

How to cook your Spagic

Author

Gianfranco Boccalon
Andrea Zoppello

1	Document Goal.....	3
1.1	Document Conventions	3
2	Versions History	3
3	Configuration Steps	4
3.1	Additional libraries	4
3.2	Spagic Service Assemblies	6
3.3	Configuration without LAN.....	6
4	Spagic components	6
4.1	Prerequisites	6
4.2	Lightweight components.....	6
4.3	Standard Binding Components.....	7
5	ServiceMix patched components	7
5.1	Lightweight components.....	7
5.2	Standard Service Engines.....	7
5.3	Standard Binding Components.....	8
6	BPEL components	8
7	Monitoring components	8
7.1	Prerequisites	8
7.2	Components	8
7.3	Configuration	9
7.4	Datasources	9
8	Related documents.....	10

1 Document Goal

The goal of this document is to provide you the information necessary to transform a clean ServiceMix installation in a Spagic environment.

1.1 Document Conventions

In this document we will refer to the version of ServiceMix as `{SMX_VERSION}`, to the version of the Spagic platform as `{SPAGIC_VERSION}`, and to the version of Apache ODE as `{ODE_VERSION}`.

Current Spagic release is 2.0.0.

Current ServiceMix release is 3.1.2.

Current Apache ODE release is 1.0.0.

2 Versions History

Version/Release n° :	1.0	Date	24/07/2007
Description	First release (English version)		
Version/Release n° :	2.0	Date	15/10/2007
Description	Updates for ServiceMix 3.1.2		
Version/Release n° :	3.0	Date	21/01/2008
Description	Updates for Spagic 2.0.0		

3 Configuration Steps

3.1 Edit the file *conf\servicemix.conf*

You have to add some folders to the *servicemix.conf* file.

The file should contain the following lines (exactly in this order):

```
load ${servicemix.home}/conf
load ${servicemix.home}/lib/hib/*.jar
load ${servicemix.home}/lib/optional/*.jar
load ${servicemix.home}/lib/talend/*.jar
load ${servicemix.home}/lib/*.jar
load ${servicemix.home}/resources/xsd
load ${servicemix.home}/resources/keystores
```

3.2 Edit the file *conf\servicemix.xml*

In *servicemix.xml* file you have to configure the listeners namespaces. Add the following definitions at the beginning of the file (in the *beans* tag, the root element):

```
xmlns:syncListeners="java://org.spagic.smx.listeners.sync"
xmlns:monitoringListeners="java://org.spagic.monitoring.smx.listeners"
```

and then add the listener for Synchronizer error management (for the details see *Spagic Studio Components.doc*) in the *sm:container* tag:

```
<sm:listeners>
  <!-- Spagic: listener for error management with Synchronizer -->
  <syncListeners:SynchronizerFaultListener/>
</sm:listeners>
```

3.3 Edit the file *conf\component.properties*

Edit the file *conf\component.properties* and add the following line:

```
servicemix-http.retryCount=0
```

This allows avoiding the retry mechanism enabled by default on HTTP Binding Component.

3.4 Additional folders

Create the folders:

- `${servicemix.home}/resources/xsd`
- `${servicemix.home}/resources/keystores`

3.5 Additional libraries

These libraries are used by the Spagic components and by the patched ServiceMix components.

You should retrieve them from the Spagic distribution and copy in your ServiceMix.

The libraries are:

- `{SERVICEMIX_HOME}\lib`
 - commons-codec-1.3.jar (Used for encoding and encoding in base64)
 - commons-dbcp-1.2.1.jar (Dependency of commons-dbutils)
 - commons-dbutils-1.1.jar (New code for simple jdbc component)
 - commons-lang-2.2.jar (Dependency of commons-dbutils)
 - groovy-all-1.0-JSR-06.jar (Groovy Lightweight se)
 - mysql-connector-java-5.0.7-bin.jar (My sql connector driver)
 - mysql-connector-java-5.0.7-license.txt (MySQL driver license)
 - ojdbc14.jar (Oracle JDBC driver)
 - ojdbc14_license.txt (Oracle driver license)
 - script.jar (needed by Groovy Lightweight component)
 - spring-support.jar (needed by mail lw service engines)

