Spagic Getting Started

Author Daniela Butano
Gianfranco Boccalon

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.		
Version/Release n° :	1.2	**Date**	03/10/2007
Description	Changed the configuration steps for datasources: from the release 1.0.1 they are JNDI based. Section 5.4: added the instructions about the taxonomyVisible attribute. Section 5.4: added the instructions for installing the JDBC driver in Tomcat. Section 5.1: changed the user for the backup database.		
Version/Release n° :	2.0	**Date**	21/01/2008
Description	Updated to Spagic 2.0.0.		
Version/Release n° :	2.1	**Date**	29/07/2008
Description	Updated to Spagic 2.2.0.		
Version/Release n° :	2.2	**Date**	10/10/2008
Description	Update some sections relative to freebXML		

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:

- **Spagic 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).
- **Spagic 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.
  The platform supports also execution of BPEL processes through the open source Apache ODE BPEL engine.
- **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 design, deploy, run and monitor of a very simple process.
For more details about Spagic Studio environment see the document *Spagic Studio User Guide.doc.*
For more details about Spagic Console, see the document *Spagic Studio User Guide*.doc.

4 Requirements

Required Tools	URL for download/Notes	Spagic Studio	Spagic Console
Database	MySQL Oracle	Required	Required
Eclipse Europa for Java EE Developers	http://www.eclipse.org/downloads/moreinfo/jee.php	Required	
GraphViz	http://www.graphviz.org/	Required	
jUDDI	http://ws.apache.org/juddi/	Optional	Optional
freebXML	http://ebxmlrr.sourceforge.net/	Optional	Optional
JDK 1.5.0_11 or later	http://java.sun.com/	Required	Required
Apache Tomcat 5.5.17	http://tomcat.apache.org		Required
Mozilla Firefox 2.0.0.x	http://www.mozilla.com		Required

5 Installation

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

In this document we will use the notation *{SERVICEMIX_VERSION}* for referring to the ServiceMix release used.

Current ServiceMix release is 3.2.1.

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) or PostgreSQL (release 8.1)
2. Create a new schema "*spagic"* and a user "*spagic"* with password "*spagic*" 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 "*spagic-bck"* and assign to the user "*spagic"* the permissions for writing into the schema.
6. Generate the tables using the *"spagic-bck-<database type>.ddl"* released in the *spagic-metadb* package.
7. If you installed the MySQL Server and the processes' s messages contain the attachments bigger than 1 MB, you have to configure the variable max_allowed_packet = 16M into the file: {*MySQL Server path*}*/my.ini* and restart the server.

5.2 Service Manager

To install the Service Manager follow these steps:

1. To install ServiceMix, copy the folder "*apache-servicemix-{SERVICEMIX_VERSION}*", 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-{SERVICEMIX_VERSION}\hotdeploy".* 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-{SERVICEMIX_VERSION}\conf\jndi.xml"* file so that the datasource "*java:comp/env/jdbc/metadb*" links to the database just created.
3. **If you installed an Oracle database instead of MySQL** then configure the "*apache-servicemix-{SERVICEMIX_VERSION}\conf\hibernate.cfg.xml"* file so that the property "***hibernate.dialect***" has the value "org.hibernate.dialect.OracleDialect".
4. To launch ServiceMix, launch the command "*bin\servicemix* from the *apache-servicemix-{SERVICEMIX_VERSION}"* 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 application, so you have simply to get it from Spagic distribution (from the package *spagic-studio*) and unzip it.
   In this document we will refer to the Eclipse folder as *SPAGIC_STUDIO_HOME*.
   If you want to create Spagic Studio from scratch, instead of using the distribution package, please refer to the document "*How to cook your Spagic Studio*".

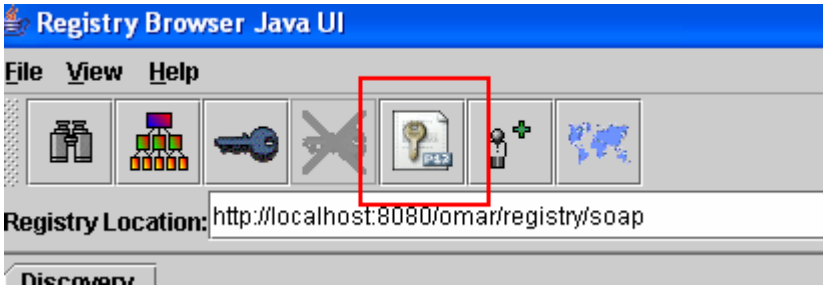
3. Spagic Studio needs to connect to Spagic metadabase. Ensure that the database is running and that permissions are configured properly on database server.

