ) and |	Parentheses enclose optional items in command syntax. The vertical bar separates syntax items in a list of choices. For example, any of the following three items can be entered in this syntax: `persistPolicy (Never | OnTimer | OnUpdate | NoMoreOftenThan)`
Note:	A note highlights important supplemental information.
Caution:	A caution highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results.

Table 1 Formatting Conventions

Additional Documentation

In addition to this guide, the following guides are available in the JBoss ESB 4.7 documentation set:

1. **JBoss ESB 4.7** *Administration Guide*: How to manage JBossESB.

2. **JBoss ESB 4.7** *Programmers Guide*: How to use JBossESB.

3. **JBoss ESB 4.7** *Release Notes*: Information on the differences between this release and previous releases.

4. **JBoss ESB 4.7** *Services Guides*: Various documents related to the services available with JBossESB.

Contacting Us

Questions or comments about JBoss ESB 4.7 should be directed to our support team.

Chapter 1

Getting Started

The quickest way to get started using JBoss ESB is by running one of the quickstarts in the *samples/quickstarts* folder. This will also perform a basic validation of your system. Before doing this however, be sure to check that your system meets the following minimum requirements:

1. JDK 5 (v1.5.0_06 recommended)
2. Ant (v1.6.5 recommended)
3. JBoss Application Server 4.2.x.GA or JBoss ESB Server 4.5.GA

There are two ways to run JBossESB. You can deploy it to JBossAS / JBossESB Server, or run standalone. This guide will concentrate on the JBoss Application Server / JBoss ESB Server scenario because these scenarios are the most common and provide the largest range of functionality to users. We suggest using the JBoss ESB Server for this guide.

The ESB components can also be deployed directly to the JBoss Application Server. If you require .EAR deployment or if you require EJB3, then you will want to use the JBoss Application Server. Below are the steps needed to install JBoss ESB to the JBoss Application server. If you plan to use the JBoss ESB Server, no additional installation is required.

The JBoss ESB Server is an application server that serves as a convenient and lightweight container to deploy to. It is a stripped down version of the JBoss application server that will deploy all of your WAR, SAR, but does not contain EJB3 libraries or deployers. The main advantage of using the ESB Server over using the JBoss ESB 4.7 Server is that it has a much quicker boot time than the JBoss Application Server, which is helpful during development.

Downloading

This document assumes you have Ant (1.6.5 or higher) and Java5 installed on your machine, and that you have a fresh copy of JBossAS. So now go and download the JBoss ESB 4.7 distribution from http://labs.jboss.com/portal/jbossesb/downloads. There are three corresponding distribution versions, jbossesb-server-{version}, jbossesb-{version} and jbossesb-{version}-src. The standalone JBossESB Server execution requires the jbossesb-server-{version} distribution while deployed execution requires the jbossesb-{version} distribution.

If you wish to use the JBoss Application Server, download the JBoss ESB 4.7 distribution from the above URL. Then download JBoss AS 4.2.2.GA from http://labs.jboss.com/portal/jbossas/downloads JBoss AS 5 is supported starting from version 5.1.0.GA and can be downloaded from the same location.

Installation to the JBoss Application Server (not required for ESB Server)

- Use jbossesb-{version}/install/deployment.properties.example as a template to create install.deployment.properites.
- Edit your version of *install/deployment.properties*. Open this file and edit the following lines if needed to represent the directory where you have installed the JBoss application server and the server configuration you wish. Most users choose “default” for their configuration setting:

```
# application server root directory
org.jboss.esb.server.home=/jboss-4.2.2.GA
# the instance of jboss you are running(default)
org.jboss.esb.server.config=default
```

- Run 'ant' (default target) from the *install* directory. This will deploy JBossESB to your JBossAS instance. It copies several .sar and .esb archives files and directories (jbossesb.esb, jbpm.esb, jbrules.esb, smooks.esb, spring.esb, soap.esb) into the application server's deploy directory.
- Start your server. There is a *bin* directory within the JBoss Application Server or the JBoss ESB Server (whichever you have chosen to use). Execute the run script (*run.sh* on Linux/Mac OS X/Unix, *run.bat* on Windows).
- Verify the server you have chosen is running by visiting http://localhost:8080 with your web browser.

Installation to the JBoss Application Server 5.1.0.GA

Installation steps.

