

# **Apache ServiceMix @version@ Installation Guide**

**Guillaume Nodet <gnodet@apache.org>**  
**Gert Vanthienen <gertv@apache.org>**  
**Lars Heinemann <lhein@apache.org>**  
**Jean-Baptiste Onofre <jbonofre@apache.org>**  
**Bruce Snyder <bsnyder@apache.org>**

# Apache ServiceMix @version@ Installation Guide

by Guillaume Nodet, Gert Vanthienen, Lars Heinemann, Jean-Baptiste Onofre, and Bruce Snyder  
Copyright © 2009 Apache Software Foundation



## Table of Contents

1. Introduction .....	1
2. Installation Requirements .....	2
Hardware .....	2
Operating Systems .....	2
Environment .....	2
Get Apache ServiceMix @version@ .....	2
3. Standalone Binary Distribution Installation Procedure .....	3
Uncompress .....	3
Starting ServiceMix .....	3
Stopping ServiceMix .....	3
4. ServiceMix Into A JEE Application Server Installation Procedure .....	4
5. ServiceMix Configuration .....	5
6. Deploying components and services into ServiceMix .....	7
7. Tuning ServiceMix .....	8

## List of Examples

3.1. Windows .....	3
3.2. Linux/Unix .....	3
3.3. Windows .....	3
3.4. Linux/Unix .....	3



# Chapter 1. Introduction

Apache ServiceMix is an open source ESB (Enterprise Service Bus) that combines the functionality of a Service Oriented Architecture (SOA) and an Event Driven Architecture (EDA) to create an agile, enterprise ESB.

Apache ServiceMix is a distributed ESB built from the ground up on the Java Business Integration (JBI) specification JSR 208 and released under the Apache license. The goal of JBI is to allow components and services to be integrated in a vendor independent way, allowing users and vendors to plug and play.

ServiceMix is lightweight and easily embeddable, has integrated Spring support and can be run at the edge of the network (inside a client or server), as a standalone ESB provider or as a service within another ESB. You can use ServiceMix in Java SE or Java EE application server.

ServiceMix uses ActiveMQ to provide remoting, clustering, reliability and distributed failover.

ServiceMix is completely integrated into Apache Geronimo, which allows you to deploy JBI components and services directly into Geronimo. ServiceMix is being JBI certified as part of the Geronimo project.

ServiceMix can be embedded into a JEE application server such as JBoss, Oracle Weblogic or IBM Websphere.

ServiceMix includes a complete JBI container supporting all parts of the JBI specification including:

- NMR (Normalized Message Router)
- JBI Management MBeans
- full support for the JBI deployment units with hot-deployment of JBI components.

ServiceMix also provides a simple to use client API for working with JBI components and services.

ServiceMix includes many JBI components including HTTP, JMX, CXF, BPEL, etc.

# Chapter 2. Installation Requirements

## Hardware

- 100 MB of disk space for the ServiceMix @version@ binary distribution.

## Operating Systems

- *Microsoft Windows*: XP SP2, 2000, 2003, Vista, 7
- *Unix*: any Unix/Linux platform that supports Java.

## Environment

- Java Developer Kit (JDK) 1.5.x (Java 5) for both deployment and compiling.

## Get Apache ServiceMix @version@

You can download Apache ServiceMix @version@ here: <http://servicemix.apache.org/download.html>.

Apache ServiceMix is provided in two formats:

- tarball for Unix platform
- zip for Windows platform

# Chapter 3. Standalone Binary Distribution Installation Procedure

## Uncompress

First you need to uncompress the downloaded archive (tarball or zip) in the directory of your choice (where you have write permissions). Define the `SERVICEMIX_HOME` variable:

### Example 3.1. Windows

```
> set SERVICEMIX_HOME=C:\apache-servicemix-@version@
```

### Example 3.2. Linux/Unix

```
$ export SERVICEMIX_HOME=/apache-servicemix-@version@
```

## Starting ServiceMix

### Example 3.3. Windows

```
> cd %SERVICEMIX_HOME%  
> .\bin\servicemix
```

### Example 3.4. Linux/Unix

```
$ cd $SERVICEMIX_HOME  
$ ./bin/servicemix
```

## Stopping ServiceMix

For both Windows and Unix installations, terminate ServiceMix by typing "CTRL-C" on the command line where ServiceMix is running. ServiceMix uses the Java shutdown hook to cleanly shut down the container.