- `{SERVICEMIX_HOME}\lib\hib`: Hibernate libraries and dependencies
 - Hibernate and dependencies
 - ant-antlr-1.6.5.jar
 - antlr-2.7.6.jar
 - asm-attrs.jar
 - asm.jar
 - c3p0-0.9.1.jar
 - cglib-2.1.3.jar
 - commons-logging-1.0.4.jar
 - dom4j-1.6.1.jar
 - ehcache-1.2.3.jar
 - hibernate3.jar
 - jaxen-1.1-beta-7.jar
 - jta.jar
 - Spagic Persistence Layer library
 - metadb-model-{SPAGIC-VERSION}.jar

- `{SERVICEMIX_HOME}\lib\talend`: libraries for Talend integration (these libraries are usually included within Talend jobs)
 - ant.jar
 - axis.jar
 - commons-discovery-0.2.jar
 - db2jcc.jar
 - db2jcc_license_cu.jar
 - edtfpj-1.5.4.jar
 - file_delimited.jar
 - javacsv.jar
 - jaxrpc.jar
 - jconn3.jar
 - jtds-1.2.jar
 - jxl.jar
 - ldapjdk.jar
 - load_product.jar
 - postgresql-8.1-405.jdbc3.jar
 - postgresql-8.1-405.jdbc3-license.txt
 - saaj.jar
 - systemRoutines.jar
 - userRoutines.jar

- `{SERVICEMIX_HOME}\lib\optional`: libraries used by the lightweight components.
 - commons-collections-3.1.jar
 - commons-pool-1.2.jar
 - jaxen-1.1-beta-11.jar
 - spring-jms-2.0.1.jar

3.6 Spagic Service Assemblies

The HTTP BC (*servicemix-http-{SMX_VERSION}-installer.zip*) should be installed before installing Spagic service assemblies.

There are some Spagic Service Assemblies to deploy, to complete the Spagic installation.

The service assemblies are:

- **getResources-sa{SPAGIC_VERSION}.zip**: service assembly used by Spagic Studio to retrieve the ServiceMix datasource configuration.
- **restart-sa{SPAGIC_VERSION}.zip**: service assembly used by the Spagic Console to restart the failed processes.
- **bckdyndata-sa{SPAGIC_VERSION}.zip**: service assembly used by the Spagic Console to backup historic data from the metadatabase, to the backup metadatabase.
- **deldyndat-sa{SPAGIC_VERSION}.zip**: service assembly used by the Spagic Console to delete dynamical data from the metadatabase.

3.7 Configuration without LAN

If you want to disable ActiveMQ multicast because you are working not connected to a LAN, change the *activemq.xml* file removing the *discoveryUri* attribute from the `<amq:transportConnector>` tag:

```
<amq:transportConnector uri="tcp://localhost:61616" discoveryUri="multicast://default"/>
```

becomes

```
<amq:transportConnector uri="tcp://localhost:61616"/>
```

And remove also the tag `<amq:networkConnector>`.

4 Spagic components

We describe the Spagic components included in the Spagic platform. You should include these components in your ServiceMix environment to extend its base features with the Spagic features.

4.1 Prerequisites

Before installing Spagic components you have to install the component ServiceMix Lightweight Container (*servicemix-lwcontainer-{SMX_VERSION}-installer.zip*) and to add the library *servicemix-components-{SMX_VERSION}.jar* in the `{SERVICEMIX_HOME}\lib`.

4.2 Lightweight components

These components are installed in the `{SERVICEMIX_HOME}\lib\optional` folder.

You should retrieve them from the Spagic distribution and copy in your ServiceMix.

The components are:

- **spagic-advanced-jdbc-{SPAGIC_VERSION}.jar**: Advanced JDBC SE. For performing any type of operation on database.
- **spagic-manage-{SPAGIC_VERSION}.jar**: Management SE. Used by Spagic Studio to retrieve information about a running ServiceMix.

