

The logo features the word "FusionSource" in a white, sans-serif font, oriented vertically on a black background. The black background is a vertical strip on the left side of the page, with a jagged, torn-paper edge on its right side. Behind this strip, a red and white abstract pattern is visible, resembling a stylized flame or a torn piece of paper.

FusionSource

Fuse ESB Enterprise **Glossary**

Integration Everywhere

Glossary

Updated: 15 Mar 2012

Copyright © 2011 FuseSource Corp. All rights reserved.

Legal Notices

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

Actional, Apama, Apama (and Design), Artix, Business Empowerment, DataDirect (and design), DataDirect Connect, DataDirect Connect64, DataDirect Technologies, DataDirect XML Converters, DataDirect XQuery, DataXtend, Dynamic Routing Architecture, EdgeXtend, Empowerment Center, Fathom, FUSE Mediation Router, FUSE Message Broker, FUSE Services Framework, IntelliStream, IONA, Making Software Work Together, Mindreef, ObjectStore, OpenEdge, Orbix, PeerDirect, POSSENET, Powered by Progress, PowerTier, Progress, Progress DataXtend, Progress Dynamics, Progress Business Empowerment, Progress Empowerment Center, Progress Empowerment Program, Progress OpenEdge, Progress Profiles, Progress Results, Progress Software Developers Network, Progress Sonic, ProVision, PS Select, Savvion, SequeLink, Shadow, SOAPscope, SOAPStation, Sonic, Sonic ESB, SonicMQ, Sonic Orchestration Server, SonicSynergy, SpeedScript, Stylus Studio, Technical Empowerment, WebSpeed, Xcalia (and design), and Your Software, Our Technology—Experience the Connection are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries. AccelEvent, Apama Dashboard Studio, Apama Event Manager, Apama Event Modeler, Apama Event Store, Apama Risk Firewall, AppsAlive, AppServer, ASPen, ASP-in-a-Box, BusinessEdge, Business Making Progress, Cache-Forward, DataDirect Spy, DataDirect SupportLink, FUSE, FuseSource, Future Proof, GVAC, High Performance Integration, ObjectStore Inspector, ObjectStore Performance Expert, OpenAccess, Orbacus, Pantero, POSSE, ProDataSet, Progress Control Tower, Progress ESP Event Manager, Progress ESP Event Modeler, Progress Event Engine, Progress RFID, Progress RPM, Progress Software Business Making Progress, PSE Pro, SectorAlliance, SeeThinkAct, Shadow z/Services, Shadow z/Direct, Shadow z/Events, Shadow z/Presentation, Shadow Studio, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Sonic Business Integration Suite, Sonic Process Manager, Sonic Collaboration Server, Sonic Continuous Availability Architecture, Sonic Database Service, Sonic Workbench, Sonic XML Server, The Brains Behind BAM, WebClient, and Who Makes Progress are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Any other trademarks contained herein are the property of their respective owners.

Third Party Acknowledgments

Fuse ESB v4.3.1 incorporates ASM v3.1 from INRIA, France Telecom. Such technology is subject to the following terms and conditions: ASM: a very small and fast Java bytecode manipulation framework Copyright (c) 2000-2005 INRIA, France Telecom All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Fuse ESB v4.3.1 incorporates HSQL Database Engine v1.7.3.3 from Hypersonic SQL Group. Such technology is subject to the following terms and conditions: Hypersonic SQL License Copyright Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes Hypersonic SQL." Products derived from this software may not be called "Hypersonic SQL" nor may "Hypersonic SQL" appear in their names without prior written permission of the Hypersonic SQL Group. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes Hypersonic SQL." This software is provided "as is" and any expressed or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Hypersonic SQL Group or its contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption). However caused any on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage. This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

Fuse ESB v4.3.1 incorporates JBoss.org – Javassist v3.9.0.GA, The contents of these files are subject to the Mozilla Public License Version 1.1 (the "License") provided below; you may not use this file except in compliance with the License. You may also obtain a copy of the License at <http://www.mozilla.org/MPL/>. Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. For Javassist v3.9.0.GA, The Original Code is Javassist. The Initial Developer of the Original Code is Shigeru Chiba. Portions created by the Initial Developer are Copyright (C) 1999-2008 Shigeru Chiba. All Rights Reserved.

