Progress
FUSE™

FUSE™ Services Framework

Using the Library

Version 2.2.x
April 2009

PROGRESS
S O F T W A R E

Using the Library

Version 2.2.x

Publication date 17 Jul 2009
Copyright © 2001-2009 Progress Software Corporation and/or its subsidiaries or affiliates.

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/framework`.

This method will limit the search to the FUSE Services Framework 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 `Oneway` annotation you would enter **`site:fusesource.com/docs/framework/2.1 Oneway`** into the Google search field. Only results from the FUSE Services Framework version 2.1 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

11

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 Services Framework 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 Services Framework.

- Getting Started with the FUSE™ Product Family[1] provides an overview of the entire FUSE product family and explains how FUSE Services Framework fits into the picture.

- The Logisticx Tutorial Guide[2] contains tutorials that show how FUSE Services Framework 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 Web services and using FUSE Services Framework.

- *Writing WSDL Contracts* provides an overview to writing logical interfaces using WSDL.

Installation

The following books discuss how to install FUSE Services Framework and migrate from earlier versions:

- Release Notes[4]

- *Installation Guide* describes the requirements needed to install FUSE Services Framework.

---

[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+Services+Framework+v2.2+Release+Notes

- *Migrating to 2.2.x* describes what need to be done when upgrading your applications to using FUSE Services Framework 2.2.x.

Developing services

The following books describe how to develop applications using FUSE Services Framework:

- *Developing Applications Using JAX-WS* provides detailed information on developing services using the JAX-WS APIs.

- *Developing RESTful Services* describes how to develop and deploy RESTful applications using FUSE Services Framework.

- *Developing Interceptors* provides detailed information on writing interceptors that can be added to the FUSE Services Framework message processing chain.

- *Developing Applications with JavaScript* describes how to develop and deploy distributed applications using JavaScript.

Configuring and deploying services

The following books describe how to configure the FUSE Services Framework runtime:

- *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's bindings and transports.

- *Configuring and Deploying Endpoints* provides detailed information on how to configure the FUSE Services Framework runtime and how to package applications for deployment into various containers.

Security

The *Security Guide* provides details on configuring the FUSE Services Framework to work in applications requiring security.

Integration with other platforms

J2EE Integrations provides detailed information on using FUSE Services Framework with JEE application servers.

Reference

The follow resources provide basic reference material:

- *Tool Reference* is a quick reference to FUSE Services Framework's command-line tools.

- API Documentation[5]

Further reading

In addition to the above books, you may also want to read the documentation for each of the components that FUSE Services Framework bundles. This documentation is available from the projects responsible for developing the component.

---

[5] http://fusesource.com/docs/framework/2.2/api/index.html

Chapter 3. Reading Paths

Service Consumer Developers

Overview

Developing a service consumer is, perhaps, the most basic task a user of FUSE Services Framework would perform. In general, consumer developers will be given a WSDL contract and need to generate the stubs needed to make requests on the remote service. This reading path provides a quick introduction to WSDL, a description of generating the JAX-WS stubs, and a discussion of how to configure your consumer for deployment.

Path

For a quick introduction to developing service consumers, you can read the following:

1. "Introducing WSDL Contracts" in *Writing WSDL Contracts* provides a quick overview of the contents of a WSDL document.

2. "Developing a Consumer From a WSDL Contract" in *Developing Applications Using JAX-WS* describes how to develop a service consumer using a provided WSDL document. It discusses how to use the command-line tools to generate the stubs and what additional code is required.

3. "Developing a Consumer Without a WSDL Contract" in *Developing Applications Using JAX-WS* describes how to develop a service consumer if you are provided a Java interface instead of a WSDL document.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Next steps

The following provide additional information about consumer development:

• "Developing Asynchronous Applications" in *Developing Applications Using JAX-WS* describes how to develop consumers that make asynchronous invocations on remote services.

- "Using XML in a Consumer" in *Developing Applications Using JAX-WS* describes how to use the `Dispatch` interface to develop consumers that work with raw XML messages.

- "Working with Contexts" in *Developing Applications Using JAX-WS* describes how to use the JAX-WS context mechanism.

