

Spagic Service Manager Concepts And Configuration Guide

Author Andrea Zoppello

1	Document Goal	3
2	Versions History	3
3	Introduction	4
3.1	Understanding the Pieces.....	4
4	Spagic Service Manager in depth	5
4.1	Starting the service manager and the startup phase	5
4.2	Using the Service Manager with another SPAGIC INSTANCE	9
4.3	Instance INI File Detailed Parameters.....	9
4.4	The Spagic.xml file and the Distributed Mode	10
4.4.1	Simplest spagic.xml (local mode distribution not enabled)	11
4.4.2	Two master nodes on the same cluster	11
4.4.3	Master Slave Configuration (not yet fully supported)	11

1 Document Goal

The goal is to explain the basic concepts and all configuration details, about the spagic service manager.

2 Versions History

Version/Release n° :	1.0	Date	04/05/2010
Description	First release (English version)		

3 Introduction

At a first glance Spagic could seem to be a Complex Platform to use. This is mainly because:

- **The Overall Solution** is composed by a **set of pieces working together** instead of distributing a whole package.
- **A wide range of capabilities and configuration options is offered.**

The goal of this document is to introduce users to the Spagic architecture, and concepts and then to examine the configuration detail offered.

3.1 Understanding the Pieces

As you can see from the picture Spagic is made by the following parts:

- **Spagic Service Manager**

The service manager is the runtime part of spagic, basically it's the software that acts as server for Spagic Processes/Services and Connector

- **Spagic Studio Tools**

It's the set of "Eclipse Based" development tools for developing/creating/configuring services/processes/connectors that will be deployed on service manager

- **Spagic Monitoring and Web Tools**

In this category we find the Spagic Monitoring Console that is the tool to monitor your system during runtime phase, and SpagicTaskList web application that serves as a simple Task management application

4 Spagic Service Manager in depth

When you get a Spagic distribution you could find the service manager in **SPAGIC_DISTRIBUTION_HOME/service-manager**.

One important thing you must care about is **not to make confusion** between **service-manager** and **spagic-instances**.

- The **service manager** is composed by the equinox OSGi container, with the spagic runtime core services and components. The service manager is only one and is composed of the following parts:
 - Spagic3 binary executable (Spagic3.exe or Spagic3.sh)
 - The equinox osgi container configuration file (/configuration folder)
 - The OSGi bundles of the server (/plugins folder)
- A **spagic-instance** instead is a particular configuration for the service manager. In particular you could have more than one spagic-instances, and choose which one to execute within a service manager. A Spagic instance is composed by the following part:
 - INI File (Instance ini.file)
 - Service/Connector descriptor
 - Datasource references
 - JMS connection factories descriptor
 - Processes

4.1 Starting the service manager and the startup phase

To start the service manager you must run the following command:

```
spagic3 [-debug] -console
```

To understand what happens here and the following paragraph in this document we must know what happens during the startup phase.

When Equinox is started using the native launchers it will look for a **ini file**. If we **don't specify with a specific argument which ini file to use the Equinox launcher look for a file called as the Launcher executable with the ini extension**. This mean that with the default launch command above, the launcher looks for a file called **Spagic3.ini**

A typical ini file looks like:

```
--launcher.XXMaxPermSize  
256M  
-vmargs
```

```
-Dosgi.requiredJavaVersion=1.6
-Declipse.ignoreApp=true
-Dosgi.noShutdown=true
-Dspagic.home=C:/Scrappy
-Dea.maxThreadPoolSize=1000
-Xms40m
-Xmx512m
```

Despite what other parameters mean, the most important parameter is the **spagic.home** parameter because this is the parameter that sets the SPAGIC_INSTANCE_HOME we want to use when running the service manager.

An important thing to say is that the **spagic.home** parameter is not mandatory and if it's not set, Spagic service manager uses as the spagic.home the folder **SERVICE_MANAGER_HOME/default**.

Before to see in the following paragraph how to use the same service manager with another instance it's useful to see in detail what happens during the startup phase

1. If **spagic.home** (as specified in the ini or **SERVICE_MANAGER_HOME/default**) does not exist, the service manager will create a complete new instance structure in the spagic.home. Otherwise if the instance directory is already there spagic will not overwrite anything there.
2. After the configuration (and eventually creation) of the instance directory the startup phase will configure logging services using the file located in **spagic.home/logback.xml**. As for the creation of the home folder if the logback.xml is already there it will be used, otherwise spagic will create one and use this.
3. After that the startup tries to locate a file called **spagic.xml** (as for the other ones this file is created with default configuration by spagic if it's not already there). This file is the one that contains node identification and clustering information in the case we're using spagic in a distributed environment where more than one node is present.
4. As the last steps spagic provides to create some services in the **instance.home** that are used by the service manager itself and it proceeds to configure the **two mandatory datasources** with the **references to the database that a service instance needs to use**.

