

Spagic Getting Started

Author

Daniela Butano

1	Document Goal.....	3
2	Versions History	3
3	Introduction.....	4
4	Requirements	5
5	Installation	6
5.1	Persistence layer – MetaDB	6
5.2	Service Manager	6
5.3	Spagic Studio	6
5.4	Spagic Console	7
6	Developing a Process.....	8
6.1	Create a Spagic project.....	8
6.2	Design the process.....	8
6.3	Publish the process.....	9
6.4	Deploy the process.....	9
7	Run the Process	10
8	Monitor the Process.....	12
8.1	Authentication.....	12
8.2	Processes List.....	13
8.3	Process Graph	14
8.4	Processes Instances List.....	15
8.5	Process Instance Graph.....	16
8.6	Advanced Search	17
9	Related documents.....	17

1 Document Goal

The goal of this document is to provide you with an introduction on using Spagic platform by designing, deploying and monitoring a sample process after the Spagic installation and configuration.

2 Versions History

Version/Release n°:	1.0	Date	29/06/2007
Description	First release (English version)		
Version/Release n°:	1.1	Date	03/08/2007
Description	Added the steps for the creation of the backup database.		

3 Introduction

Spagic is a solution composed by a set of visual tools and back-end applications oriented towards design, realization, deploy and monitoring of ESB infrastructures adherent to the SOA paradigm:

- **Studio:** Eclipse environment that consents the use of a single interface in order to create components and processes managing the entire development cycle: design, service registry configuration (UDDI & ebxml), metadata management, WSDL generation, rules definition, mapping, data integration ([Talend Open Studio](#) generated job), custom services and orchestration.
- **Enterprise Monitoring:** for the monitoring, through Ajax interface, of the system information (system monitoring), services & processes (services monitoring) and business indicators (BAM). The monitor tool, realized on SpagoBI, can be easily extended through the realization of report and dashboard towards its own business intelligence indicators.
- **Service Manager:** using the open source ESB [ServiceMix](#) project and the Spagic listener it is possible to manage correlated processes and information during the services orchestration. New binding components have been implemented in synchronous modality also (See *Spagic Studio Components.doc*).
- **Persistence layer - MetaDB:** the different services and the rules of process and extraction of the relevant information are described and classified through a metadata system. Therefore the repository manages the entire life cycle of the components and traces their use in order to optimize the monitoring activities. The registration is made through Spagic studio and at runtime on Service Manager by means the specific listener.

The document was organized in the following way: there is a first section related to the Spagic complete installation (in order: MetaDB, ServiceMix, Spagic Studio and Spagic Console), then there are the sections related to deploy, run and monitor of a very simple process.

For more details about Spagic Studio environment see the document *Spagic Studio Components.doc*.

For more details about Spagic Console, see the document *Spagic Console.doc*.

4 Requirements

Required Tools	URL for download/Notes	Spagic Studio	Spagic Console
Database	MySQL Oracle	X	X
Eclipse Web Tools	http://download.eclipse.org/webtools/downloads/	X	
GraphViz	http://www.graphviz.org/	X	
jUDDI	http://ws.apache.org/juddi/	Optional	Optional
JDK 1.5.0_11 or later	http://java.sun.com/	X	X
Apache Tomcat 5.5.17	http://tomcat.apache.org		X
Mozilla Firefox 2.0.0.x	http://www.mozilla.com		X

Note: X means Required.

5 Installation

The following steps allow you to install the entire Spagic platform.

5.1 Persistence layer – MetaDB

To setup the metadatabase follow these steps:

1. Install MySQL Server (release 5.0 or higher) or Oracle (release 9i or 10g).



Important Note: if you choose to use Oracle, you have to include manually the driver “*jdbc14.jar*” in Spagic Studio and ServiceMix, manually.

For Spagic Studio the driver has to be copied in “*apache-servicemix-3.1-incubating\lib\optional*”.

For ServiceMix the driver has to be copied in

“*SPAGIC_STUDIO_HOME\plugins\com.bull.cimero.pluginEditor_2.8.0.jar\lib*”.

2. Create a new schema “*smx*” and a user “*smx*” with password “*smx*” and the permissions for writing into the schema.
3. Generate the tables using the “*spagic-metadb-<database type>.ddl*” released in the *spagic-metadb* package.
4. Launch the scripts “*setup-<database type>.sql*” released in the *spagic-metadb* package, to load all the configuration tables.
5. **Optional step:** install the backup database for monitoring data, if you want to clean periodically the monitoring data. Create a new schema “*smx-bck*” and assign to the user “*smx*” the permissions for writing into the schema.
6. Generate the tables using the “*spagic-bck-<database type>.ddl*” released in the *spagic-metadb* package.

5.2 Service Manager

To install the Service Manager follow these steps:

1. To install ServiceMix 3.1, copy the folder “*apache-servicemix-3.1-incubating*”, released in the *spagic-service-manager* package, into your Spagic installation folder (you should create this new folder). All base components are already installed (component http, tcp, jms....) into “*apache-servicemix-3.1-incubating\install*”. The monitoring system, composed by a listener, monitoring all the exchanges handled by ServiceMix, and a process, listening for the queue populated by the listener, is already installed and configured.
2. **If you installed the database on a different machine from the ServiceMix machine** then configure the “*apache-servicemix-3.1-incubating\conf\hibernate.cfg.xml*” file so it could link to the database “*smx*” just created.
3. To launch ServiceMix, launch the command “*bin\servicemix*” from the *apache-servicemix-3.1-incubating* folder.

5.3 Spagic Studio

To install Spagic Studio on client machine follow these steps:

1. Get graphviz installation package from <http://www.graphviz.org/> and install it. If you're using windows, simply run graphviz executable file and follow the wizards. During the installation steps take in mind the location where the executable dot program was installed.