1. Download JBoss AS 5.1.0.GAand unzip it.
2. Follow the same steps as listed above for “Installation to the JBoss Application Server (not required for ESB Server)”

Depending on whether you access certain application (like the jopr console) you might need to increase the memory settings when starting the server (in run.conf):

-Xms128m -Xmx512m -XX:PermSize=200m -XX:MaxPermSize=500m

Windows script update for JBoss AS 5

The run.bat script contains an error which prevents wise from compiling the webservice proxies. In order to fix the type you must edit the run.bat script and locate the following lines

```
    if "x%JAVAC_JAR%" == "x" (
      set "RUNJAR=%JAVAC_JAR%;%JBOSS_HOME%\bin\run.jar"
    ) else (
      set "RUNJAR=%JBOSS_HOME%\bin\run.jar"
    )
```

The fix involves switching the set statements so that they read as follows

```
    if "x%JAVAC_JAR%" == "x" (
      set "RUNJAR=%JBOSS_HOME%\bin\run.jar"
    ) else (
      set "RUNJAR=%JAVAC_JAR%;%JBOSS_HOME%\bin\run.jar"
    )
```

Scoped Deployments in JBoss AS 5

In ESB 4.x you can specify that a deployment be scoped by configuring this in deployment.xml:

```
<jbossesb-deployment>
 <depends>jboss.esb:test=server</depends>
 <depends>jboss.esb.qa.junit.destination:service=Queue,name=esb_gateway_channel
_versioned</depends>
 <depends>jboss.esb.qa.junit.destination:service=Queue,name=esb_channel_version
ed</depends>
 <loader-repository>
  org.jboss.soa.esb:loader=simple-scoped
  <loader-repository-config>
   java2ParentDelegaton=false
 </loader-repository-config>
 </loader-repository>
</jbossesb-deployment>
```

The AS5 deployers ignore this section of deployment.xml. Instead for a ESB 4.x scoped deployment, users will be required to create a **META-INF/jboss-classloading.xml:**

```
<classloading xmlns="urn:jboss:classloading:1.0" domain="simple-scoped" parent-
first="false" />
```

Java 6

When using Java 6 the following jars are required to be available in <server_home>/lib/endorsed

- jaxb-api.jar
- jbossws-native-jaxrpc.jar
- jbossws-native-jaxws-.jar
- jbossws-native-jaxws-ext.jar
- bossws-native-saaj.jar

ESB archive deployment strategies

JBossESB is packaged and shipped with base services. A service should be deployed in an ESB archive consisting of an action code + configuration. The idea behind an ESB archive is that it is a deployable service unit. An ESB archive is simply a zip file with an .esb extension. You can deploy as many ESB archives as you please. You can influence the deployment order of archives using the deployment.xml, which specifies start-order dependencies.

Typically you would deploy an ESB archive to the 'deploy' directory. ESB archives should enable you to move services between servers simply by moving the corresponding ESB archive. An ESB archive file has the following structure:

```
├──META-INF
│    ├──jboss-esb.xml
│    ├──deployment.xml
```

```
      │    └───MANIFEST.MF
      ├───<java classes>
      ├───<jars>
      └───<queue-service.xml>
```

- <jboss-esb.xml>: contains the service configuration (listener and actions), as well as provider configuration.
- <deployment.xml>: is optional, but can be used for 2 reasons:
  - make this .esb archive depend on other archives, to specify classloading order.
  - make the deployment of this .esb archive scoped.
- <java classes>: your custom action classes in standard package hierarchy.
- <jars>: additional jar archives your actions depend on.
- <queue-service.xml>: if the 'providers' section of the jboss-esb.xml references queues or topics, you can deploy their configuration in the ESB archive. Note that this is strictly a convenience and any other way to deploy these queues is fine too. We recommend this approach to keep your deployments as self-contained as possible therefore keeping dependency management simple.

JBossESB ships with a number of standard service archives:

- jbossesb.esb - internal services like the DeadLetterService,
- jbrules.esb - needed for rules evaluation using JBossRules, for services like the Content-Based Router
- jbpm.esb - needed for default provider of Business Process Management jBPM, and
- smooks.esb - default message transformation engine Smooks.

These services are deployed by default, but you should be able to remove them if you don't need these service deployments.

The Hello World QuickStart

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

To run this QuickStart:

1. Start your Server in a way that allows you to view the output console.   If you are on Windows, it is preferable not to run JBoss as a Windows Service for the purposes of this guide.

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

3. Make sure that *install/deployment.properties* has the correct configuration and home directory settings for your server.