- *Developing Interceptors* provides detailed information on writing interceptors that can be added to the FUSE Services Framework message processing chain.

- "Using SOAP 1.1 Messages" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's SOAP 1.1 binding.

- "Configuring a Consumer" in *Using the Bindings and Transports* describes how to configure consumer-side HTTP endpoints.

- "Using the HTTP Transport in Decoupled Mode" in *Using the Bindings and Transports* describes how to configure a consumer to use a decoupled HTTP conduit.

- "Using JMS" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's JMS transport.

- "Using XML Documents" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's XML binding.

Java-First Service Developers

Overview

One of the strengths of the JAX-WS specification is that it provides a way to develop services without using WSDL. In Java first development an annotated interface acts as the contract. This reading path walks you through documentation concerning the creation of a service endpoint interface, the code needed to deploy the service as a standalone application, and configuring the runtime.

Path

For a quick introduction to developing a service without using WSDL, you can read the following:

1. "Creating the SEI" in *Developing Applications Using JAX-WS* describes how to develop a SEI for your service.

2. "Annotating the Code" in *Developing Applications Using JAX-WS* describes the annotations used by the JAX-WS framework.

3. "Publishing a Service" in *Developing Applications Using JAX-WS* describes the code needed to publish a service as a standalone application.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Next steps

The following provide additional information about consumer development:

• "Deploying to the Spring Container" in *Configuring and Deploying Endpoints* describes how to deploy a service into the FUSE Services Framework Spring container.

• "Deploying to a Servlet Container" in *Configuring and Deploying Endpoints* describes how to deploy a service into a servlet container.

- "Using XML in a Service Provider" in *Developing Applications Using JAX-WS* describes how to use the `Provider` interface to develop services that work with raw XML messages.

- "Working with Contexts" in *Developing Applications Using JAX-WS* describes how to use the JAX-WS context mechanism.

- *Developing Interceptors* provides detailed information on writing interceptors that can be added to the FUSE Services Framework message processing chain.

- "Using SOAP 1.1 Messages" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's SOAP 1.1 binding.

- "Sending Binary Data with SOAP MTOM" in *Using the Bindings and Transports* provides detailed information on sending binary data using the FUSE Services Framework's SOAP binding.

- "Configuring a Service Provider" in *Using the Bindings and Transports* describes how to configure server-side HTTP endpoints.

- "Using JMS" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's JMS transport.

- "Using XML Documents" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's XML binding.

WSDL-First Service Developers

Overview

WSDL is an XML grammar used to describe services. In many instances, the first step in developing a service is describing it in WSDL. The Java code that implements the actual service is generated from the WSDL document. This reading path walks you through the process of creating a WSDL document describing a service, generating the required skeleton code, implementing the service, and deploying the service as a standalone application.

Path

For a quick introduction to developing a service using WSDL, you can read the following:

1. "Introducing WSDL Contracts" in *Writing WSDL Contracts* describes the contents of a WSDL document.

2. "Defining Logical Data Units" in *Writing WSDL Contracts* describes how to define complex data types in XML Schema.

3. "Defining Logical Messages Used by a Service" in *Writing WSDL Contracts* describes how to define the messages exchanged by your service.

4. "Defining Your Logical Interfaces" in *Writing WSDL Contracts* describes how to define your service's interface.

5. "Using SOAP 1.2 Messages" in *Using the Bindings and Transports* describes how to add a SOAP 1.2 binding to a WSDL document.

6. "Adding a Basic HTTP Endpoint" in *Using the Bindings and Transports* describes how to add an HTTP port to a WSDL document.

7. "Top-Down Service Development" in *Developing Applications Using JAX-WS* describes how to implement your service.

8. "Publishing a Service" in *Developing Applications Using JAX-WS* describes the code needed to publish a service as a standalone application.

9. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

10. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Next steps

The following provide additional information about consumer development:

• "Deploying to the Spring Container" in *Configuring and Deploying Endpoints* describes how to deploy a service into the FUSE Services Framework Spring container.