In Windows (Italian language) default installation graphviz will install the dot program in C:\Programmi\ATT\Graphviz\bin\dot.exe

2. Spagic Studio is distributed as an eclipse plugin. Get it from Spagic distribution (from the package *spagic-studio*) and install it on a clean eclipse with WebTools complete distribution. This can be found here: <http://download.eclipse.org/webtools/downloads/>.

We tested the plugin with Eclipse 3.2 and WTP 1.5.

In this document we will refer to the Eclipse folder as *SPAGIC_STUDIO_HOME*.

3. Spagic Studio needs to connect to Spagic metadatabase. Ensure the Meta database is running and that permissions are configured properly on database server.
4. An installation of tomcat where we install JUDDI if we want to publish our services in UDDI Registries. To install JUDDI refer to its installation documentation.

To start you only need to launch eclipse.exe in *SPAGIC_STUDIO_HOME*.

5.4 Spagic Console

To install Spagic Console follow the next steps:

1. Create into a database a new schema "*spagobi*" for SpagoBI environment, create a new user "*spagobi*" with password "*bispage*" and associate him the schema privileges for writing and reading. Execute the "*spagobi-<database type>.ddl*", released with Spagic in the package *spagic-console/DDL*, to create the tables, and execute the script *loadSpagobi-<database type>.sql* to load the configuration data.
2. Install the Apache Tomcat 5.5.17; to install it refer to its installation documentation (<http://tomcat.apache.org>)
3. Install the web application SpagicConsole, you should copy into *apache-tomcat-5.5.17\webapps* folder the *Spagic.war* released with Spagic.
4. **If you used a configuration different from the default suggested in this document**, before starting Tomcat it's necessary to verify the following Spagic Console **configuration files** and update them:

- *\SpagicConsole\WEB-INF\conf\cms.xml*: set the location of the SpagoBI CMS repository in the *value* attribute of the "*repository_path*" parameter, if you installed the folder externally to the Web application. For example, if you installed *cmsspagobi* folder into "C:\temp\cmsspagobi" set the value attribute as *value="/temp/cmsspagobi"*;
- *\SpagicConsole\WEB-INF\conf\data_access.xml*: set the values of parameters *connectionString*, *user*, *userPassword* with the proper values to connect to the persistent layer metadb (default schema is "*smx*" and default user is "*smx*");
- *\SpagicConsole\WEB-INF\conf\spagobi\spagobi.xml*: in the parameter *<HIBERNATE-CFGFILE>* set the location of Hibernate configuration file used by *SpagoBI*. If you installed MySQL database, the file used is *\SpagicConsole\WEB-INF\classes\hibernate.cfg.mysql.xml*.
Update the Hibernate configuration file with the proper values to connect to SpagoBI schema (default schema is "*spagobi*").
- *\SpagicConsole\WEB-INF\conf\jmx\server.xml*: set the *jmxUrl* to the URL for connecting to ServiceMix by JMX. The URL is written by ServiceMix on its console, on the startup.
- Update the Hibernate configuration file *\SpagicConsole\WEB-INF\classes\hibernate.cfg.xml* so it could link to Spagic database (default schema and user is "*smx*");
- If you used UDDI, update the file *\SpagicConsole\WEB-INF\classes\Serviceregistry.properties* to set the URL of jUDDI application and all the parameters for the Spagic database connection.

6 Developing a Process

Spagic Studio is an Eclipse environment that allows the use of a single interface in order to create components and processes managing the entire development cycle. For the details, see the document *Spagic Studio Components.doc*.

In this section we'll shortly describe all steps for developing the processes using a simple example.

6.1 Create a Spagic project

Open Spagic Studio and create a workspace for working, for example call it *SpagicDemo*.

As first step, configure the Spagic Studio Preferences from *Window->Preferences->Spagic Preference*, set the attributes for the MetaDB, the jUDDI application and the graphviz application.

To create a new project, select *File->New ->Other->Spagic->Spagic Project*, insert a Project Name, *StartSpagic*, click *Finish*.

In the workspace, the project just created has a standard structure containing the followings directories:

- **Integration Process:** contains all Spagic file describing the service assembly in terms of endpoints and flow between them;
- **Mappings:** contains resources used by the mapping component. Almost of this resources will be XSLT file.
- **Scripts:** contains resources that are used by Scripting Components. At the moment groovy is the language for the scripting so this folder will contain groovy file;
- **SemanticRules:** contains resources used by Semantic Validator Component;
- **SyntaxRules:** contains resources that are used by the Syntax Validator Component;
- **WsdIFiles:** contains resources automatically generated by Spagic Studio if the process contains entry endpoint relative to HTTP Component configured to be a SOAP Provider.