Fuse ESB v4.3.1 incorporates ANTLR, Another Tool for Language Recognition v2.7.7 from Terence Parr. Such technology is subject to the following terms and conditions: SOFTWARE RIGHTS ANTLR 1989-2006 Developed by Terence Parr Partially supported by University of San Francisco & jGuru.com. We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commercial software. We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed. The primary ANTLR guy: Terence Parr <parrt@cs.usfca.edu> <parrt@antlr.org>.

Table of Contents

A	7
B	9
C	11
D	13
E	15
F	17
I	19
J	21
L	23
M	25
N	27
O	29
P	31
Q	33
R	35
S	37
T	39
U	41
V	43
W	45
X	47

A

abstract contract	See logical contract .
abstract head element	An XML Schema element that cannot appear in an instance document. When a substitution group's head element is declared as abstract with <code>abstract="true"</code> , a member of that element's substitution group must be used instead.
advisory	See advisory message .
advisory message	A special type of message that contains administrative information about the message broker. They are sent by the broker to special advisory topics. See Also advisory topic .
advisory topic	A group of special topics that are created by a message broker that are used for monitoring the state of the broker. The broker sends messages about a variety of internal broker events. Clients subscribing to these topics receive advisory messages about these objects.
agent	See fabric agent .
anyType	The root type for all XML Schema type definitions hierarchies. All primitive types are derivatives of this type, as are all user-defined complex types.
application server	A software platform that provides the services and infrastructure required to develop and deploy middle-tier applications. Middle-tier applications implement the business logic necessary to provide web clients with access to enterprise information systems. In a multi-tier architecture, an application server sits beside a web server or between a web server and enterprise information systems. Application servers provide the middleware for enterprise systems. JBoss, WebLogic and WebSphere are J2EE application servers.

B

binding	A description of the message format and protocol details for a set of operations and messages. Bindings are created based on the information specified in a WSDL <code>binding</code> element.
binding component	A JBI component that provides connectivity to services external to the JBI environment.
Blueprint	A dependency injection framework designed for use in an OSGi container. It is governed by the Blueprint Container Specification in the OSGi Service Platform Release 4 Version 4.2 Enterprise Specification. See Also dependency injection .
bundle	The primary deployment format used in Fuse ESB Enterprise. They are either ZIP or JAR files that contain resources and classes for providing a set of functionality to other bundles or to the end user. Bundles differ from standard JAR files in that they must contain metadata describing the bundle and its dependencies. See Also Fuse Application Bundle .

C

CamelContext	A single routing rule base that defines the context for configuring routes, and specifies which policies to use during message exchanges between endpoints.
choice complex type	An XML Schema construct defined using the <code>choice</code> element to constrain the possible elements in a complex data type. When using a choice complex type, only one of the elements defined in the complex type can be present at a time.
client	An application or process that requests services from other applications known as servers. The server processes may be running on the same or a different machine. In the context of a SOA network, a client process is called a consumer or service consumer. messaging—An application that uses the message broker to communicate with other applications. These applications use one of the broker's client API to connect to and interact with the broker.
cluster	A collection of clustered services. See Also clustered service .
clustered service	A service that can be discovered via Fuse Fabric and has master/slave support.
component	A factory that creates a routing endpoint that connects to a particular message source or message sink.
composite destination	A virtual destination that serves as a proxy for multiple destinations. Producers can send messages to the composite destination and it will be automatically sent to all of the physical destinations that make up the composite destination. See Also virtual destination .
concrete contract	See physical contract .
connection	A bridge between a client and a broker connector. See Also transport connector , network connector , network of brokers .
connection factory	An object that a client uses to create a connection to a broker. A factory supports attributes that configure the quality of service for the connections it creates.
connector	An object that connects clients to a broker.