4. **Optional step**: an installation of Tomcat with JUDDI installed, if you want to publish your services in UDDI Registries. To install JUDDI refer to its installation documentation. After the installation, it needs to create manually in the "*publisher*" table a new publisher with "*publisherId*" set to "*eng*", the same value configured in the Service Registries List Wizard on Spagic Studio (for detail see the document "*Spagic Studio User Guide*", section "*Manage Process Publication on Service Registries*").

5. **Optional step**: an installation of Tomcat with freebXML installed, if you want to publish your services in ebXML Registries. To install freebXML refer to its installation documentation (http://ebxmlrr.sourceforge.net/wiki/index.php/Install)

   After the installation you should execute the following steps:

   1. launch the omar web application (http://localhost:8080/omar);
   2. create a new user account following the instructions visualized on the web page (use alias: eng, password: engeng);
   3. import the key (*.p12) generated to your Web browser following the instructions visualized on the web page;
   4. import the key (*.p12) into the keystore located in the *<OMAR_HOME>/ jaxr-ebxml/security* directory, using the omar GUI application in the following way. Open the command line, run the command *build run.browser*, select the *Registry Location* and click the following button:



   The Registry Browser Java Gui will display the following window:

   The "*Keystore Alias*" field has the value "*eng*", the same value configured in the Service Registries List Wizard on Spagic Studio (for detail see the document "*Spagic Studio User Guide*", section "*Manage Process Publication on Service Registries*"). The "*Private Key Password*" field has the value "*engeng*" (the same configured on Spagic Studio), the "*File to Import*" contains the path of the file **.p12* just generated.

   Note: the keystore password is "*ebxmlrr*".

   5. Copy the keystore containing the key with alias eng, from the *<OMAR_HOME>/ jaxr-ebxml/security* directory to *keystore* directory located into the "*org.spagic.ui*" plugin in *SPAGIC_STUDIO_HOME/plugins*

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. Install the Apache Tomcat 5.5.17; to install it refer to its installation documentation (http://tomcat.apache.org).
2. Install the database JDBC driver for the database you are using in the Tomcat *common\lib* folder:
   - If you are using MySQL, please use *mysql-connector-java-5.0.7-bin.jar* or later.
   - If you are using Oracle 9i/10g use the JDBC driver for JDK 1.4 and Oracle 10g (it works also on 9i). The driver name is *ojdbc14.jar*.

   You can retrieve both the drivers from the *lib* folder of the ServiceMix released in Spagic.
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:
   - **If you installed the Console on a different machine from the ServiceMix machine**:
     a) Update the file *\SpagicConsole\WEB-INF\conf\jmx\servers.xml* to set the *jmxUrl* to the URL for connecting to ServiceMix by JMX. The URL is written by ServiceMix on its console, on the startup.
     b) Update the file *\SpagicConsole\WEB-INF\conf\console\console.xml* to set the *locationRestartWS, locationBackupWS and locationDeleteWS* parameters to the URL where you installed ServiceMix. These are the URLs of the restart, backup and clean database operations.
   - **If you installed the database on a different machine from the Tomcat machine**, update the file *\SpagicConsole\META-INF\context.xml* so it could link to the Spagic database (with the JNDI name *jdbc/spagic*).
   - **If you installed an Oracle database instead of MySQL** then configure the "*\SpagicConsole\WEB-INF\classes\hibernate.cfg.xml*" file so that the property "***hibernate.dialect***" has the value "*org.hibernate.dialect.OracleDialect*".
   - **If you used UDDI**:
     a) Update the file *\SpagicConsole\WEB-INF\classes\Serviceregistry.properties* to set the URL of jUDDI application.
     b) Configure the "*\SpagicConsole\WEB-INF\conf\console\console.xml"* file so that the attribute "*taxonomyVisible*" has the value *"TRUE"* and the attribute "*serviceAttribute*" has the value "*uddi*"
   - **If you used ebXML**:
     a) Update the file *\SpagicConsole\WEB-INF\classes\ jaxr-ebxml.properties* to set the URL of freebXML application (jaxr-ebxm.soap.url property)
     b) Configure the "*\SpagicConsole\WEB-INF\conf\console\console.xml"* file so that the attribute "*taxonomyVisible*" has the value *"TRUE"* and the attribute "*serviceAttribute*" has the value "*ebXML*"

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 documents *Spagic Studio User Guide* and *Spagic JBI 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 graphviz application, select the service registry.