During the startup phase two datasources are defined in instance.home/datasources as follows:

```
Metadb.ds ( metadatabase location )
<?xml version="1.0" encoding="UTF-8"?>
<spagic:ds
  xmlns:spagic="urn:org:spagic3"
  xmlns="urn:org:spagic3"
  id="metadb">
```

```

<!-- MySQL
<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"/>
-->

<!-- H2 SERVER -->
<property name="driverClassName" value="org.h2.Driver"/>
<property name="url" value="jdbc:h2:tcp://localhost/~ /spagic"/>
<property name="username" value="spagic"/>
<property name="password" value="spagic"/>

<!-- POSTGRES SQL
<property name="driverClassName" value="org.postgresql.Driver"/>
<property name="url" value=" jdbc:postgresql://localhost/spagic"/>
<property name="username" value="spagic"/>
<property name="password" value="spagic"/>
-->
</spagic:ds>

```

Jbpm.ds (metadabase location)

```

<?xml version="1.0" encoding="UTF-8"?>
<spagic:ds
  xmlns:spagic="urn:org:spagic3"
  xmlns="urn:org:spagic3"
  id="jbpm">

  <!-- My SQL
  <property name="driverClassName" value="com.mysql.jdbc.Driver"/>
  <property name="url" value="jdbc:mysql://localhost:3306/jbpm"/>
  <property name="username" value="jbpm"/>
  <property name="password" value="jbpm"/>
  -->

  <!-- H2 SERVER -->
  <property name="driverClassName" value="org.h2.Driver"/>
  <property name="url" value="jdbc:h2:tcp://localhost/~ /jbpm"/>
  <property name="username" value="jbpm"/>
  <property name="password" value="jbpm"/>

```

```
<!-- POSTGRES SQL
<property name="driverClassName" value="org.postgresql.Driver"/>
<property name="url" value="jdbc:postgresql://localhost/jbpm"/>
<property name="username" value="jbpm"/>
<property name="password" value="jbpm"/>

-->
</spagic:ds>
```

As you could see from the above file the default datasources are referring to a local h2 database installation. The important thing is that if you need to change the datasource of metadb or of the jbpm you could change that files but the two datasources must have metadb, and jbpm as the ids.

5. Once the above steps are completed some daemon threads are launched by the service manager, in particular we've :
 - a. The daemons thread that are polling the services/and connector folders for .service and .connector file
 - b. The daemon thread that is responsible for managing datasource (**.ds file located in spagic.home/datasources**) configuration.
 - c. The daemon thread that is responsible for managing jms connection factories (**.cf file located in spagic.home/connectionfactories**) configuration.

4.2 Notes on Demo Mode and the H2 Database

When you run the Spagic Service manager *without specifying the "spagic.mode" parameters* or setting it as **"spagic.mode=demo"** the servicemanager will use a run H2 as the default database for persistence.

The H2 database is very easy to use and to embed for that reason is useful for the spagic.mode. On the other side **it has some drawbacks when used in production scenario, where for example it miss the "row-level locking" feature.**

For that reason when you're running spagic service manager in demo mode is suggested to keep the number of thread pools to 1 as follow.

So in the instance ini file use the following option when using the demo mode:

ea.maxThreadPoolSize = 1

4.3 Using the Service Manager with another SPAGIC INSTANCE

In the previous paragraph we've understood the difference between the service manager and a spagic instances concept and we've examined in details the spagic startup phase.

To resume we've understood that basically the service manager when starts take the configuration of the instance to run from:

1. The ini file where we have some parameters (in particular the spagic.home)
2. Directly from the instance home directory: spagic.xml, logback.xml, datasources, and jms connection factories.

Now the last thing we need to see is how to run the service manager to point to another instance. At a first glance the solution seems to be to edit the Spagic3.ini file and change the spagic.home parameter, but **THIS IS NOT THE RIGHT WAY TO DO THAT.**

The reason for which this is the wrong solution is that the ini file is part of the configuration of the instance itself (because in the ini file there's not only the spagic.home but also other parameters of the instance).