consumer	<p>services—The end user of a service, also called a client for that service. The more exact term in the context of a service-oriented network is service consumer.</p> <p>messaging—An application that consumes messages from a messaging destination.</p> <p>routing—The source of messages in a route.</p>
contract	<p>A description of the messages and formats accepted and generated by a service. A service's contract is specified in a WSDL document that defines the interface and all connection-related information for that interface. A WSDL contract contains two sets of components: logical (or abstract) and physical (or concrete).</p> <p>The logical components of the contract are those that describe the data types, message formats, operations, and interfaces for the services defined in the contract. Logical components are specified with the WSDL elements <code>types</code>, <code>message</code>, <code>portType</code>, and <code>operation</code>.</p>

D

dead letter channel	An EIP processor that handles messages that cannot be delivered to the intended recipient.
dead letter queue	A special destination used by the message broker to hold undeliverable messages.
dependency injection	A form of inversion of control, where an object's external dependencies are given to it, either programmatically or through a framework that is driven by configuration information. The result is to decouple dependent objects and allow the dependencies to be resolved at run time.
destination	A logical holding area for messages in a message broker. Clients publish messages to and consume messages from destinations. See Also queue , topic .
discovery agent	A mechanism that advertises the list of available message brokers to message clients and other message brokers. See Also dynamic discovery .
durable subscriber	A message consumer that receives all messages published on a topic, including those published while the subscriber is inactive.
dynamic discovery	A mechanism for clients to become aware of the existence of brokers through the use of a discovery agent. See Also discovery agent .

E

endpoint	services—The point of contact that a service provides for its consumers. routing—The sources and sinks of messages in a route.
endpoint reference (EPR)	A self-contained object that describes the network contact and policy information for an endpoint, as defined in the WS-Addressing standard.
Enterprise Application Integration (EAI)	Enterprise Application Integration (EAI), the use of software and architectural principles to integrate disparate enterprise applications.
Enterprise Integration Patterns	A collection of patterns describing common EAI problems. For more information see http://www.enterpriseintegrationpatterns.com/ .
Enterprise Management System (EMS)	Enterprise Management System(EMS), a set of integrated tools that enable system administrators to manage large-scale production environments.
Enterprise Service Bus (ESB)	The infrastructure that allows service providers and service consumers to interact in a distributed environment. The bus handles the delivery of messages between different middleware systems, and provides management, monitoring, and mediation services such as routing, service discovery, or transaction processing.
exclusive consumer	A mechanism that ensures that only one consumer connected to a queue can consume messages.

F

fabric agent	The service running inside a container that is responsible for configuring and provisioning the container according to the profiles assigned to the container in the fabric registry.
facet	A rule in an XML Schema definition used in the derivation of user-defined simple types. Common facets include length, pattern, totalDigits, and fractionDigits.
failover	<p>A transport that automatically moves to a new connection in the event that its current connection fails.</p> <p>A cluster architecture where clients are able to migrate from a failed broker to a running broker.</p>
fault message	<p>A message containing error or exception information passed between a service and its consumers. Fault messages are defined using the <code>fault</code> element in a WSDL document.</p> <p>See Also request-response operation, solicit-response operation.</p>
feature	A unit of OSGi deployment that enables you to deploy multiple bundles in a single step.
feature repository	An XML file that defines one or more features.
feature URL	A URL that points to a feature repository file.
from	The DSL command that creates a message source for a route.
Fuse Application Bundle (FAB)	A bundle that uses a POM file to specify its dependencies.
Fuse Fabric	An open source project that provides the technology layer used to provide configuration and deployment services to a collection of distributed containers. It also allows for discovery of endpoints and messaging destinations across all of the containers in the fabric.