To create a new project, select *File->New->Project->Spagic 2.0->Spagic 2.0 Project*, insert a Project Name, *StartSpagic*, click *Finish*.

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

- **Bpel:** contains all BPEL files and artifacts generated when using BPEL technology;
- **Deployables:** contains the following files produced by Spagic Studio:
  - **<Process name>_v_<version>-sa.zip**: service assembly to be deployed in ServiceMix.
  - **<Process name>_v_<version>.deploy**: deploy file necessary to publish the process on the Spagic MetaDB, if you don't want to use Spagic Studio but a Spagic command line tool (necessary for example when migrating the processes from test environments to production environments).
  - **<Process name>_v_<version>.properties**: optional file created if within the processes are used some configurable parameters.
- **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;
- **WsdlFiles**: 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

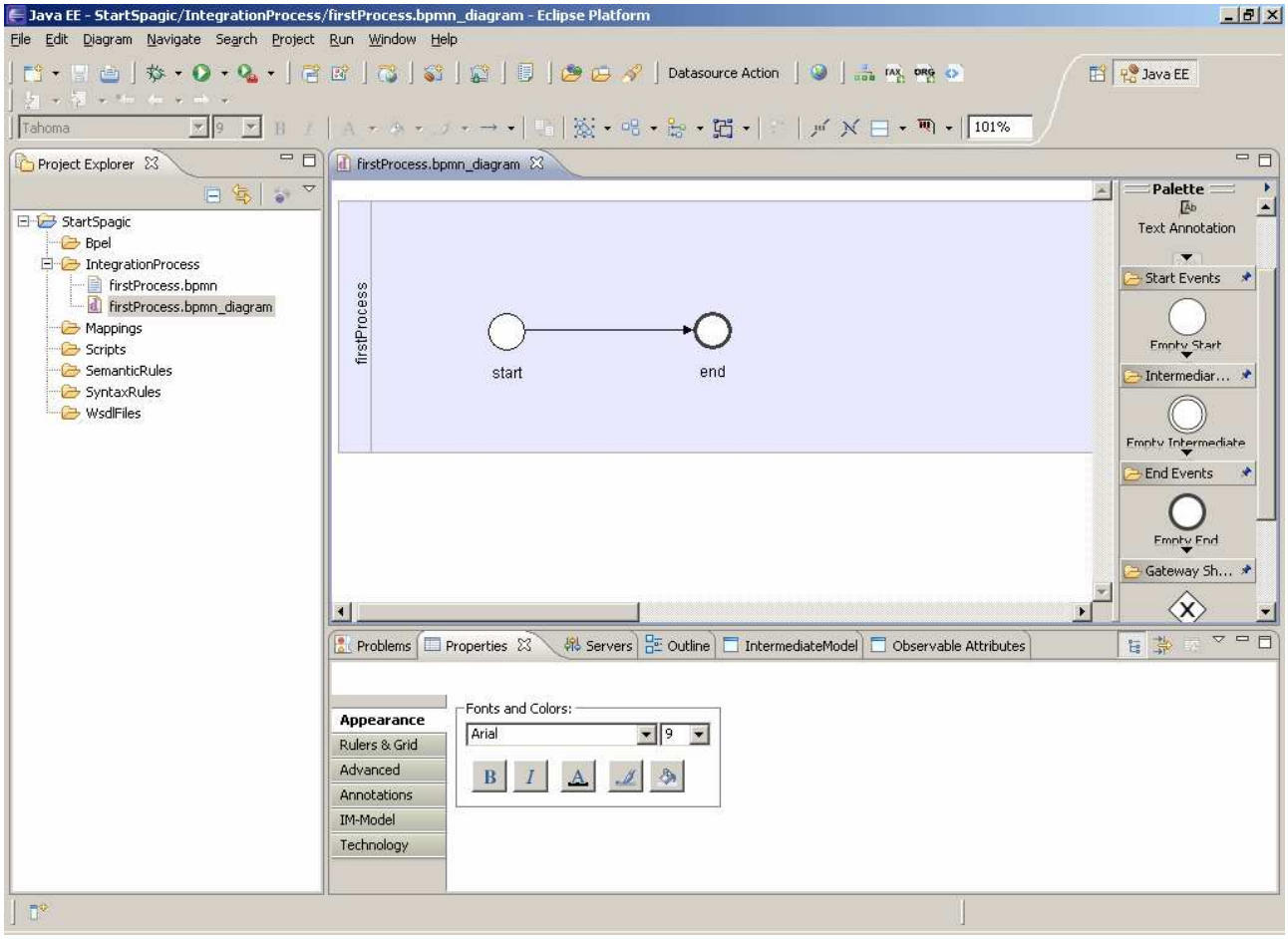
To define a new process, select the *Integration Process* folder, open the context menu and *New->Other->Bpmn Diagram.* Insert a file name, for example *firstProcess.*