4. Type “ant deploy” to deploy the helloworld .esb package archive to your application server.

5. Type “ant runtest”.

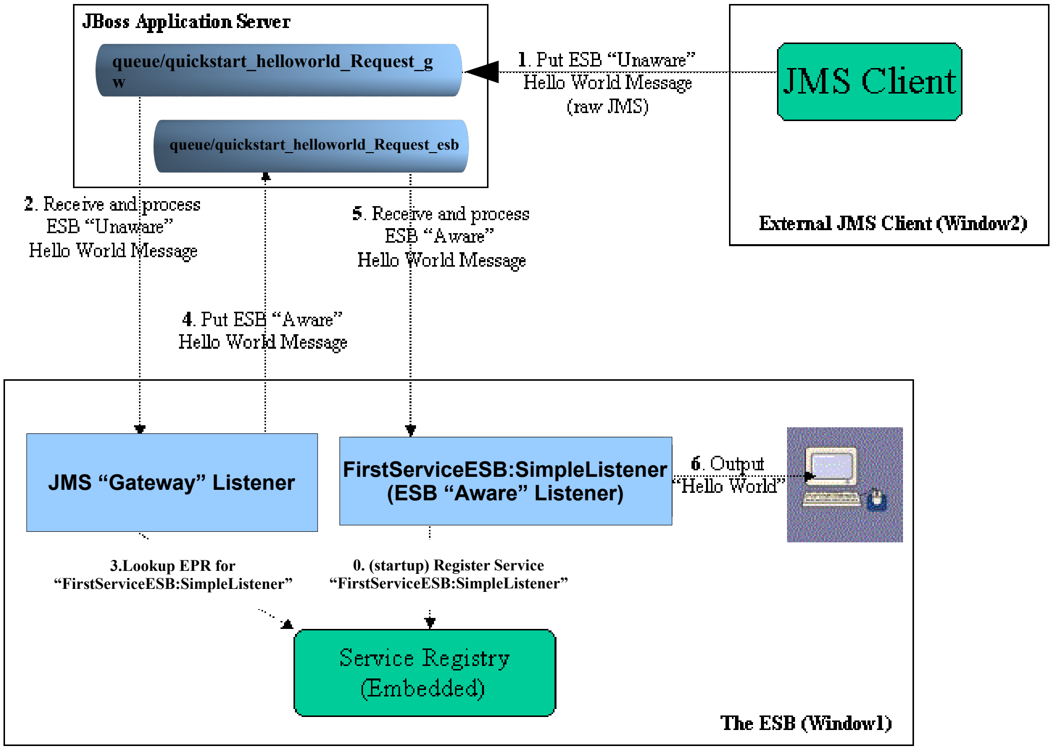
6. Switch back to your application server's console. You should soon see a “Hello World” message appear in your application server's console.

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

You can find more detailed directions on how to setup the quickstart examples by running “ant help-quickstarts” under any of the specific quickstart directories. To get information on how to run a particular quickstart under different deployment scenarios, change directory to the specific quickstart and type “ant help”.

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 JBossESB[1].


JBoss Application Server
queue/quickstart_helloworld_Request_gw
queue/quickstart_helloworld_Request_esb
1. Put ESB "Unaware" Hello World Message (raw JMS)
JMS Client
External JMS Client (Window2)
2. Receive and process ESB "Unaware" Hello World Message
5. Receive and process ESB "Aware" Hello World Message
4. Put ESB "Aware" Hello World Message
JMS "Gateway" Listener
FirstServiceESB:SimpleListener (ESB "Aware" Listener)
6. Output "Hello World"
3.Lookup EPR for "FirstServiceESB:SimpleListener"
0. (startup) Register Service "FirstServiceESB:SimpleListener"
Service Registry (Embedded)
The ESB (Window1)

Window1 shows each of the main "ESB" components used in this sample:

1. **Service Registry**: This is a JAXR Registry implementation. In this QuickStart, the registry uses RMI based communication. See *docs/services/RegistryConfiguration.pdf* for more details on the Registry Service.

2. **JMS Gateway Listener**: A "Gateway Listener" is one of the key architectural components within JBossESB. This listener type is, as its name would suggest, the gateway to the ESB from endpoints outside the domain of the ESB. In this case, we're using a JMS Gateway.