I

i18n	An abbreviation for internationalization, used in the context of preparing products, especially software and documentation, for use in more than one national locale and language.
in message	The message being processed by the processors in the route.
input message	<p>A message passed from a service consumer to a service. When mapped into Java, the parts of an input message are mapped into a method's parameter list. Input messages are defined using the <code>input</code> element in a WSDL document.</p> <p>See Also request-response operation, solicit-response operation, one-way operation.</p>
interface	<p>The external touch point between applications to collaborate or share functional behavior. Interfaces are completely described by the combination of logical and physical portions of a WSDL document.</p> <p>Once defined in a contract, an interface is the abstract boundary that a service exposes. A service's interface is the set of message types and message exchange patterns through which service consumers can interact with that service. In a WSDL 1.0 document, interfaces are defined using the <code>portType</code> element.</p>
intermediary	A service whose main role is to process all received messages in a value-added way, such as converting them from one data format to another, or routing them to another service. An intermediary has characteristics of both a service provider and a service consumer. Most intermediaries have an intermediary contract, which is similar in form to a service contract, except that it includes rules for processing messages.

J

Java API for RESTful Services (JAX-RS)	A standardized set of APIs and annotations designed to simplify the creation of Web applications using REST architectural principles.
Java API for XML Web Services (JAX-WS)	A document centric API for Web services. It was designed to take the place of JAX-RPC in Web services and Web applications.
Java Architecture for XML Binding (JAXB)	An API that provides a way to bind an XML Schema to a representation in Java code.
Java Authentication and Authorization Service (JAAS)	A Java security framework for user-centric security to augment the Java code-based security.
Java Business Integration (JBI)	A specification for a standards-based, vendor-neutral architecture, based on SOA principles, for the integration of disparate applications, service providers, and service consumers. JBI-compliant components are expected to plug in and interoperate with other JBI-compliant components. This frees vendors to concentrate on supplying components that implement their particular area of expertise without worrying about implementing the other necessary portions of a complete solution. JBI also frees end-users to pick and choose among many JBI-compliant components to assemble a SOA network sized to their needs, without locking in to one vendor's approach. The JBI specification was developed by the Java Community Process. Compare with SCA.
Java Database Connectivity (JDBC)	An API specified in Java technology that provides Java applications with access to databases and other data sources.
Java Management eXtensions (JMX)	A Java technology that supplies tools for managing and monitoring applications, system objects, devices, and service-oriented networks.
Java Message Service (JMS)	A Java API implementing a messaging standard that allows application components based on J2EE to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous.
Java Naming and Directory Interface (JNDI)	A set of APIs specified in Java technology that assists Java applications with interfacing to multiple naming and directory services.
Java Platform, Enterprise Edition 5 (JEE)	A specification and toolkit from Oracle for the development and deployment of enterprise applications. JEE is the Java 5 version of J2EE.
JavaScript Object Notation (JSON)	A text-based, human-readable format for representing simple data structures JSON is specified in RFC 4627. The official Internet media type for JSON is

`application/json`. JSON format is often as an alternative to XML for the serialization and transmission of structured data over a network connection.

L

l10n	An abbreviation for localization, used in the context of preparing products, especially software and documentation, for use in more than one national locale and language. Localization is the process of translating the elements of a product for a particular locale and language.
list type	A data type defined in an XML Schema definition as a space-separated list of primitive type elements, defined using the <code>xsd:list</code> element.
logical contract	<p>The abstract portion of a WSDL document that defines the data types, message types, and the interfaces for the services defined in the contract. The logical contract answers questions such as:</p> <ul style="list-style-type: none">• What kinds of data will this service work with?• What kinds of data are grouped together for processing?• What operations are related and what are their interfaces? <p>WSDL elements used in the logical contract include: <code>portType</code> element, <code>operation</code> element, <code>message</code> element, and <code>types</code> element. Compare with physical contract.</p>

