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
JBoss, Home of Professional Open Source Copyright 2006, JBoss Inc., and individual contributors as indicated by the @authors tag. All rights reserved.

See the copyright.txt in the distribution for a full listing of individual contributors. This copyrighted material is made available to anyone wishing to use, modify, copy, or redistribute it subject to the terms and conditions of the GNU General Public License, v. 2.0. This program is distributed in the hope that it will be useful, but WITHOUT A WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details. You should have received a copy of the GNU General Public License, v. 2.0 along with this distribution; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Software Version
JBoss ESB 4.2 Milestone Release 1

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.

© Copyright 2007 JBoss Inc.
## Table of Contents

**Contents**..................................................................iv  
**About This Guide...........................................................5**  
  - What This Guide Contains..........................5  
  - Audience......................................................5  
  - Prerequisites................................................5  
  - Organization..................................................5  
  - Documentation Conventions.........................5  
  - Additional Documentation..........................6  
  - Contacting Us..............................................6  

**Content Based Routing.................................................8**  
  - Introduction..................................................8  
  - JBossRules based Router.................................8  
  - Create a Content-Based Router.......................8  
  - Create a ruleSet..............................................9  
  - Use the XPath Custom Rule Language.............11  
  - Use the Content Based Router.......................11  
  - Splitter..........................................................12  
  - Aggregator ....................................................12
About This Guide

What This Guide Contains
The Content Based Routing contains contain important information on changes to JBoss ESB 4.2 Milestone Release 1 since the last release and information on any outstanding issues.

Audience
This guide is most relevant to engineers who are responsible for administering JBoss ESB 4.2 Milestone Release 1 installations.

Prerequisites
None.

Organization
This guide contains the following chapters:
• Chapter 1, Content-Based Routing: this chapter contains information on how to use the content based routing capabilities in JBossESB.

Documentation Conventions
The following conventions are used in this guide:
<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Italic</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Bold</strong></td>
<td>Emphasizes items of particular importance.</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>Text that represents programming code.</td>
</tr>
<tr>
<td>**Function</td>
<td>Function**</td>
</tr>
<tr>
<td>**( ) and</td>
<td>**</td>
</tr>
<tr>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>A note highlights important supplemental information.</td>
</tr>
<tr>
<td><strong>Caution:</strong></td>
<td>A caution highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results.</td>
</tr>
</tbody>
</table>

Table 1 Formatting Conventions
Additional Documentation

In addition to this guide, the following guides are available in the JBoss ESB 4.2 Milestone Release 1 documentation set:

1. JBoss ESB 4.2 Milestone Release 1 Trailblazer Guide: Provides guidance for using the trailblazer example.

2. JBoss ESB 4.2 Milestone Release 1 Getting Started Guide: Provides a quick start reference to configuring and using the ESB.

3. JBoss ESB 4.2 Milestone Release 1 Programmers Guide: How to use JBossESB.

4. JBoss ESB 4.2 Milestone Release 1 Release Notes: Information on the differences between this release and previous releases.

5. JBoss ESB 4.2 Milestone Release 1 Administration Guide: How to manage the ESB.

Contacting Us

Questions or comments about JBoss ESB 4.2 Milestone Release 1 should be directed to our support team.
Content Based Routing

Introduction

The Content Based Router (CBR) in the JBossESB can be used to route message to the next destination based on the content of the message. By default the ESB will use JBossRules as it's evaluation engine, however this is left configurable. In the jbossesb-properties.xml there is property called 'org.jboss.soa.esb.routing.cbrClass' in the 'messagerouting' section:

```xml
<properties name="messagerouting">
  <property name="org.jboss.soa.esb.routing.cbrClass"
    value="org.jboss.internal.soa.esb.services.routing.cbr.JBossRulesRouter"/>
</properties>
```

which can be used to plug in another evaluation engine.

**JBossRules based Router**

By default the Content Based router uses JBossRules. Read on to find how to bring up CBR, how to give it a specific rule set, and how to send a message to the CBR.

**Create a Content-Based Router**

To bring up a Content-based router you need to bring up the CBR-service. Currently this is done by bringing up ContentBasedRouting Service. To do this you need to add an xml-chapter to your deployment-config.xml that looks like the following example which uses JMS as transport protocol:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<jbossesb xmlns="http://anonsvn.labs.jboss.com/labs/jbossesb/trunk/product/etc/schemas/xml/jbossesb-1.0.xsd" parameterReloadSecs="10">
  <providers>
    <jms-provider name="localhost"
      connection-factory="ConnectionFactory"
      jndi-context-factory="org.jnp.interfaces.NamingContextFactory"
      jndi-URL="localhost" />
    <jms-bus busid="QueueA">
      <jms-message-filter
        dest-type="QUEUE"
        dest-name="queue/A"
        selector="service='CBRouting-SerializableOrXml'" />
    </jms-bus>
  </providers>
  <services>
    <service category="MessageRouting"
      name="ContentBasedRoutingService"
      
```
Define a service and add a ContentBasedRouting

Cbr specific properties to the (jms-)listener:

1. ruleSet: The set of rules that should be used to evaluate the content. Only 1 ruleSet can be given for each CBR. Just define more router if you need composite CBR routing.

2. ruleLanguage: In JBossRules you can define a custom (business) language. This (optional) field can be used to pass in that custom language to go with the defined ruleSet.

3. RuleReload: the rules will reload 'hot' if refreshed

4. A set of destinations. The destination-name is what will be used in the ruleSet. The service-category and service-name are then used to deliver the message.

**Create a ruleSet**

A rule set can be created using the JBossIDE which includes a plug-in for JbossRules. Figure 1 shows a screen shot the plug-in.
To add a new rule simple click on “New Rule resource”. Make sure your rules import at least

```java
import org.jboss.soa.esb.message.Message;
```

The message will be asserted into the working memory of the rules engine. You can use sub objects of message in your evaluation. This is demonstrated in the JBossESBRules.drl ruleSet where the rule checks if the type is XML or Serializable. Secondly you have to make sure to define:

```java
global java.util.List destinationServices;
```

The destinationServices List is returned and should contain one or more destination services. The format of the destination service should be:

```
<service-category>::<service-name>
```
If you save your rule into the product/core/services/rules package it will be jarred up into the jbossesb-rules.jar. However as long as you your rules on the classpath somewhere JBossRules should find it.

**Use the XPath Custom Rule Language**

For XML-based messages it is convenient to do XPath based evaluation. For this we ship a custom rule language defined in the XpathLanguage.dsl. To use it you need to reference it in your ruleSet with:

```drl
expander XPathLanguage.dsl
```

and you need to pass it in the name of this file in the attributes like:

```drl
ruleSet="JBossESBRules-XPath.drl"
ruleLanguage="XPathLanguage.drl" />
```

Currently the XPath Language makes sure the message is of the type JBOSS.XML and it defines

1. `xpathMatch "<element>"`: yields true if is an element by this name is matched.

2. `xpathEquals "<element>"", "<value>");`: yields true if the element is found and it's value equals the value.

You can define your own Custom Rule Language.

**Splitter**

To make a message go to more then 1 destination (split) simply set up your rules to match more then one destinations. You can also use the Static Router with multiple destinations to achieve this. In either the StaticRouter action or ContentBasedRouter action you can set the process="split", which will enable the Aggregator Action to aggregate the message further down to processing path.

Note that those destinations could be different message transformers to customize the outgoing messages.

**Aggregator**

The Aggregator action can be used to aggregate message that where split back into 1 message. This message will contain as many message attachments as the messages that where aggregated. The next action in the pipeline should be a transformer action to use the attachments to build a new aggregated message.