6.2 Design the process

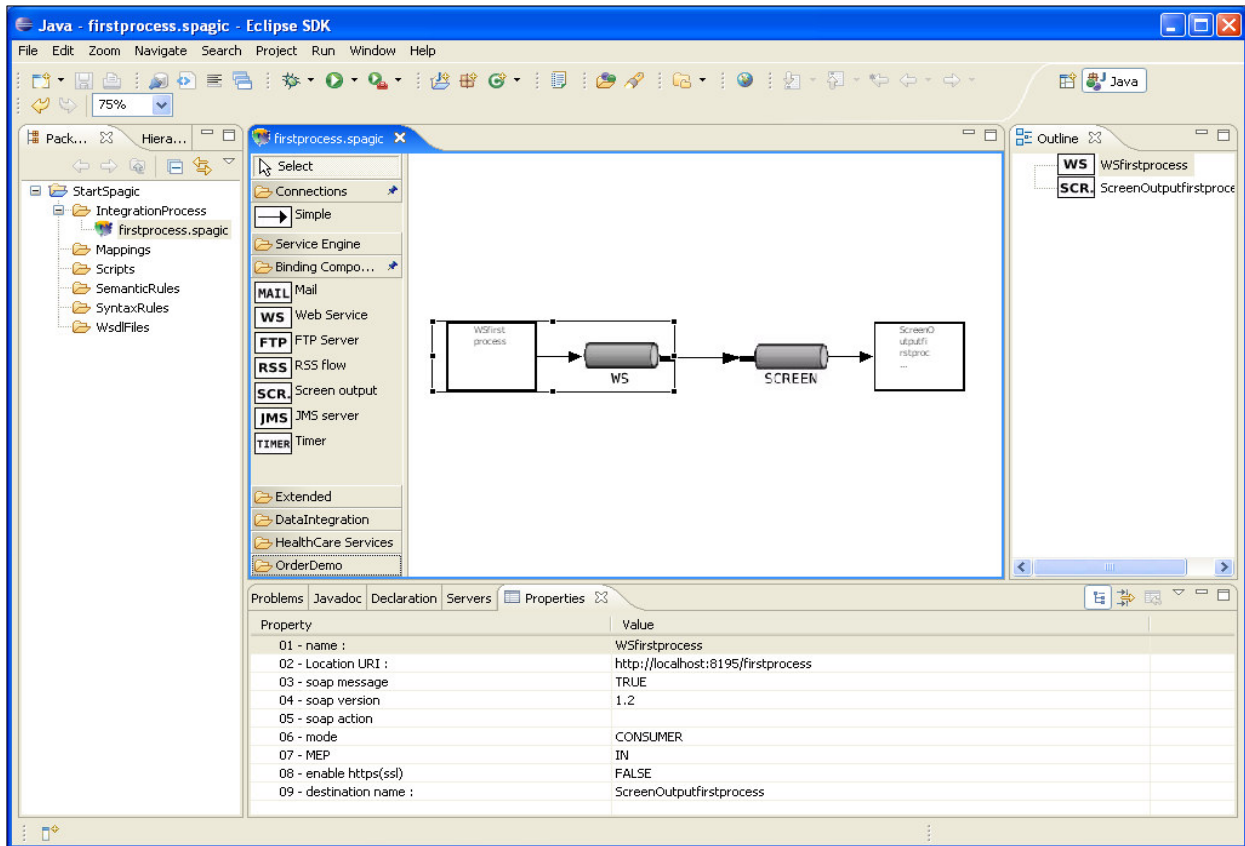
To define a new process, select the *Integration Process* folder, open the context menu and *New->Other->Spagic->Spagic File*. Insert a name with *.Spagic* extension, for example *firstprocess.Spagic*.

When the file has been created in the *Integration Process* folder, the visual Spagic editor will be opened.

To design a very simple process, select, from the component palette located at the left of the editor, the binding component *Web Service* and drag it to editing area. Click the endpoint just created in the editor area to configure its properties. Set the name with *WSfirstprocess* and the location URI with, for example, <http://0.0.0.0:8195/firstprocess>.

Select from the component palette located at the left of the editor, the binding component *Screen Output* and drag it to editing area. Set the name with *ScreenOutputfirstprocess*.

Create a simple connection, between the two components, selecting from component palette the arrow into *Connections->Simple* and then selecting in editor area the two components to link.



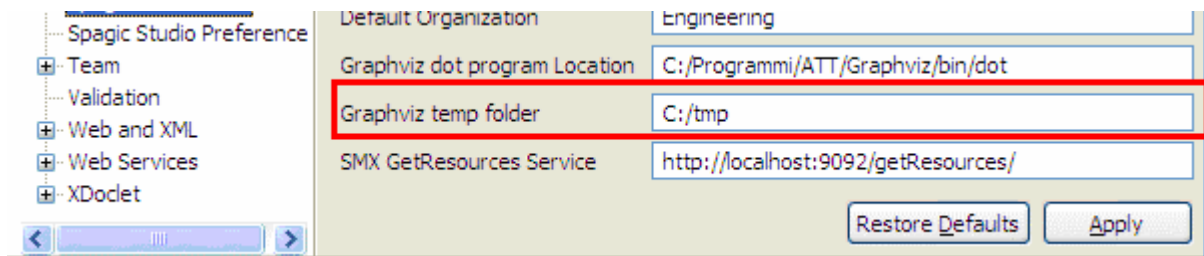
Save the process just defined.

6.3 Publish the process

After designing the process, you have to publish it: this step will save the process information (properties, flow...) into the MetaDB.

In the editor area, open the context menu and select *Publish in Database*.

If you have some errors, verify that you installed *graphviz* tool and you created the temporary directory that should be configured in the Spagic Preferences, in the item *Graphviz temp folder*.



6.4 Deploy the process

To deploy the process, in the editor area, open the context menu and select *Generate JBI package*; the *firstprocessJar.zip* will be created in the *Integration Process* folder.

Start ServiceMix launching the command `bin\servicemix.bat` (or `.sh`) from `apache-servicemix-3.1-incubating`.

