Progress FUSE™

FUSE™ Mediation Router

Using the Library

Version 1.6
April 2009

PROGRESS SOFTWARE

Using the Library

Version 1.6

Publication date 17 Jul 2009

Legal Notices

Table of Contents

List of Tables

Chapter 1. Resources

Document Conventions

Typographical conventions

This book uses the following typographical conventions:

Table 1.1. Typographical Conventions

`fixed width`	Fixed width (Courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `javax.xml.ws.Endpoint` class. Constant width paragraphs represent code examples or information a system displays on the screen. For example: `import java.util.logging.Logger;`
`Fixed width italic`	Fixed width italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example: `% cd /users/`*`YourUserName`*
Italic	Italic words in normal text represent *emphasis* and introduce *new terms*.
Bold	Bold words in normal text represent graphical user interface components such as menu commands and dialog boxes. For example: the **User Preferences** dialog.

Keying conventions

This book uses the following keying conventions:

Table 1.2. Keying Conventions

No prompt	When a command's format is the same for multiple platforms, the command prompt is not shown.
`%`	A percent sign represents the UNIX command shell prompt for a command that does not require root privileges.
`#`	A number sign represents the UNIX command shell prompt for a command that requires root privileges.
`>`	The notation > represents the MS-DOS or Windows command prompt.
`...`	Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.
`[ ]`	Brackets enclose optional items in format and syntax descriptions.

`{ }`	Braces enclose a list from which you must choose an item in format and syntax descriptions.
`|`	In format and syntax descriptions, a vertical bar separates items in a list of choices enclosed in `{ }` (braces).

Admonition conventions

This book uses the following conventions for admonitions:

Table 1.3. Admonitions

🗎	Notes display information that may be useful, but not critical.
💡	Tips provide hints about completing a task or using a tool. They may also provide information about workarounds to possible problems.
❗	Important notes display information that is critical to the task at hand.
▼	Cautions display information about likely errors that can be encountered. These errors are unlikely to cause damage to your data or your systems.
❌	Warnings display information about errors that may cause damage to your systems. Possible damage from these errors include system failures and loss of data.

Searching the Library

Overview

There are two ways you can search the FUSE Services Framework library:

- the FUSE source search field
- Google

FUSE source search

The FUSE source web site has an integrated search feature that will return hits from the entire Web site. Using it may return a number of non-documentation hits.

The FUSE source site search is acessed using the **Search** box that appears in the header of the main documentation site.

Google search

Use Google directly and narrow the search scope using `site:fusesource.com/docs/router`.

This method will limit the search to the FUSE Mediation Router documentation. You can limit the scope further by adding a version number to the `site:` string.

For example to search the documentation for references to using the `RotuerBuilder` class you would enter **`site:fusesource.com/docs/router/1.4 RotuerBuilder`** into the Google search field. Only results from the FUSE Mediation Router version 1.4 library will be returned.

Required Reading

Overview

The books in this library assumes that the reader is familiar with a number of key technologies including:

- Apache Maven
- The Spring Framework
- Ant

Apache Maven

Apache Maven is a plug-in based build system. The FUSE products have extensive Maven tooling to that assists in developing solutions. It is used extensively throughout the documentation.

To get more information on using Maven see the following:

- *Maven: The Definitive Guide*[1]
- *Developing with Eclipse and Maven*[2]
- The Maven Website[3]

The Spring Framework

The Spring Framework is an open source Java platform that aims to simplify developing complex Java applications. The FUSE products use the Spring Framework for several purposes.

For more information see the Spring Framework Reference[4].

Ant

Apache Ant is a popular Java build tool. It is used to build a number of the samples provided with the FUSE products. The FUSE products also include Ant-based development tools.

For more information about using Ant see the *Apache Ant User Manual*[5].

[1] http://www.sonatype.com/products/maven/documentation/book-defguide
[2] http://www.sonatype.com/books/m2eclipse-book/reference
[3] http://maven.apache.org/index.html
[4] http://static.springframework.org/spring/docs/2.0.x/reference/
[5] http://ant.apache.org/manual/index.html

Open Source Project Resources

Apache CXF	**Web site:** http://cxf.apache.org/ **User's list:** `<users@cxf.apache.org>`
Apache Tomcat	**Web site:** http://tomcat.apache.org/ **User's list:** `<users@tomcat.apache.org>`
Apache ActiveMQ	**Web site:** http://activemq.apache.org/ **User's list:** `<users@activemq.apache.org>`
Apache Camel	**Web site:** http://camel.apache.org **User's list:** `<users@camel.apache.org>`
Apache ServiceMix	**Web site:** http://servicemix.apache.org **User's list:** `<users@servicemix.apache.org>`

Chapter 2. Books

Overview

The FUSE Mediation Router library is divided into distinct sections that help make it easier to find the needed documentation.

Getting Started

The following books are intended to help new users become familiar with FUSE Mediation Router.

- Getting Started with the FUSE™ Product Family[1] provides an overview of the entire FUSE product family and explains how FUSE Mediation Router fits into the picture.
- The Logisticx Tutorial Guide[2] contains tutorials that show how FUSE Mediation Router can be used in conjunction with the other FUSE products to solve complex problems.
- The Glossary[3] provides definitions for commonly used terms relating to integration and using FUSE Mediation Router.
- *Getting Started* provides an overview of using FUSE Mediation Router and describes how to generate a simple FUSE Mediation Router application using Maven.