So **THE RIGHT WAY TO CONFIGURE A NEW INSTANCE** is to:

1. Create a new ini file (give the name you prefer for example Alternate.ini)
2. Edit the new file and specify the parameters of the new instance
3. Launch Spagic using the “**-launcher.ini**” parameters to specify this new ini file as follow:

```
spagic3 --launcher.ini <PATH_TO_INI_FILE> [-debug] -console
```

4.4 Instance INI File Detailed Parameters

Parameter Name	Description	Syntax	Default Value
spagic.mode	Let us to switch from demo/production mode. In the demo mode when the service manager is started it provides also to start the database and to create the tables, in production mode the database is already expected to be created.	-Dspagic.mode=demo Or -Dspagic.mode=production	demo
spagic.home	The path to the instance directory	-Dspagic.home=C:\Scrappy	SERVICE_MANAGER_HOME/default
ea.maxThreadPoolSize	The size of event admin thread pool size	-Dea.maxThreadPoolSize=1000	100

spagic.attachment.store	Attachment Store Type: fs = FileSystem db = Database attachment datasource is the one described in attachment.ds Or -Dspagic.attachment.store=db	-Dspagic.attachment.store=fs fs
fs.root	If attachment store is fs determine the fs root	-Dfs.root=C:\tmp INSTANCE_HOME/store/attachments
spagic.exchange.store	Exchange Store Type: osgi.pref = Use OSGi preference store Or db = database -Dspagic.exchange.store=db	-Dspagic.exchange.store=osgi.pref osgi.pref
jbpn.clobAsText	Set to true when using database where clob does not exist and you need to use text	-Djbpn.clobAsText=true false
org.hibernate.dialect	The hibernate dialect to detect. Spagic try to autodetermine dialect but in some situations it could be useful to specify it	 Autodetermined

4.5 The Distributed Mode and Spagic.xml

When spagic is used in cluster more than one instances in the same network are running, for this reason each spagic instance running need to be identified with a unique identifier.

The spagic.xml file contains different types of informations but basically it's needed for:

1. Identifying the node (nodeid attribute)
2. Identify if this node is a master or slave node (node type)
3. Determine if the node use the distributed broker (distributed)

When spagic is used within a distributed configuration, the service manager use a particular **"distributed broker"** implementation that relies on a JMS. So for the distributed mode we need to install and set up a JMS Broker/Server, before to start the service manager.

Although the "distributed broker implementation" relies on JMS at the moment Spagic only support the **ActiveMQ (5.3.X)** the open source jms server by apache.

To start the ActiveMQ Broker:

- Get a distribution from: from <http://activemq.apache.org/activemq-531-release.html>
- Unzip in your favourite folder (ACTIVEMQ_HOME) and start the server with:

ACTIVEMQ_HOME/bin/activemq

In the following paragraph we'll show some typical configurations for the distributed mode:

4.5.1 Simplest spagic.xml (local mode distribution not enabled)

Node1/spagic.xml
<pre><?xml version="1.0" encoding="UTF-8"? <spagic nodeid="node1" nodetype="master" distributed="false"> <cluster clusterid="SpagicCluster" masterslave="false" referredmaster="mymaster" distributedtopic="tcp://ahost:61616/exampleTopic" /> </spagic></pre>

4.5.2 Two master nodes on the same cluster

Node1/spagic.xml	Node2/spagic.xml
<pre><?xml version="1.0" encoding="UTF-8"? <spagic nodeid="node1" nodetype="master" distributed="true"> <cluster clusterid="SpagicCluster" masterslave="false" referredmaster="mymaster" distributedtopic="tcp://ahost:61616/exampleTopic" /> </spagic></pre>	<pre><?xml version="1.0" encoding="UTF-8"? <spagic nodeid="node2" nodetype="master" distributed="true"> <cluster clusterid="SpagicCluster" masterslave="false" referredmaster="mymaster" distributedtopic="tcp://ahost:61616/exampleTopic" /> </spagic></pre>

4.5.3 Master Slave Configuration (not yet fully supported)

Node1/spagic.xml	SlaveNode1/spagic.xml
------------------	-----------------------

<pre> <?xml version="1.0" encoding="UTF-8"? <spagic nodeid="node1" nodetype="master" distributed="true"> <cluster clusterid="SpagicCluster" masterslave="false" referredmaster="mymaster" distributedtopic="tcp://ahost:61616/exampleTopic"/> </spagic> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"? <spagic nodeid="slaveNode1" nodetype="slave" distributed="true"> <cluster clusterid="SpagicCluster" masterslave="true" referredmaster="node1" distributedtopic="tcp://ahost:61616/exampleTopic"/> </spagic> </pre>
---	---