- **spagic-restart-{SPAGIC_VERSION}.jar**: Restart SE. Used by Spagic Studio to manage process restart.
- **spagic-synchronizer-{SPAGIC_VERSION}.jar**: Synchronizer SE. Allow to build in-out processes using In-Only components.
- **spagic-talend-{SPAGIC_VERSION}.jar**: Talend SE. Allow to call Talend processes.
- **spagic-attachment-{SPAGIC_VERSION}.jar**: attachment management SE. Allow to process attachments in several ways.

4.3 Standard Binding Components

These components are installed in the `{SERVICEMIX_HOME}\install` folder.

You should retrieve them from the Spagic distribution and copy in your ServiceMix.

The components are:

- **spagic-tcp-{SPAGIC_VERSION}-installer.zip**: TCP binding component for reading and writing through TCP sockets.
- **spagic-jdbc-{SPAGIC_VERSION}-installer.zip**: JDBC poller component. Allow polling on a database table, sending the new inserted data through normalized messages.

5 ServiceMix patched components

In this chapter we describe the patched ServiceMix components: the patches are necessary for fixing unresolved bugs, or for extending the components with features necessary to Spagic.

5.1 Lightweight components

These components are released in the `{SERVICEMIX_HOME}\lib\optional` folder, in the archive **spagic-smxpatches-{SPAGIC_VERSION}.jar**.

The patched/extended components are:

- **Groovy**: Patched bug [SM-781] <https://issues.apache.org/activemq/browse/SM-781>
- **JDBC**: The original one was not working well, changed implementation introduced commons-dbutils usage.
- **ValidateComponent**: Added support to handle fault, not with JBI fault but with a specific fault message.
- **XsltComponent**: Patched transform content method that has some problems when message content is of type StaxSource.

5.2 Standard Service Engines

These components are released in the `{SERVICEMIX_HOME}\install` folder.

You should retrieve them from the Spagic distribution and copy in your ServiceMix.

The components are:

- **servicemix-drools-{SMX_VERSION}-installer.zip**: introduced namespace configuration, DbHelper class in drools working memory to improve rule expression language, and introduced defaultTargetService.
Patched bug [SM-888] <https://issues.apache.org/activemq/browse/SM-888>
Patched bug [SM-924] <https://issues.apache.org/activemq/browse/SM-924>
- **servicemix-eip-{SMX_VERSION}-installer.zip**.
Patched bug [SM-879] <https://issues.apache.org/activemq/browse/SM-879>
EIP Pipeline original code does not copy message properties.
Added more capabilities to SplitAggregator (envelope name, create a unique envelope).
Added the RecipientListAggregator class that implements the recipient-list-aggregator component.

5.3 Standard Binding Components

These components are released in the `{SERVICEMIX_HOME}\install` folder.

You should retrieve them from the Spagic distribution and copy in your ServiceMix.

The components are:

- **servicemix-file-{SMX_VERSION}-installer.zip**: the component supports the new attribute “*append*”, used when it’s used as output binding component, to decide if the file has to be created from scratch, or if it has to be used in append mode.

6 BPEL components

Spagic supports execution and monitoring of BPEL processes through usage of the BPEL Service Engine Apache ODE.

After downloading the Apache ODE JBI SE, and **before installing**, you have to modify it to support monitoring.

Open the `ode.jbi.properties` file within the component and modify the value of the property `ode-jbi.event.listeners`:

```
ode-jbi.event.listeners=org.spagic.monitoring.ode.listeners.OdeListener
```

Add to the `lib` folder of the component the following libraries retrieved from the Spagic distribution:

- `spagic-monitor-{SPAGIC_VERSION}.jar`
- `spagic-ode-listener-{SPAGIC_VERSION}.jar`

Add to the `META-INF\jbi.xml` file the following libraries:

```
<path-element>lib/spagic-monitor-{SPAGIC_VERSION}.jar</path-element>
<path-element>lib/spagic-ode-listener-{SPAGIC_VERSION}.jar</path-element>
```

After saving the zip file, it can be deployed in the `deploy` folder.

7 Monitoring components

This section explains how to install the Spagic monitoring components on ServiceMix.

7.1 Prerequisites

The JMS BC (`servicemix-jms-{SMX_VERSION}-installer.zip`) and the ServiceMix Shared Libraries (`servicemix-shared-{SMX_VERSION}-installer.zip`) should be installed before installing Spagic components.