After ServiceMix was started, copy the `.zip` generated in the `deploy` folder and verify in the ServiceMix console that the process was deployed without errors.

```

ServiceMix
INFO - ComponentMBeanImpl - Starting component: transformerForSupplier
INFO - JBIContainer - Activating component for: [container=ServiceMix,name=supplierScreenOutput] with
service: <http://servicemix.org/cheese>supplierScreenOutput component: org.apache.servicemix.components.util.StreamWri
erComponent@f79885
INFO - ComponentMBeanImpl - Initializing component: supplierScreenOutput
INFO - ComponentMBeanImpl - Starting component: supplierScreenOutput
INFO - AutoDeploymentService - Directory: deploy: Finished installation of archive: storeJar.zip
INFO - AutoDeploymentService - Directory: deploy: Archive changed: processing transformerJar.zip ...
INFO - ServiceAssemblyLifeCycle - Starting service assembly: transformer
INFO - ServiceUnitLifeCycle - Initializing service unit: transformerhttp-su
INFO - ServiceUnitLifeCycle - Initializing service unit: transformerlw-su
INFO - ServiceUnitLifeCycle - Starting service unit: transformerhttp-su
INFO - jetty - jetty-6.0.1
INFO - jetty - Started SelectChannelConnector @ localhost:8194
INFO - ServiceUnitLifeCycle - Starting service unit: transformerlw-su
INFO - JBIContainer - Activating component for: [container=ServiceMix,name=myTransformertransformer]
with service: <http://servicemix.org/cheese>myTransformertransformer component: org.apache.servicemix.components.xslt.X
ltComponent@2d465a
INFO - ComponentMBeanImpl - Initializing component: myTransformertransformer
INFO - ComponentMBeanImpl - Starting component: myTransformertransformer
INFO - JBIContainer - Activating component for: [container=ServiceMix,name=myScreenOutputtransformer]
with service: <http://servicemix.org/cheese>myScreenOutputtransformer component: org.apache.servicemix.components.util
StreamWriterComponent@128bff9
INFO - ComponentMBeanImpl - Initializing component: myScreenOutputtransformer
INFO - ComponentMBeanImpl - Starting component: myScreenOutputtransformer
INFO - AutoDeploymentService - Directory: deploy: Finished installation of archive: transformerJar.zip
INFO - AutoDeploymentService - Directory: deploy: Archive changed: processing firstprocessJar.zip ...
INFO - ServiceAssemblyLifeCycle - Starting service assembly: firstprocess
INFO - ServiceUnitLifeCycle - Initializing service unit: firstprocesshttp-su
INFO - ServiceUnitLifeCycle - Initializing service unit: firstprocesslw-su
INFO - ServiceUnitLifeCycle - Starting service unit: firstprocesshttp-su
INFO - jetty - jetty-6.0.1
INFO - jetty - Started SelectChannelConnector @ localhost:8195
INFO - ServiceUnitLifeCycle - Starting service unit: firstprocesslw-su
INFO - JBIContainer - Activating component for: [container=ServiceMix,name=ScreenOutputfirstprocess]
with service: <http://servicemix.org/cheese>ScreenOutputfirstprocess component: org.apache.servicemix.components.util.S
reamWriterComponent@e762ec
INFO - ComponentMBeanImpl - Initializing component: ScreenOutputfirstprocess
INFO - ComponentMBeanImpl - Starting component: ScreenOutputfirstprocess
INFO - AutoDeploymentService - Directory: deploy: Finished installation of archive: firstprocessJar.zip

```

7 Run the Process

To run the process you can use a simple HTML page that invokes, with an AJAX call, the Web Service defined in the `firstprocess` process.

You can find this page in the distribution, in the package `spagic-service-manager`, within the folder `sample-client`.

```

<HTML>
<head>
<title>ServiceMix SOAP Binding Example</title>
<script type="text/javascript">
function getHTTPObject() {
    var xmlhttp = false;
    if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
        try {
            xmlhttp = new XMLHttpRequest();
        } catch (e) {
            xmlhttp = false;
        }
    }
    if (xmlhttp) {
        xmlhttp.onreadystatechange=function() {
            if (xmlhttp.readyState == 4) { /* 4 : état "complete" */
                response.value = "STATUS: " + xmlhttp.statusText + "\n" + xmlhttp.responseText;
            }
        }
    }
    return xmlhttp;
}

```

```

}
function sendSoapIO() {
    var xmlhttp = getHTTPObject();
    if (!xmlhttp) {
        alert('could not create XMLHttpRequest object');
        return;
    }
    var request = document.getElementById("request");
    var response = document.getElementById("response");
    try {
        netscape.security.PrivilegeManager.enablePrivilege("UniversalBrowserRead UniversalBrowserWrite");
        xmlhttp.open("POST", "http://localhost:8195/firstprocess/", true);
    } catch (e) {
        alert('error opening');
    }
    xmlhttp.send(request.value);
}
</script>
</head>
<body>
<h1>ServiceMix SOAP Binding Example</h1>
<table>
    <tr>
        <td>
            <textarea id="request" style="width:600px;height:400px"><?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Body>
        <FIRSTPROCESS/>
    </env:Body>
</env:Envelope>

            </textarea>
        </td>
        <td>
            <textarea id="response" style="width:600px;height:400px">
            </textarea>
        </td>
    </tr>
    <tr>
        <td colspan=2>
            <input type="button" value="Send SOAP" onClick="sendSoapIO();" />
        </td>
    </tr>
</table>
</body>
</HTML>

```

Open the HTML page with Mozilla and click the button *sendSOAP* to start a new execution of the process.

8 Monitor the Process

Spagic Console is a web application to monitor, through Ajax interface, the system information (system monitoring), the services, the processes (services monitoring) and the business indicators (BAM).

It's recommended to use the Firefox Mozilla 2.0.0.x


For more details about Spagic Console please read the document *Spagic Console.doc*.