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

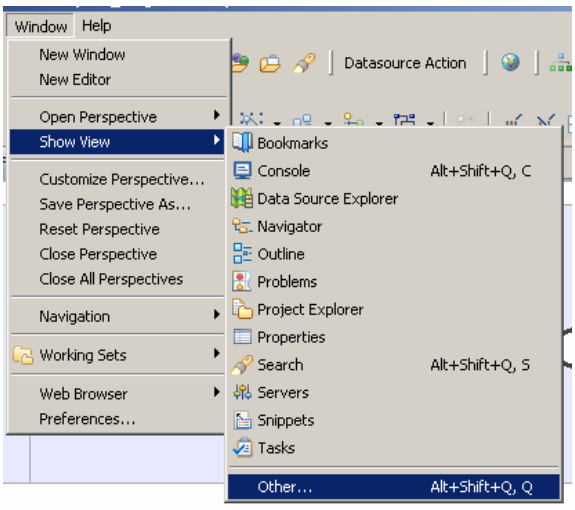
To design a very simple process, select, from the component palette located at the right of the editor, an *Empty Start* element and put it to the editing area. Then put an element *Empty End* on the same process, and link the start element with the end element.

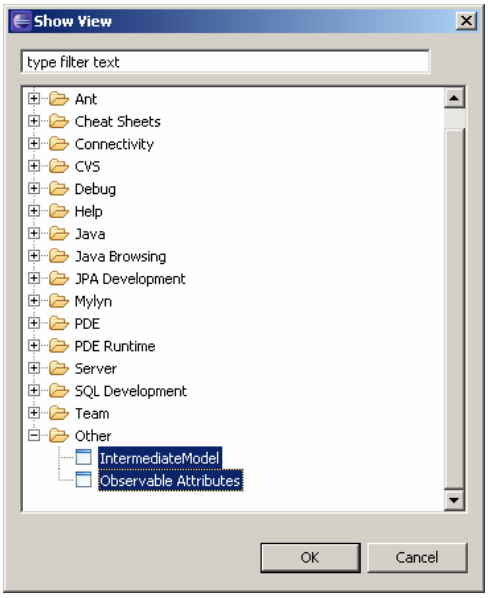
Double click on the pool name and change it to “firstProcess”; assign also a name to the start element (“start”) and to the end element (“end”).

The result should be similar to the following screenshot.



Before proceeding you have to add 2 views to the Editor: choose the command *Window->Show View->Other* and select the views *Intermediate Model* and *Observable Attributes*.





