Red Hat JBoss A-MQ 6.2

Glossary

A reference to the terms used when talking about Red Hat JBoss A-MQ

Last Updated: 2018-06-15

Red Hat JBoss A-MQ 6.2 Glossary

A reference to the terms used when talking about Red Hat JBoss A-MQ

JBoss A-MQ Docs Team
Content Services
fuse-docs-support@redhat.com

Legal Notice

Abstract

This guide defines a number of terms that are specific to Red Hat JBoss A-MQ and the messaging and integration space.

Table of Contents

CHAPTER 1. GENERAL COMPUTER TERMS

Abstract

This chapter defines a number of general computing terms and abbreviations.

DEFINITIONS

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.

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.

Java Management eXtensions, JMX

A Java technology that supplies tools for managing and monitoring applications, system objects, devices, and service-oriented networks.

Java Database Connectivity, JDBC

An API specified in Java technology that provides Java applications with access to databases and other data sources.

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 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.

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.

marshalling

The process of taking in-memory objects and converting them to a binary or textual format for transmission over a transport.

See also: unmarshalling

OASIS

An international consortium that drives the development, convergence, and adoption of Web services standards. See http://www.oasis-open.org.

Spring framework

A comprehensive programming and configuration model for modern Java-based enterprise applications the uses dependency injection.

See also: dependency injection

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

CHAPTER 2. COMMON MESSAGING TERMS

DEFINITIONS

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.

client

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.

consumer

An application that consumes messages from a messaging destination.

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.

destination

A logical holding area for messages in a message broker. Clients publish messages to and consume messages from destinations.

See also: queue, topic

durable subscriber

A message consumer that receives all messages published on a topic, including those published while the subscriber is inactive.

message

An atomic unit of data that is passed between two or more clients. A message consists of three components:

- 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

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.

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.

master/slave

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.

producer

An application that creates messages and posts them to a messaging destination.

point-to-point messaging

A messaging style where messages are sent between two known endpoints. This messaging style is typically implemented using queues.

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.

queue

A destination that uses first in/first out semantics.

See also: destination

request-reply pattern

A messaging pattern in which a message producer receives a message and returns a correlated message.

Session

A JMS object that provides a single-threaded context for producing and consuming messages. JMS clients use the `Session` object to create producers, consumers, messages, and other artifacts used to work with messages.

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.

transport

A standards-based network protocol, such as HTTP or STOMP, that defines how objects communicate over a network.

topic

A destination that uses publish and subscribe semantics.

See also: destination

CHAPTER 3. RED HAT JBOSS A-MQ MESSAGING TERMS

DEFINITIONS

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.

Apache ActiveMQ

An open source project that provides the messaging technology for Red Hat JBoss A-MQ.

See http://activemq.apache.org

cluster

- A group of brokers among which clients can failover.

  See also: failover.

- A collection of clustered services.

  See also: clustered service

connection

A bridge between a client and a broker connector or between two brokers in a network of brokers.

See also: transport connector, network connector, network of brokers

connector

An object that connects clients to a broker.

composite destination

A virtual destination that serves as a proxy for multiple destinations. Producers can send messages to the composite destination an it will be automatically sent to all of the physical destinations that make up the composite destination.

See also: virtual destination

dead letter queue

A special destination used by the message broker to hold undeliverable messages.

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

discovery agent

A mechanism that advertises the list of available message brokers to message clients and other message brokers.

See also: dynamic discovery

exclusive consumer

A mechanism that ensures that only one consumer connected to a queue can consume messages.

failover

- A transport that automatically moves to a new connection in the event that its current connection fails.
- A cluster architecture where clients are able to migrate from a failed broker to a running broker.

network of brokers

A group of brokers that are linked together to operate as a single logical unit.

network connector

A configuration entity used to link brokers together to form a network of brokers.

See also: network of brokers

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

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.

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.

transport connector

An address at which a message broker accepts client connections.

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

CHAPTER 4. OSGI TERMS

DEFINITIONS

OSGi

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 dynamically resolve dependencies between bundles and can handle having multiple versions of a bundle deployed simultaneously.

The OSGi specifications are maintained by the OSGi Alliance. See http://www.osgi.org.

Apache Karaf

An open source project that provides the OSGi runtime container used by Red Hat JBoss A-MQ.

See http://karaf.apache.org

bundle

The primary deployment format used in Red Hat JBoss A-MQ. 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

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

child container

A container that is created by a container on the same host. Child containers are run on the same host as their parent container, but each child runs in a separate JVM.

When created using the console's **`admin:create-container`**, a child container inherits the features, feature repositories, and configuration from its parent. When a child container is created using the **`fabric:container-create`** command, the **`fabric:contaier-create-child`**, command, or the management console, it does not inherit any configuration from its parent.

Regardless of how they are created, child containers can be started and shutdown from their parent container's console without using SSH.

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.

persistent identifier, PID

A registration property used by the OSGi Configuration Admin Service to identify a group of configuration attributes.

CHAPTER 5. FUSE FABRIC TERMS

DEFINITIONS

agent

See Fabric Agent

clustered service

A service that can be discovered via Fuse Fabric and has master/slave support.

ensemble

See Fabric Ensemble

Fuse Application Bundle, FAB

A bundle that uses a POM file to specify its dependencies.

Fuse Fabric

An open source project that provides a distributed runtime registry that provides configuration, deployment, and discovery services to a collection of distributed containers.

See also: fabric

Fabric Agent

The service running inside a Fabric Container that is responsible for configuring and provisioning the container according to the profiles assigned to the container . It is also responsible for updating the registry with runtime information about the services in container.

fabric

A group of containers that are connected to a common Fabric Ensemble. The ensemble makes it possible for all of the containers to share runtime information about the services deployed in each container and allows them to share common configuration profiles.

Fabric Container

An Apache Karaf-based container that is managed by a Fabric Agent.

See also: Fabric Agent

Fabric Registry

A ZooKeeper-based distributed registry that stores runtime and configuration information about the services in a fabric.

Fabric Server

A server that, as part of a Fabric Ensemble, provides a number of services that bind a fabric. These services include the Fabric Registry, dynamic load balancing, and location transparency.

See also: Fabric Registry

Fabric Ensemble

A group of one or more Fabric Servers that provide a number of services that bind a fabric. These

services include the Fabric Registry, dynamic load balancing, and location transparency.

See also: Fabric Registry, Fabric Server

managed container

See Fabric Container

non-managed container

An Apache Karaf-based container that is registered with a fabric, but is *not* managed by a Fabric Agent.

profile

A set of data that defines runtime artifacts and configuration settings for provisioning a Fabric Container.

registry

See Fabric Registry

standalone container

A container that is not part of a fabric and does not have a Fabric Agent installed.

standalone broker

See standalone container

version

A collection of configuration profiles in a Fabric Registry.

See also: profile