Installation

The following books discuss how to install FUSE Mediation Router and migrate from earlier versions:

- Release Notes[4]
- *Installation Guide* describes the requirements for installing FUSE Mediation Router.

[1] http://fusesource.com/docs/getting_started/index.html
[2] http://fusesource.com/docs/logistix/index.html
[3] http://fusesource.com/docs/glossary/index.html
[4] http://fusesource.com/wiki/display/ProdInfo/FuSE+Mediation+Router+v1.6+Release+Notes

- *Migrating to 1.6* describes things that need to be considered when migrating to 1.6 of FUSE Mediation Router.

Developing routes

The following books provide information on implementing routes:

- *Defining Routes* provides an introduction to defining routes using the Java fluent DSL and the Spring XML syntax.
- *Implementing Enterprise Integration Patterns* describes how you can use FUSE Mediation Router to implement *Enterprise Integration Patterns* (from the book of the same name by Gregor Hohpe and Bobby Woolf).
- *Programmer's Guide* provides more details of how to program routing processors and describes how to implement your own custom components.

Deploying routes

Deployment Guide describes how to deploy a FUSE Mediation Router application either into a Spring container or as a standalone application.

Reference

The follow resources provide basic reference material:

- *Component Reference* **[DRAFT]** provides a reference to most of the FUSE Mediation Router components.

Chapter 3. Reading Paths

Route Deployers

Overview

Defining and deploying routes are the central tasks in FUSE Mediation Router. A route is essentially a pipeline that accepts messages from a consumer endpoint, processes the messages, and then transmits the messages to a producer endpoint. The path described here helps you to learn the basic principles of defining routes.

Path

For a quick introduction to defining and deploying routes, read the following:

1. "Creating a Simple Content-Based Router" in *Getting Started* describes how to generate and deploy a simple route into the Spring container (based on Maven and Eclipse).
2. The next step is to learn the basic syntax of route definitions. There are two alternative approaches you can use:
   - "Defining Routes in Java DSL" in *Defining Routes* explains how to define routes using the Java DSL (Domain Specific Language).
   - "Defining Routes in XML" in *Defining Routes* explains how to define routes using Spring XML configuration.
3. "Basic Principles of Route Building" in *Defining Routes* explains the basic essentials of route building, including exception handling and bean integration.
4. "Message Routing" in *Implementing Enterprise Integration Patterns* provides in-depth descriptions of the standard processors you can insert into a route.

Next steps

The following references describe additional topics of interest to route deployers:

- HTTP Component[1] and Jetty Component[2], to integrate your routes with HTTP.

[1] http://activemq.apache.org/camel/http.html
[2] http://activemq.apache.org/camel/jetty.html

- CXF Component[3], to create Web services or CORBA endpoints for your routes.
- XQuery Component[4], JDBC[5], and JPA[6], to integrate your routes with a database.
- Velocity Component[7], StringTemplate Component[8], to process messages using a Velocity or a String template. Templating is a powerful and simple technique for reformatting messages, which frequently comes in useful when defining a route.

[3] http://activemq.apache.org/camel/cxf.html
[4] http://activemq.apache.org/camel/xquery-endpoint.html
[5] http://activemq.apache.org/camel/jdbc.html
[6] http://activemq.apache.org/camel/jpa.html
[7] http://activemq.apache.org/camel/velocity.html
[8] http://activemq.apache.org/camel/stringtemplate.html

Component Developers

Overview

FUSE Mediation Router provides a huge range of standard components that enable you to integrate your routes with a variety of transport protocols and third party tools. Occasionally, however, you might find it necessary to implement a custom component to support a non-standard integration. FUSE Mediation Router makes it relatively easy for you to design and implement your own custom components.

Path

For a quick introduction to developing a custom component, read the following:

1. "Introducing FUSE Mediation Router" in *Getting Started* introduces basic concepts and describes the FUSE Mediation Router architecture.
2. You can generate starting point code for developing a component using the `camel-archetype-component` Maven archetype. For detailed instructions on how to use the Maven archetypes in the Eclipse IDE, see "Creating a Simple Content-Based Router" in *Getting Started*.
3. Before starting to design your own component, it is a good idea to browse through the standard components in *Component Reference* and Component Reference[9].
4. "Understanding Message Formats" in *Programmer's Guide* provides a detailed description of the data types used to hold message data in FUSE Mediation Router.
5. "Implementing a Component" in *Programmer's Guide*, and the following chapters, provide detailed guidelines for developing your own custom component.

[9] http://activemq.apache.org/camel/components.html

FUSE ESB 4 Developers

Overview

FUSE Mediation Router can also be deployed into a 4 container. In fact, if you want to implement *Enterprise Integration Patterns* in , it is recommended that you use FUSE Mediation Router instead of the legacy implementation that was originally built into .

Path

For a quick introduction to defining and deploying routes in the context of FUSE ESB 4, read the following:

1. To get started with deploying router applications into , it is recommended that you read the relevant Getting Started guide from the documentation set on the http://fusesource.com Web site:
   - *Enterprise Integration Patterns (OSGi Container)*.
   - *Enterprise Integration Patterns (JBI Container)*.
2. "Basic Principles of Route Building" in *Defining Routes* provides some essential examples of how to define routes.
3. If you are migrating enterprise integration patterns from the legacy implementations to FUSE Mediation Router, you should read Appendix A in *Implementing Enterprise Integration Patterns*.