7.2 Components

The components necessary for monitoring the Spagic platform are:

- **Monitoring listener**: this listener monitor all the exchanges handled by ServiceMix and copy them to a queue for further processing.
The listener code is in the file “**spagic-listener-{SPAGIC_VERSION}.jar**” stored in the `lib/optional` folder.
- **Monitoring process**: this is a Service Assembly composed of a single JMS BC, which listens for the queue populated by the Monitoring listener.
The JMS BC is the standard JMS component with a custom message processor.
The process is stored in the file “**spagic-monitorService-sa{SPAGIC_VERSION}.zip**” and should be copied in the ServiceMix “`deploy`” folder.

7.3 Configuration

To configure the monitoring you have to:

- Configure the queue for the listener. This is done by the file "**activemq.xml**":

```
<!-- Spagic: queue for SMX monitor service -->
<amq:queue id="smxMonitorQueue" physicalName="org.spagic.monitor.SMXQueue"/>
```

This configuration should be inserted after the tag `<amq:broker>` (**after, not within**).

- Configure the JMS connection factory. This is done in the "**jndi.xml**" file:

```
<!-- Spagic jms connection factory -->
<entry key="java:comp/env/jms/SpagicJmsConnectionFactory">
  <amq:connectionFactory brokerURL="${activemq.url}" />
</entry>
```

- Configure the listener. This is done in the "**servicemix.xml**" file and it is composed of several steps:
 - Configure the JMS factory for the listener (before the `sm:container` tag):

```
<!-- Spagic: JMS factory for monitor listener -->
<bean id="spagicJmsFactory" class="org.springframework.jndi.JndiObjectFactoryBean">
  <property name="jndiName" value="java:comp/env/jms/SpagicJmsConnectionFactory"/>
</bean>
```

- Configure the listener itself (in the `sm:container` tag):

```
<sm:monitoringListeners>
  <!-- Spagic: listener for monitor service -->
  <listeners:AuditingExchangeASyncListener>
    <property name="connectionFactory" ref="spagicJmsFactory"/>
    <property name="destinationQueue" ref="smxMonitorQueue"/>
  </listeners:AuditingExchangeASyncListener>
  <!-- Other listeners -->
  .....
</sm:listeners>
```

- In ServiceMix you should have also the **hibernate.cfg.xml** in the "`conf`" folder, and the `metadb-model{SPAGIG-VERSION}.jar` in the "`lib/hib`" folder.

7.4 Datasources

Spagic needs two datasources:

- MetaDB Datasources (Mandatory):** Is the datasource that configure the connection pool with the metadatabase where Spagic store all monitoring informations.

It must be configured in `${servicemix.home}/conf/jndi.xml` with the key `java:comp/env/jdbc/metadb`

- **Backup MetaDB (Optional):** Is the datasource that configure the connection pool with the backup metadb where spagic move historic data using the backup service.
It must be configured in `${servicemix.home}/conf/jndi.xml` with the key `java:comp/env/jdbc/metadb-bck`

Here an example of the jndi.xml fragment defining the datasources:

```
<entry key="java:comp/env/jdbc/metadb">
  <bean id="metadb-ds"
    class="org.apache.commons.dbcp.BasicDataSource"
    destroy-method="close">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
    <property name="url" value="jdbc:mysql://localhost:3306/spagic"/>
    <property name="username" value="spagic"/>
    <property name="password" value="spagic"/>
  </bean>
</entry>

<entry key="java:comp/env/jdbc/metadb-bck">
  <bean id="metadb-ds"
    class="org.apache.commons.dbcp.BasicDataSource"
    destroy-method="close">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
    <property name="url" value="jdbc:mysql://localhost:3306/spagic-bck"/>
    <property name="username" value="spagic"/>
    <property name="password" value="spagic"/>
  </bean>
</entry>
```

8 Related documents

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

1. *Spagic Studio Components.doc*: detail document about *Spagic Studio* environment.
2. *Spagic Console.doc*: detail document about *Spagic Console* monitoring application.