# **Chapter 4. ServiceMix Into A JEE Application Server Installation Procedure**



# Chapter 5. ServiceMix Configuration

This section contains description of main ServiceMix directories and files :

- *activemq-data* - this directory is created by the ServiceMix at startup time. It contains the messages and transactions repository of the embedded Apache ActiveMQ.
- *bin/servicemix* - contains the main startup script of Apache ServiceMix.
- *conf/activemq.xml* - contains the embedded Apache ActiveMQ configuration. This file shouldn't be modified as it's managed by internal ServiceMix processes.
- *conf/components.properties* - contains the tuning of embedded JBI components. See the tuning section for more informations.
- *conf/servicemix.conf* - contains the ServiceMix launcher classloading definition. This file shouldn't be modified.
- *conf/servicemix.properties* - it's the main ServiceMix configuration file. It defines the port numbers used by the ServiceMix. You can change this file if one of the port number is already bound on your system. By default, ServiceMix uses the following ports :
  - 1099 (rmi.port) - the core RMI port used by the ServiceMix.
  - 61616 (activemq.port) - the embedded Apache ActiveMQ port.
- *conf/servicemix.xml* - contains the main ServiceMix Spring application context. This file mustn't be modified.
- *conf/groups.properties* - contains ServiceMix JAAS security group definition. You can define group security to access to ServiceMix via JMX.
- *conf/jmx.xml* - contains ServiceMix JMX layer application context. This file mustn't be modified.
- *conf/jndi.properties* - contains the default ServiceMix JNDI definition. This file shouldn't be modified as it's managed by ServiceMix internal processes.
- *conf/keystore.jks* - contains the security keys and certificates used by ServiceMix security layer. This file mustn't be modified.
- *conf/log4j.xml* - contains ServiceMix logging definition. You can change this file to change the default log level of ServiceMix components. You can add new loggers if required.
- *conf/login.properties* - contains ServiceMix JAAS security definition. This file mustn't be changed.
- *conf/security.xml* - contains ServiceMix Security layer application context. This file mustn't be modified.
- *conf/tx.xml* - contains ServiceMix Transaction layer application context. This file mustn't be modified.
- *conf/users-credentials.properties* - contains ServiceMix JAAS security users and credentials (in case of external security module). You can define users allowed to administrate ServiceMix via JMX.
- *conf/users-passwords.properties* - contains basic ServiceMix JAAS security users and passwords (internal). You can add users allowed to administrate ServiceMix via JMX.
- *data/log/servicemix.log* - this file is created by ServiceMix at startup time. It contains main ServiceMix log file.

- *data* - this directory is created by ServiceMix at startup time. It contains all required file (storage) required by ServiceMix internal components.
- *extras* - this directory contains extras JAR files required by JBI components deployed in ServiceMix. If you want to add JAR files in ServiceMix classloader, use **lib** directory.
- *hotdeploy* - this directory contains JBI components (services) automatically deployed by ServiceMix.
- *lib* - this directory contains all required JAR files for ServiceMix. You can add new JAR file in this directory if required.

# **Chapter 6. Deploying components and services into ServiceMix**

DRAFT

# Chapter 7. Tuning ServiceMix

You can tune ServiceMix at three basic levels :

- *Underlying JVM* - ServiceMix uses an underlying JVM to run. By default, the JVM configuration is set to 512MB of maximum memory usage (that is enough for most of usage). Anyway if you encounter OutOfMemory problem and bad performances, you can define JVM parameters. To do it, create a **conf/servicemix.rc** file. In this file, you can define `JAVA_MIN_MEM` and `JAVA_MAX_MEM` variables.
- *Core Engine Tuning* - in the **conf/servicemix.properties** file, you can see three properties :
  - *smx.corePoolSize* - define the startup number of ServiceMix threads. By default, 16 threads are started.
  - *smx.maximumPoolSize* - define the maximum number of ServiceMix threads. If you see some bottlenecks around thread waits, you can increase this value. By default, 32 threads maximum are allowed.
  - *smx.queueSize* - define the maximum of messages store in the embedded Apache ActiveMQ. If you see some bottlenecks around message waits, you can increase this value. By default, ServiceMix can store 256 messages.
- *Components Tuning* - You can tune ServiceMix embedded JBI components using the **conf/components.properties**. By default, this file contains the basic tuning for the HTTP components.