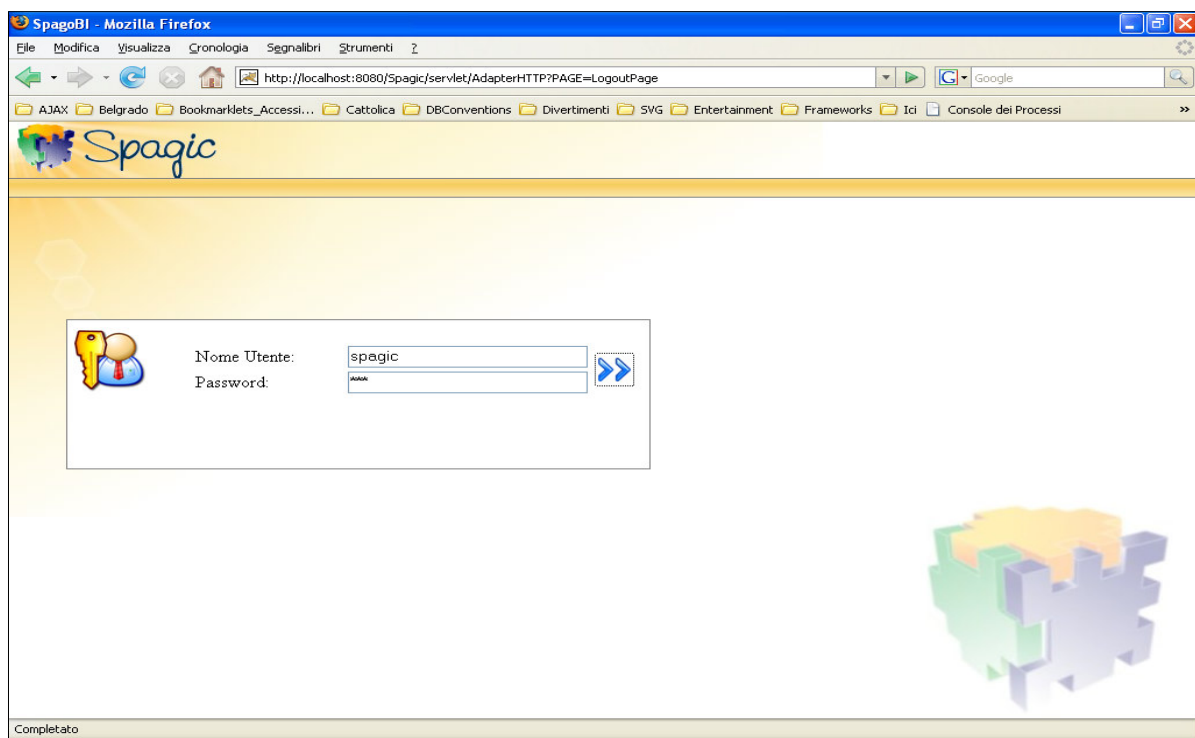
Start the Tomcat where you installed the Spagic Console.

8.1 Authentication


The URL to launch the Spagic Console is: <http://localhost:8080/Spagic/>.

The first page visualized by the web application is the authentication page.

Insert the user *Spagic* and the password *Spagic* and then click the button .



The next page contains the application menu.

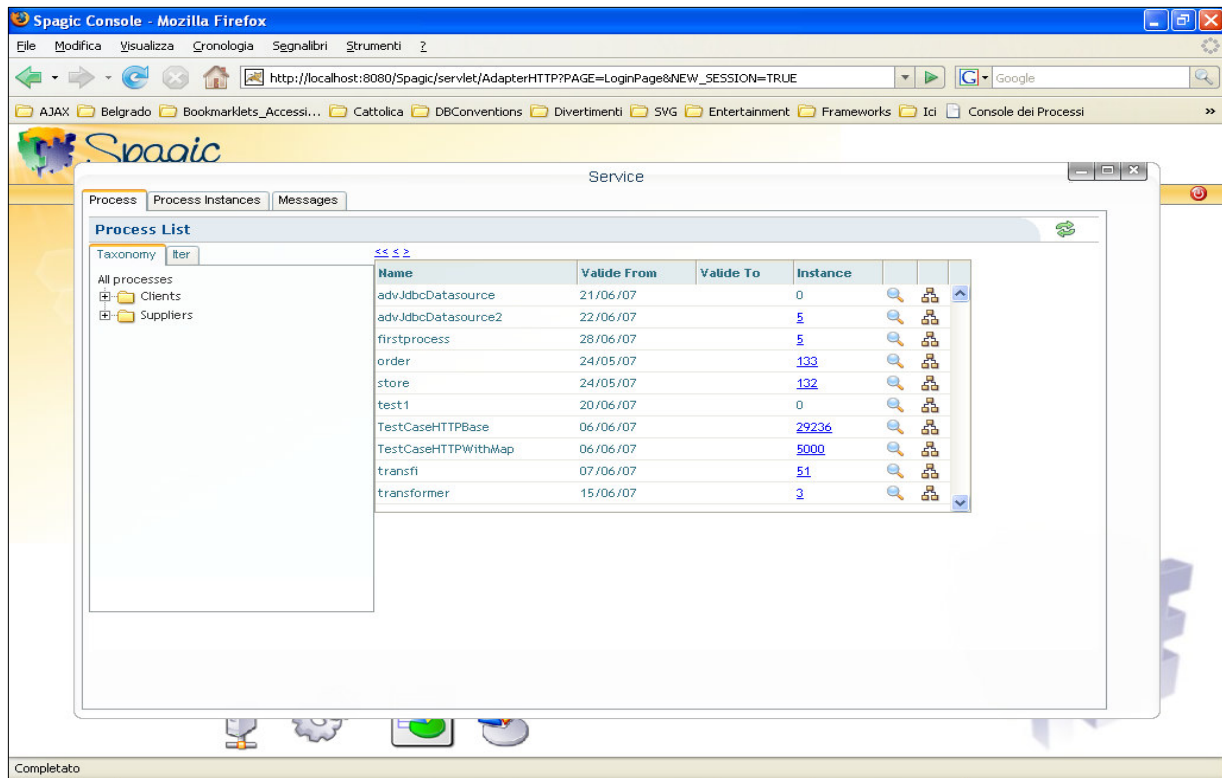
Select the icon **Service** () to view all processes (and their definitions, properties, static flow, endpoint), all instances processes (and their dynamics flows, messages, relevant data) and to do an advanced search.

8.2 Processes List

Selecting the icon *Service* and tab *Processes* the processes list is visualized. The list contains the following attributes:


- *Name*
- *Valide From*
- *Valide To*
- *Instance* contains number of instances associate to a process;
- Icon to display the detail containing the *description*, *version*, *organisation UDDI*, *name UDDI*;
- Icon to display the process graph;

The process list contains *firstprocess* process just deployed and launched.