M

marshalling	<p>The process of taking in-memory objects and converting them to a binary or textual format for transmission over a transport. See Also unmarshalling.</p>
master/slave	<p>A topology in which a single instance, the master, is active and one or more instances, the slaves, are ready to resume when the active instance stops.</p>
message	<p>messaging—An atomic unit of data that is passed between two or more clients. A message consists of three components:</p> <ul style="list-style-type: none">• headers—contain a predefined set of metadata that is used to communicate information about a message between the different parties that handle the message• properties—contain application defined metadata about a message to the different parties that handle the message• body—contains the messages payload <p>routing—The data passed processed by a route. A message consists of three components:</p> <ul style="list-style-type: none">• headers—contain metadata that is used to communicate information about a message between the different processors that handle the message• attachments—contain binary data that is associated with the message• body—contains the messages payload <p>services—Any data passed between a service provider and a service consumer, or between two endpoints. Messages are defined in using the WSDL <code>message</code> element. See Also fault message, input message, output message.</p>
message exchange pattern (MEP)	<p>The pattern of messages used by an application. There are two major message exchange patterns:</p> <ul style="list-style-type: none">• request-response—one client sends a message and expects a message to be returned• one-way—a client sends a message without expecting a response

The WSDL specification defines a number of more detailed MEPs that are all variations of the two basic patterns.
See Also [request-reply pattern](#).

message group

A collection of JMS messages that are assigned the same `JMSXGroupID`.

When used in conjunction with the `JMSXGroupSeq` message groups can be used to ensure that messages are processed in the proper sequence.

message selector

A string containing a boolean SQL statement using SQL 92 syntax that is used to select messages based on JMS message header properties.

middleware

A software communications layer that manages the interaction of disparate applications across heterogeneous hardware and network environments.

N

network bridge	A runtime directional link between brokers that is used to forward messages. Network bridges are created by network connectors. See Also network connector .
network connector	A configuration entity used to link brokers together to form a network of brokers. See Also network of brokers .
network of brokers	A group of brokers that are linked together to operate as a single logical unit.
nilable	In an XML Schema definition, an attribute of an element that specifies that the element is optional within a complex type.
normalized message router (NMR)	Part of the JBI architecture responsible for receiving message exchanges from JBI components and routing them to the appropriate component for processing. ...(New ESB)
notification operation	One type of WSDL-defined abstract operation, in which the service endpoint sends a message, but does not expect a return message.

O

OASIS	An international consortium that drives the development, convergence, and adoption of Web services standards. See www.oasis-open.org .
one-way operation	One type of WSDL-defined abstract operation, in which the service endpoint receives a message, but does not provide a return message. One-way operations specify only input message types.
operation	<p>A message interaction between a service and a service consumer. The WSDL specification provides for four types of operations:</p> <ul style="list-style-type: none">• one-way operation on page 29• request-response operation on page 35• solicit-response operation on page 37• notification operation on page 27
OSGi	<p>OSGi is set of open specifications aimed at making it easier to build and deploy complex software applications. The key piece of OSGi technology is the OSGi Framework. It defines standardized mechanism for packaging and managing application bundles. It can dynamical resolve dependencies between bundles and can handle having multiple versions of a bundle deployed simultaneously.</p> <p>The OSGi specifications are maintained by the OSGi Alliance. See http://www.osgi.org.</p>
out message	A temporary holder for .
output message	<p>A message passed from a service provider to a service consumer. When mapped into Java, the parts of an output message are mapped to a method's output parameter list, including any return value. Output messages are defined using the <code>output</code> element in a WSDL contract.</p> <p>See Also request-response operation, solicit-response operation, notification operation.</p>