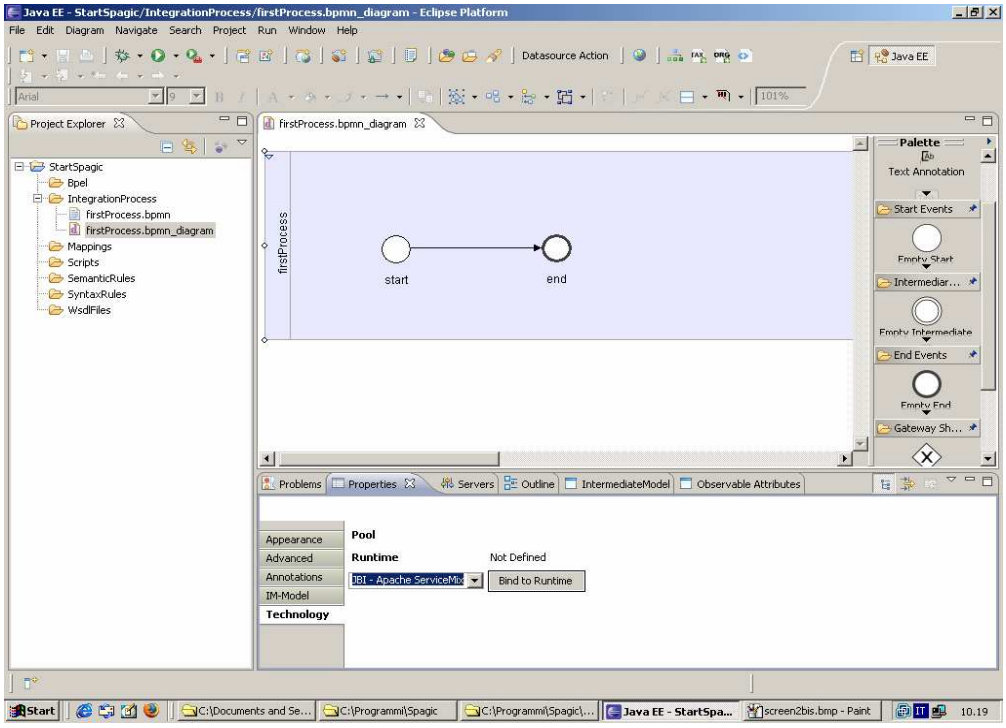
Now you are ready to choose the technology that Spagic will use to generate the process: remember that Spagic 2 now supports more technologies (JBI and BPEL), and it's in the next step that you select the proper technology for your process.

According to the technology that you'll choose you have to follow different paragraphs:

- For JBI read [6.2.1] JBI Technology
- For BPEL read [6.2.2-6.2.3] BPEL Technology, Complete with further technical definitions

6.2.1 JBI Technology

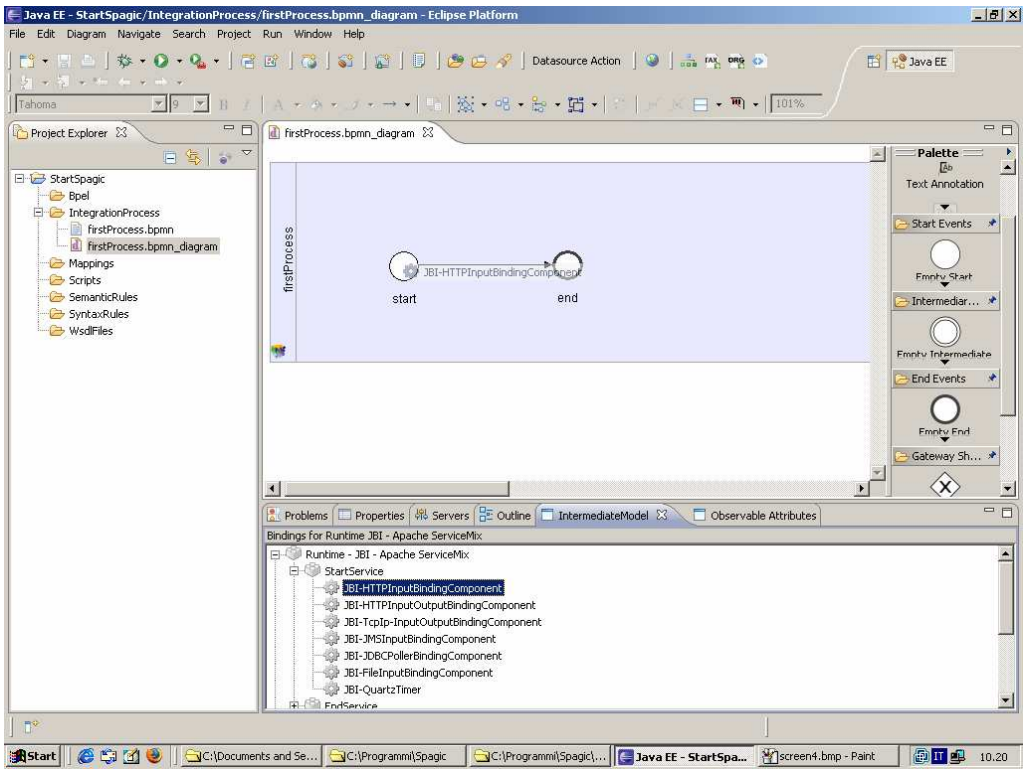
Select the pool *firstProcess* (it should be unique) and in the view *Properties* select the Tab *Technology*: here you should found a combo containing the supported technologies.
Choose "*JBI – Apache ServiceMix*" and select *Bind to Runtime*: now your process has the JBI technology associated, and you are ready to use the services provided by this specific technology.