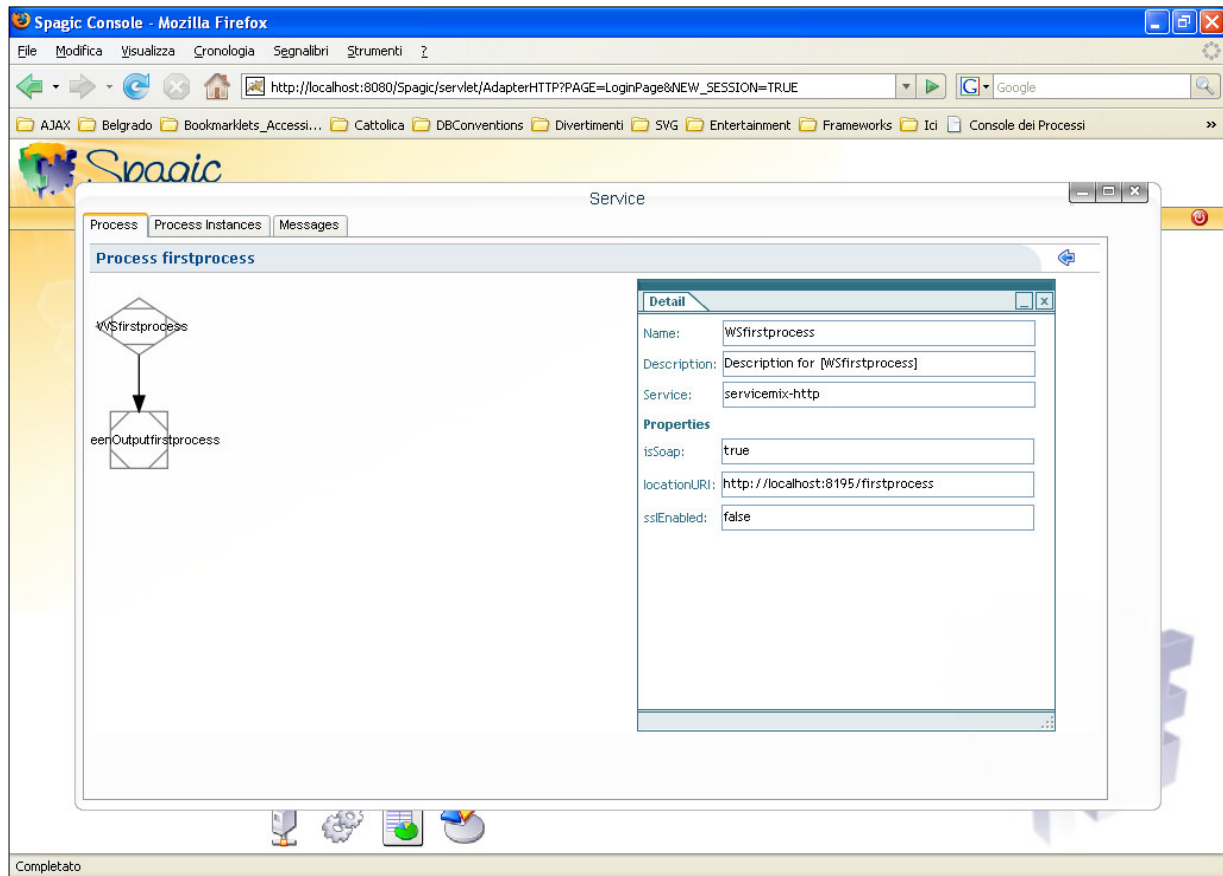


It's possible to make a search by *taxonomy* (if you choose to install jUDDI) or by *Iter* using the section in the left side of the page.

8.3 Process Graph

Selecting the icon  from the process list, it's possible to display the process graph.

Selecting the elements on the graph it's possible to display a window containing *Name*, endpoint *Description*, *Service* and its *Properties*.




The screenshot shows the Spagic Console interface in Mozilla Firefox. The browser address bar displays `http://localhost:8080/Spagic/servlet/AdapterHTTP?PAGE=LoginPage&NEW_SESSION=TRUE`. The console has tabs for 'Process', 'Process Instances', and 'Messages'. The 'Process' tab is active, showing a process graph for 'firstprocess'. The graph consists of a start node 'WSfirstprocess' connected to an output node 'eenOutputfirstprocess'. A 'Detail' window is open, displaying the following information:

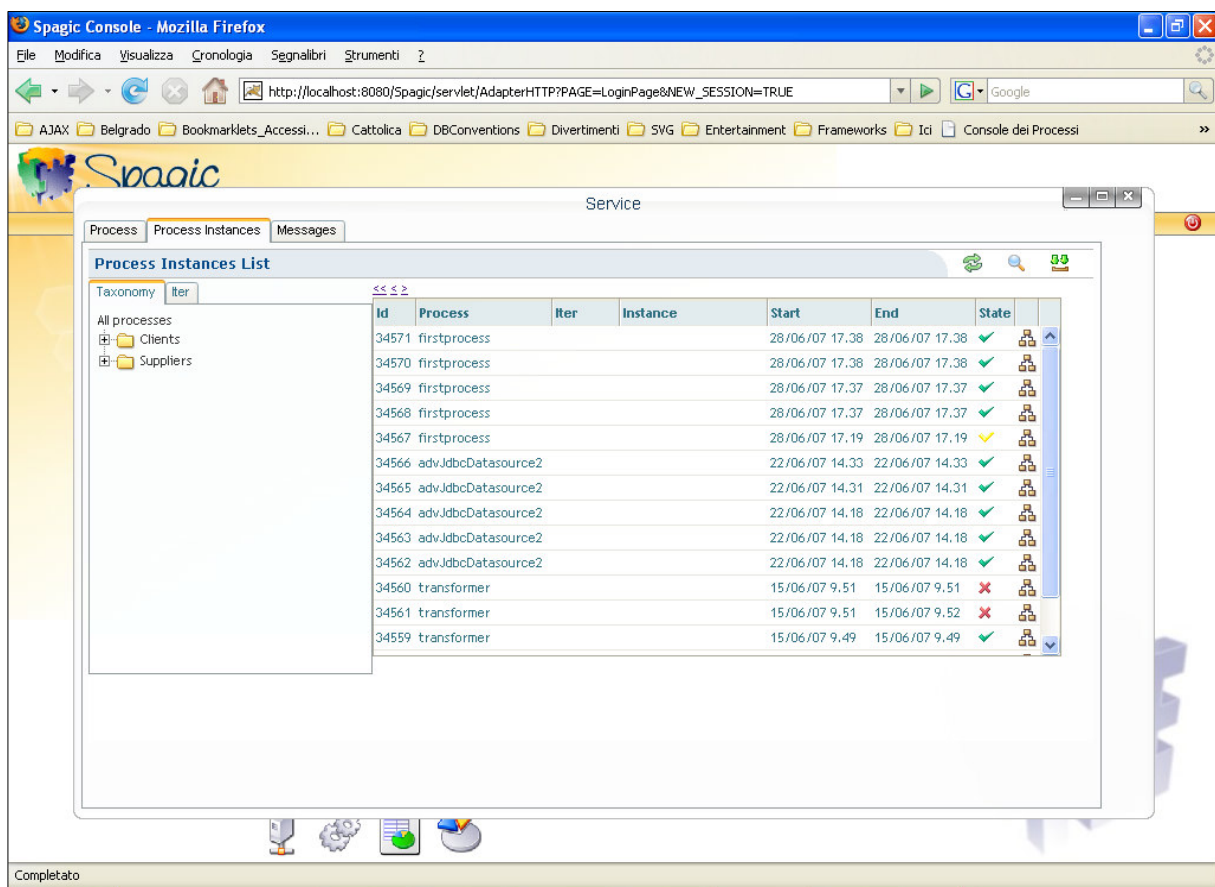
Detail	
Name:	WSfirstprocess
Description:	Description for [WSfirstprocess]
Service:	servicemix-http
Properties	
isSoap:	true
locationURI:	http://localhost:8195/firstprocess
sslEnabled:	false














The status bar at the bottom of the console shows 'Completato'.

8.4 Processes Instances List

Selecting the icon *Service* and tab *Process Instances* the process instances list is showed. The list contains:


- *Id* of instance process.
- *Name*.
- *Iter* associate to the instance.
- *Instance* contains the attributes identifying the iter.
- *Start* execution time.
- *End* execution time.
- *State* shows the instance state that can be *active* (yellow ✓), *executed* (green ✓) or *fault* (red ✕).
- Icon  to visualize the process execution detail.



Id	Process	Iter	Instance	Start	End	State	Icon
34571	firstprocess			28/06/07 17.38	28/06/07 17.38	✓	
34570	firstprocess			28/06/07 17.38	28/06/07 17.38	✓	
34569	firstprocess			28/06/07 17.37	28/06/07 17.37	✓	
34568	firstprocess			28/06/07 17.37	28/06/07 17.37	✓	
34567	firstprocess			28/06/07 17.19	28/06/07 17.19	✓	
34566	advJdbcDatasource2			22/06/07 14.33	22/06/07 14.33	✓	
34565	advJdbcDatasource2			22/06/07 14.31	22/06/07 14.31	✓	
34564	advJdbcDatasource2			22/06/07 14.18	22/06/07 14.18	✓	
34563	advJdbcDatasource2			22/06/07 14.18	22/06/07 14.18	✓	
34562	advJdbcDatasource2			22/06/07 14.18	22/06/07 14.18	✓	
34560	transformer			15/06/07 9.51	15/06/07 9.51	✕	
34561	transformer			15/06/07 9.51	15/06/07 9.52	✕	
34559	transformer			15/06/07 9.49	15/06/07 9.49	✓	

It's possible to make a search by *taxonomy* or by *iter* using the section in the left side of the page.