P

participant	A member of a SOA network, whether service provider, service consumer, or intermediary.
payload format	<p>The on-the-wire structure of messages over a given transport.</p> <p>In Web services, the payload format is specified using a WSDL <code>binding</code> element.</p>
persistent identifier (PID)	A registration property used by the OSGi Configuration Admin Service to identify a group of configuration attributes.
physical contract	<p>The concrete portion of a WSDL contract that defines the bindings and transport details used by the services defined by that contract. The physical contract answers questions such as:</p> <ul style="list-style-type: none">• How is message traffic formatted on the wire?• How and where does message traffic travel?• Is there more than one option for transmitting a request? <p>WSDL elements used in the physical contract include: <code>binding</code> element, <code>service</code> element, <code>operation</code> element, and <code>port</code> element.</p>
point-to-point messaging	A messaging style where messages are sent between two known endpoints. This messaging style is typically implemented using queues.
port	<p>The physical mechanism used to access a service.</p> <p>In Web services, ports are created based on the information specified in a WSDL <code>port</code> element.</p>
processor	A Java object that performs work on a message as it passes along a route. Processors typically perform tasks like modifying the contents of a message or determining its path through a route.
producer	An application that creates messages and posts them to a messaging destination.
profile	A set of data that defines runtime artifacts and configuration settings for provisioning a container.

publish and subscribe messaging
(pub/sub)

A messaging style where message producers send(publish) messages to a destination and interested consumers can register(subscribe) to receive messages from the destination. This style of messaging is implemented using topics.

Q

QName

Industry-standard abbreviation for qualified name, as defined in the XML namespace specification. A QName is resource name that incorporates the namespace of the specification where that resource is defined.

QNames are composed of:

- A URI representing the namespace of the resource's definition.
- The name of the resource, usually called the *localPart*.
- Some QName formats also include an alias for the namespace called the prefix.

QNames can be found in several formats. The canonical format for QNames in Fuse ESB Enterprise is the one specified in `javax.xml.namespace.QName`, which is the namespace URI enclosed in braces, followed immediately (with no punctuation) by the *localPart*. For, example:
`{http://fusesource.com/demo}SOAPHTTPService`.

Another format is used in a self-contained document such as a WSDL contract, where a qualified name is in the form `prefix:localPart`. The prefix is declared in an `xmlns` element in an XML namespace declaration in the same document. For example, `Is:SOAPHTTPService` is a qualified name, where the prefix `Is` is defined in the statement `xmlns:Is="http://www.iona.com/FixedBinding"` earlier in the same document, and `SOAPHTTPService` is a resource defined in the specification at that location.

queue

A destination that uses first in/first out semantics.
See Also [destination](#).

R

reply	A message returned by a service to a service consumer in response to a request from that consumer. See Also output message .
Representational State Transfer (REST)	An architectural style for services based on Roy Fielding's doctoral dissertation. REST takes the view that services can be fully implemented using the concepts encapsulated in the design of the Web. A service's operations are handled as if they were resources addressed by a URI. Requests are made using one of the four simple HTTP verbs: GET, PUT, POST, and DELETE.
request	A message sent from a service consumer to a service provider asking for the service to perform an action. See Also input message .
request-reply pattern	A messaging pattern in which a message producer receives a message and returns a correlated message. See Also request-response operation .
request-response operation	One type of WSDL-defined abstract operation, in which the service endpoint receives a message and returns a correlated message. Request-response operations specify input message on page 19 , output message on page 29 , and fault message on page 17 types.
resource	On the Web a resource is anything that can be identified using a URI. When developing RESTful services, a resource is a class or method that implements a piece of the application's functionality.
response	See reply .
RESTful service	A service provider implemented using RESTful principles. See Also Representational State Transfer , Java API for RESTful Services .
retroactive consumer	A consumer that indicates to the topic that every attempt is to be made to send messages that the consumer may have missed.
route	A chain of processors through which a message travels.