• "Deploying to a Servlet Container" in *Configuring and Deploying Endpoints* describes how to deploy a service into a servlet container.

• "Working with Contexts" in *Developing Applications Using JAX-WS* describes how to use the JAX-WS context mechanism.

• *Developing Interceptors* provides detailed information on writing interceptors that can be added to the FUSE Services Framework message processing chain.

• "Sending Binary Data with SOAP MTOM" in *Using the Bindings and Transports* provides detailed information on sending binary data using the FUSE Services Framework's SOAP binding.

• "Configuring a Service Provider" in *Using the Bindings and Transports* describes how to configure server-side HTTP endpoints.

• "Using JMS" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's JMS transport.

• "Using XML Documents" in *Using the Bindings and Transports* provides detailed information on using the FUSE Services Framework's XML binding.

RESTful Service Developers

Overview

Using FUSE Services Framework you can develop applications that conform to the principles of REST-based design. FUSE Services Framework allows you to develop RESTful services in three ways:

• Using the automatically generated URLs.

• Using FUSE Services Framework specific annotations to generate the URLs.

• Using the JAX-WS `Provider` interface.

Using automatic URL generation

For a quick introduction to developing RESTful services with automatically generated URLs, you can read the following:

1. "Introduction to RESTful Services" in *Developing RESTful Services* provides an overview of RESTful design principles.

2. "Using Automatic Mappings" in *Developing RESTful Services* describes how to develop a service that uses the FUSE Services Framework to automatically generate your URIs.

3. ???? describes how publish your service as a standalone application.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Using annotations

For a quick introduction to developing RESTful services with FUSE Services Framework annotations, you can read the following:

1. "Introduction to RESTful Services" in *Developing RESTful Services* provides an overview of RESTful design principles.

2. "Using Annotations" in *Developing RESTful Services* describes how to use the FUSE Services Framework annotations.

3. ???? describes how publish your service as a standalone application.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Using annotations

For a quick introduction to developing RESTful services with FUSE Services Framework annotations, you can read the following:

1. "Introduction to RESTful Services" in *Developing RESTful Services* provides an overview of RESTful design principles.

2. "Using XML in a Service Provider" in *Developing Applications Using JAX-WS* describes how to develop a service using the JAX-WS `Provider` interface.

3. "Publishing a Service" in *Developing Applications Using JAX-WS* describes the code needed to publish a service as a standalone application.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

JavaScript Developers

Service Developers

Overview

FUSE Services Framework includes a mechanism for developing services using JavaScript. The JavaScript mechanism mimics the use of the JAX-WS `Provider` interface. This path walks you through the process of developing and deploying a service using the JavaScript interface.

Path

For a quick introduction to developing a service in JavaScript, you can read the following:

1. "Defining the Metadata" in *Developing Applications with JavaScript* describes how to add provide the metadata needed to publish a service to your application.

2. "Implementing the Service Logic" in *Developing Applications with JavaScript* describes how to implement your service.

3. "Publishing Services" in *Developing Applications with JavaScript* describes how to publish your service.

4. "Setting Up Your Environment" in *Configuring and Deploying Endpoints* describes how to set up your environment to run a FUSE Services Framework application.

5. "FUSE Services Framework Configuration Files" in *Configuring and Deploying Endpoints* provides details about the files used to configure the FUSE Services Framework runtime.

Consumer Developers

Overview

FUSE Services Framework includes a mechanism for developing consumers using JavaScript. The JavaScript mechanism uses some of the features of modern Web browsers. This path walks you through the process of developing and deploying a service using the JavaScript interface.

Path

For a quick introduction to developing a consumer in JavaScript, you can read the following:

1. "Generating Consumer Code" in *Developing Applications with JavaScript* describes how to generate the stub code for implementing a consumer.

2. "Implementing the Callbacks" in *Developing Applications with JavaScript* describes how to implement the callbacks used by a JavaScript consumer.

3. "Invoking Operations on a Service" in *Developing Applications with JavaScript* describes how to invoke remote services from a JavaScript consumer.

4. "Running Clients in a Browser" in *Developing Applications with JavaScript* describes how to run your consumer.