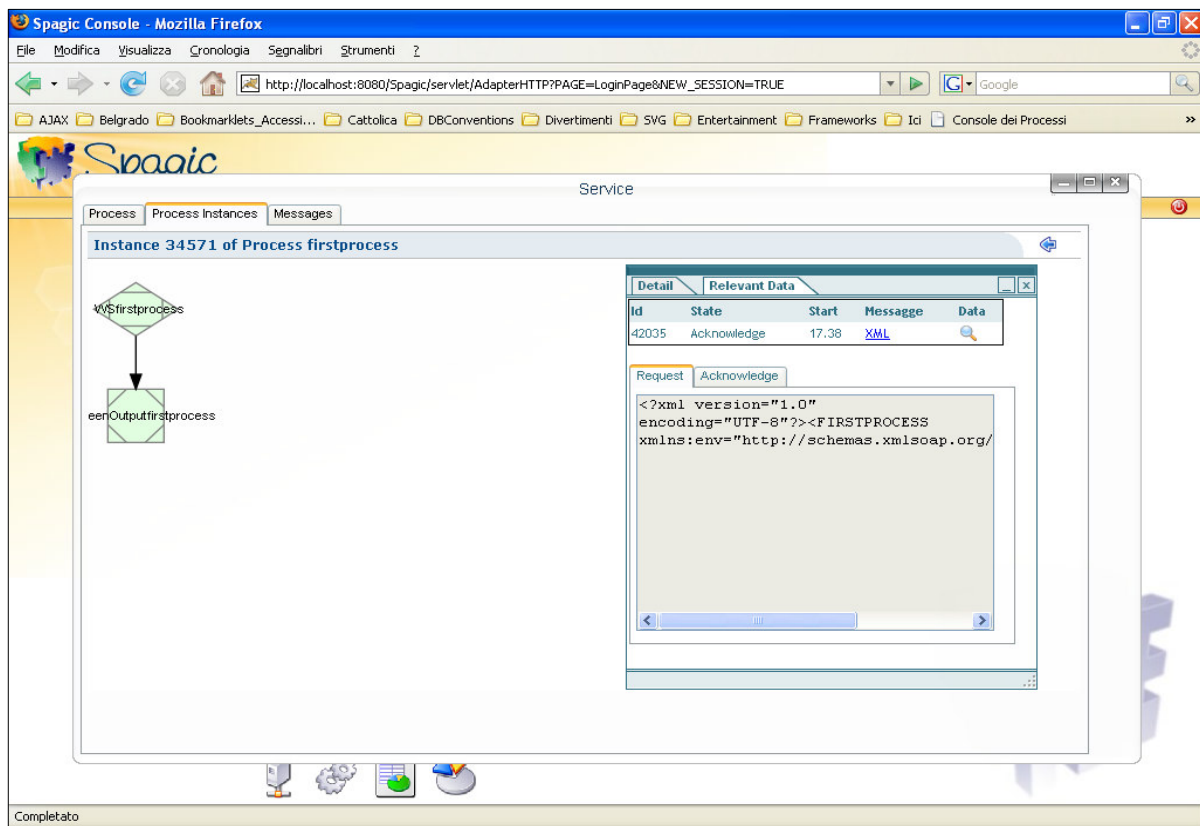
8.5 Process Instance Graph

Selecting the  icon from the process instances list it's possible to display the graph representing the execution of the process.

The color of the component represents the component state: the green color means *executed*, the yellow color means *active* and the red means error.

Selecting the elements on the graph it's possible to visualize the window containing *Detail* and *Relevant Data*.

The *Relevant Data* tab contains the messages having as destination the component selected. Selecting the *XML* link in the list, the XML message will be visualized; in this case *Request* and *Acknowledge*.



The screenshot shows the Spagic Console interface in a Mozilla Firefox browser window. The main window displays the 'Process Instance Graph' for 'firstprocess'. The graph shows a flow from 'WSfirstprocess' to 'eenOutputfirstprocess'. A detailed view of the process instance is shown, including a table of messages and an XML message viewer.

Id	State	Start	Message	Data
42035	Acknowledge	17.38	XML	

The XML message viewer shows the following content:

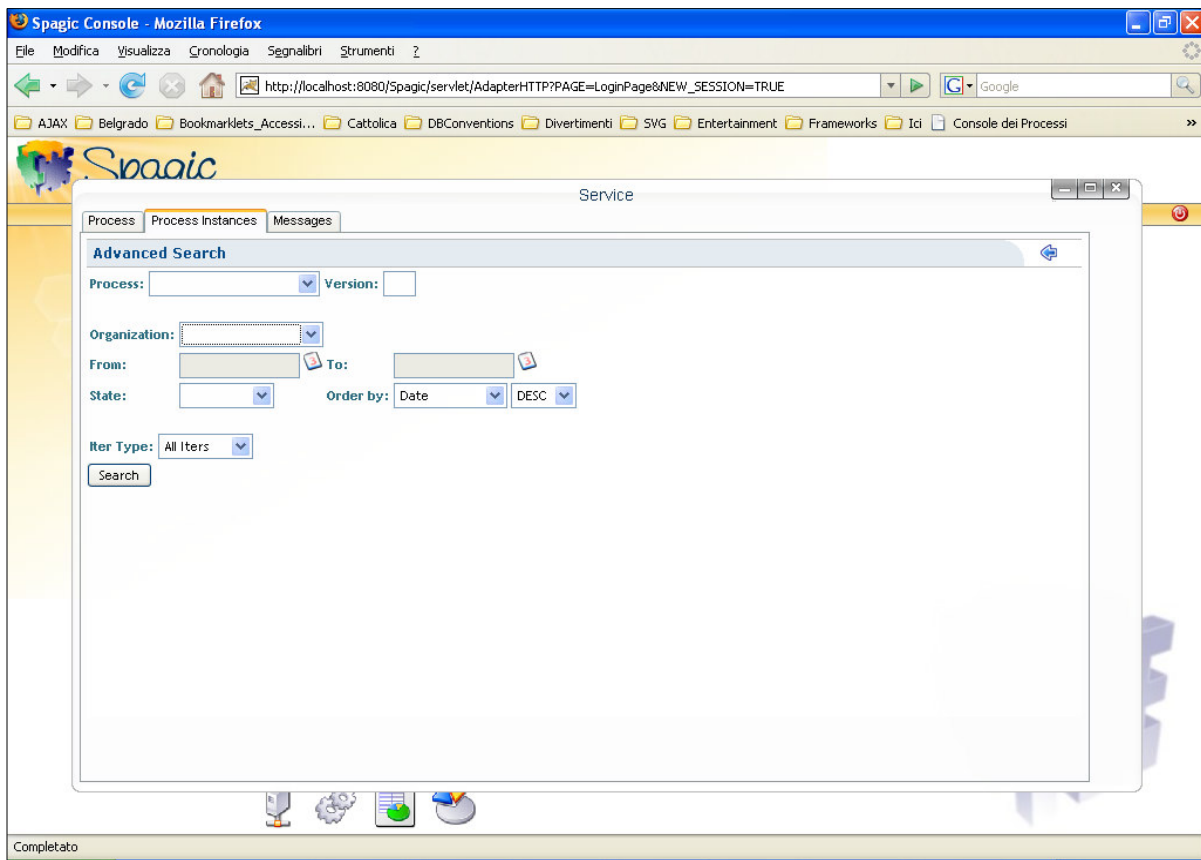
```
<?xml version="1.0"
encoding="UTF-8"?><FIRSTPROCESS
xmlns:env="http://schemas.xmlsoap.org/
```


8.6 Advanced Search

Selecting the icon *Advanced Search* in the *Process Instances List*, at the top of the page, the *Advanced Search* page is displayed.

The page contains:

- *Process*
- *Organization*
- *From and To*
- *State*
- *Order by*
- *Iter type*



9 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.