S

service assembly	A collection of service units .
Service Component Architecture (SCA)	A set of specifications that describe a model for building applications and systems using a Service-Oriented Architecture. SCA extends and complements prior approaches to implementing services, and SCA builds on open standards such as Web services. SCA is developed by a consortium of companies. Compare with JBI.
service consumer	See consumer .
service engine	A JBI component that provides business logic and transformation services and also consumes such services.
service provider	A process or application that can respond to requests from a service consumer.
service unit	Artifacts deployed to a JBI component. A service unit configures the component to provide a piece of functionality such as expose an endpoint or route messages.
Service-Oriented Architecture (SOA)	A loosely-coupled distributed architecture in which service providers make resources available to service consumers in a standardized way. SOA is language and protocol independent.
Session	A JMS object that provides a single-threaded context for producing and consuming messages. JMS clients use the <code>Session</code> object to create producers, consumers, messages, and other artifacts used to work with messages.
solicit-response operation	One type of WSDL-defined abstract operation, in which the service endpoint sends a message and receives a correlated message.
Spring framework	A comprehensive programming and configuration model for modern Java-based enterprise applications the uses dependency injection. See Also dependency injection .
store and forward	A paradigm in which brokers receive messages, store them locally, and forwards the message to a recipient when it is able to do so. The message is only deleted once it has been successfully delivered.
Streaming Text Orientated Messaging Protocol (STOMP)	A language agnostic, simple text-based protocol that allows clients to talk with any message broker supporting the protocol.

substitution group

A feature of XML Schema that allows you to define groups of elements that may be used interchangeably in instance documents. For example, a `vehicle` head element might be defined with `automobile`, `boat`, and `airplane` substitution elements, any of which could be used wherever the `vehicle` element might be used. A substitution group is defined using the `substitutionGroup` attribute of the XML Schema element. See Also [abstract head element](#).

T

topic	A destination that uses publish and subscribe semantics. See Also destination .
transport	A standards-based network protocol, such as HTTP or STOMP, that defines how objects communicate over a network. In Web services, the transport details for an endpoint are specified inside the WSDL <code>port</code> element.
transport connector	An address at which a message broker accepts client connections.
transport mediation	The capability to move a message from one transport to another. This includes transforming message data between the formats required by each protocol and managing the metadata differences between the transports. It also means managing the differences in how the protocols operate. For example, when mediating between HTTP and JMS the bridge must manage the differences between the HTTP transports synchronous, request/reply style and the JMS transports asynchronous style.

U

- Uniform Resource Identifier (URI) A string of characters used to identify or name a resource on the Internet.
- unmarshalling The process of taking a binary or textual format payload and converting that into objects.
See Also [marshalling](#).

V

version	A collection of configuration profiles in a fabric. See Also profile .
virtual destination	A logical destination that represents one or more physical destinations. See Also composite destination , virtual topic .
virtual topic	A logical topic that allows consumers to use a physical queue to consume messages from the destination. See Also virtual destination , topic , queue .

W

Web service	An open set of standards for how systems connect to each other and communicate information. The standards are based on a distributed computing framework and provide a facility for applications or systems to collaborate regardless of location, hardware, or other implementation details.
Web Services Addressing (WS-A, WS-Addressing)	A specification that provides transport-neutral mechanisms to address Web services and messages. See the WS-Addressing specification ¹ .
Web Services Description Language (WSDL)	An XML grammar for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. WSDL is the language used to express service contracts. For further information see the WSDL specification ² .
Web Services Reliable Messaging (WS-RM)	A specification that describes a protocol that allows messages to be delivered reliably between distributed applications in the presence of software component, system, or network failures.
Web Services Security (WSS)	An OASIS specification that describes enhancements to SOAP messaging to provide a means for applying security to Web services. For further details, see the WSS specification ³ .
wiretap	An EIP in which an messages are directed to an additional message channel in addition to the primary channel.
World Wide Web Consortium (W3C)	An international consortium where member organizations, a full-time staff, and the public work together to develop Web standards.

¹ <http://www.w3.org/Submission/ws-addressing/>

² <http://www.w3.org/TR/wsdl>

³ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

X

XML Schema

A language specification by the W3C that defines an XML vocabulary for defining the contents and structure of XML documents. XML Schema is a successor to XML Document Type Declarations (DTDs), but is more expressive and better designed for expressing a type system.

XML Schema Definition (XSD)

An instance of an XML schema written in the XML Schema language. An XSD defines a type of XML document in terms of constraints upon what elements and attributes may appear, their relationship to each other, and what types of data may be in them.