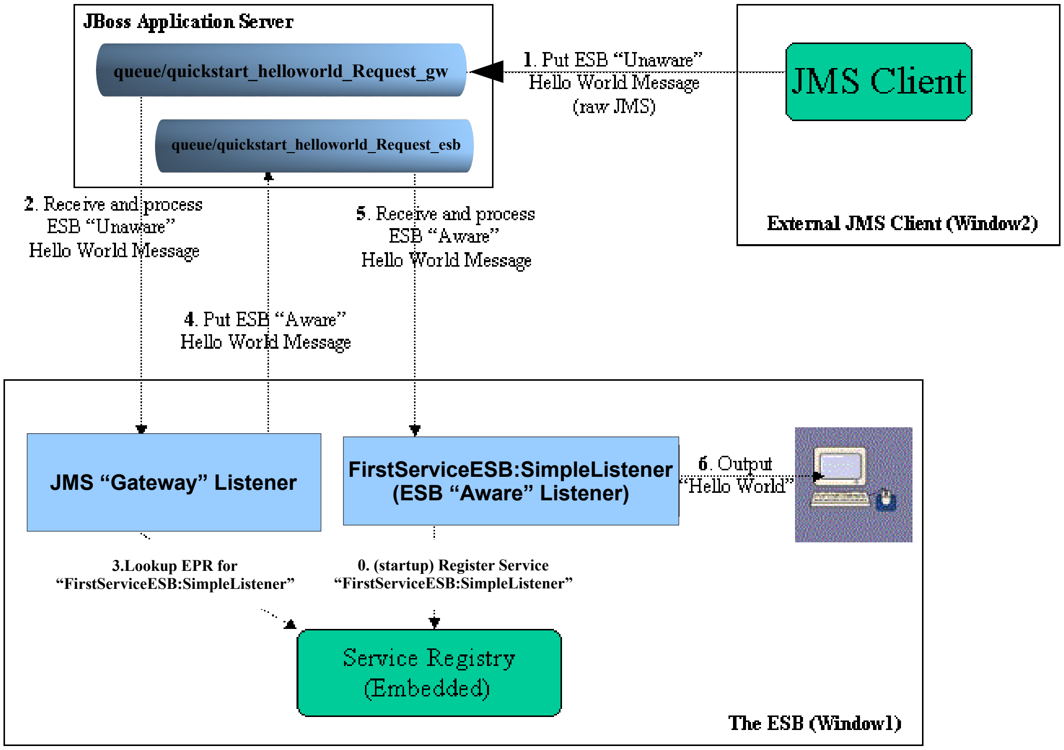
3. **The ESB Aware Service Listener**: The "FirstServiceESB:SimpleListener" ESB Aware Service Listener listens for "ESB Aware" messages on "queue/quickstart_helloworld_Request_esb". This introduces you further to the concept of ESB "Aware" and "Unaware" messages. We will touch on these next.

[1] Use the "Zoom" features of you viewer to see the diagram in more detail.

ESB Aware and Unaware Messages

JBossESB has a well defined concept of what a message is. This is defined fully in *xml/message.xsd*. This construct makes it possible to pass decorated messages payloads between components of the ESB. The message payload is typically stored in the message "Body" (see the Programmers Guide).

This makes a lot of sense from the point of services within the ESB domain being able to collaborate effectively. However, it is not practical to expect endpoints outside the domain of a JBossESB deployment to be "aware" of these internal ESB constructs. For this reason, JBossESB has the concept of ESB Aware and Unaware Messages and Endpoints, with the Gateway acting as the bridge (adapter) between the two worlds.


JBoss Application Server
queue/quickstart_helloworld_Request_gw
queue/quickstart_helloworld_Request_esb
1. Put ESB "Unaware" Hello World Message (raw JMS)
JMS Client
External JMS Client (Window2)
2. Receive and process ESB "Unaware" Hello World Message
5. Receive and process ESB "Aware" Hello World Message
4. Put ESB "Aware" Hello World Message
JMS "Gateway" Listener
FirstServiceESB:SimpleListener (ESB "Aware" Listener)
6. Output "Hello World"
3.Lookup EPR for "FirstServiceESB:SimpleListener"
0. (startup) Register Service "FirstServiceESB:SimpleListener"
Service Registry (Embedded)
The ESB (Window1)

QuickStart Sequence of Events

After starting the ESB in Window1 and before any "Hello World" messages are put on the bus, the "FirstServiceESB:SimpleListener" Service is registered with the Registry Service.

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

1. ESB Unaware JMS Client endpoint puts an ESB Unaware “Hello World” Message (plain String Object) into JMS Queue “*queue/quickstart_helloworld_Request_gw*”.

2. 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.

3. The JMS Gateway Listener uses the registry to lookup the Endpoint Reference (EPR) for “FirstServiceESB:SimpleListener” Service. This works out to be JMS Queue “queue/quickstart_helloworld_Request_esb”.

4. The JMS Gateway Listener “adapts” the message into an ESB Aware message and places it into JMS Queue “queue/quickstart_helloworld_Request_esb”.

5. “FirstServiceESB:SimpleListener” Service receives the message.

6. “FirstServiceESB:SimpleListener” Service extracts the payload from the message and prints it to the console.

Running other Quickstarts

Once you have successfully run the Helloworld QuickStart and understand the concepts involved, there are many other Quickstarts to try. Please note that the Quickstarts have different requirements which are documented in their respective readme.txt, and that not all of the Quickstarts will run in every deployment. Below is a suggested map of Quickstarts to follow in order :

Quickstarts Trailmap:

1. helloworld
2. helloworld_action
3. custom_action
4. helloworld_file_action
5. helloworld_ftp_action
6. simple_cbr
7. fun_cbr
8. business_service
9. business_rules_service
10. scripting_groovy
11. transform_CSV2XML
12. transform_XML2POJO
13. transform_XML2XML_simple
14. transform_XML2XML_date_manipulation
15. aggregator
16. bpm_orchestration1
17. bpm_orchestration2
18. webservice_consumer1
19. webservice_producer