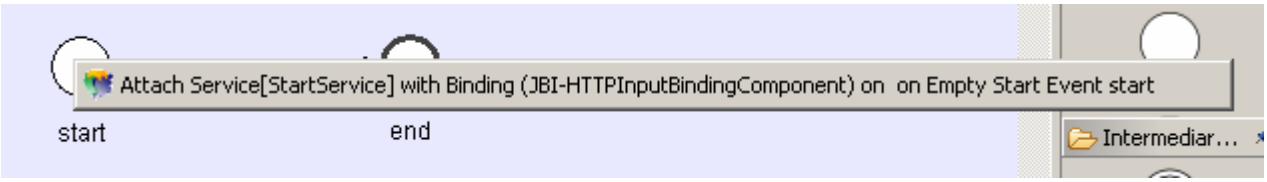
In the next steps you have to specify which specific technological services are bound to the steps of your BPMN diagram.

To do this, select the view *IntermediateModel*: you should see the list of all services available for JBI.

Open the *StartService* category and drag the service *JBI-HTTPInputBindingComponent* on the step *start*.



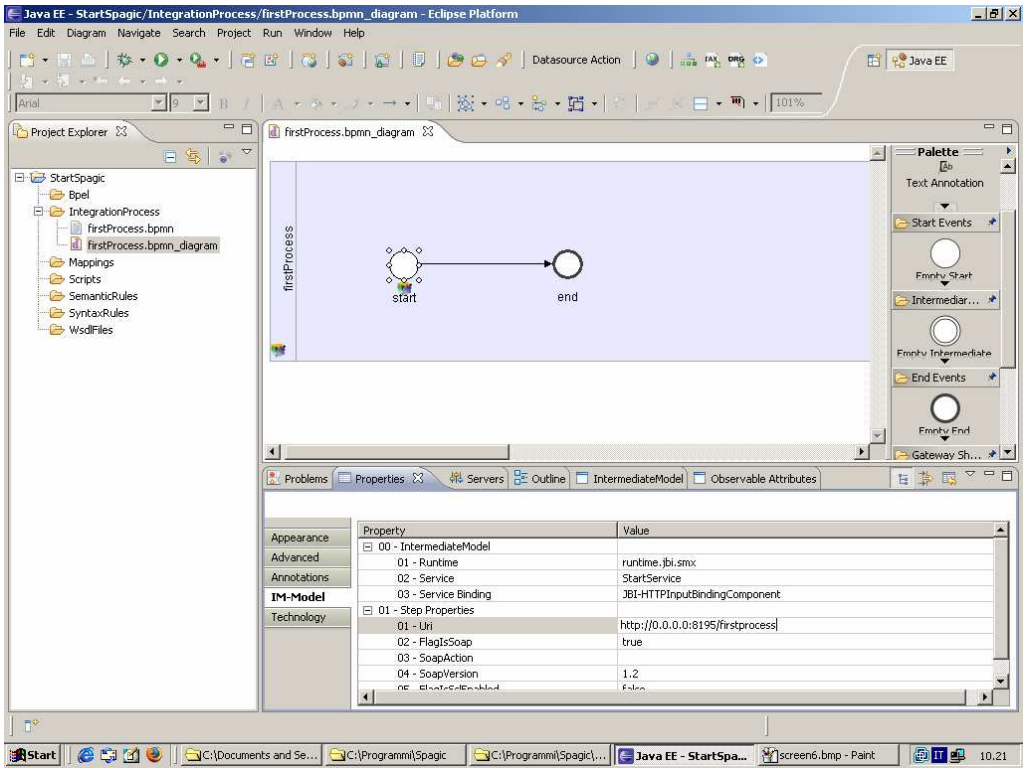
Spagic Studio asks if you want to associate this BPMN step with the JBI service (also called *Binding*) *JBI-HTTPInputBindingComponent* that is an HTTP Binding Component configured as In-Only.



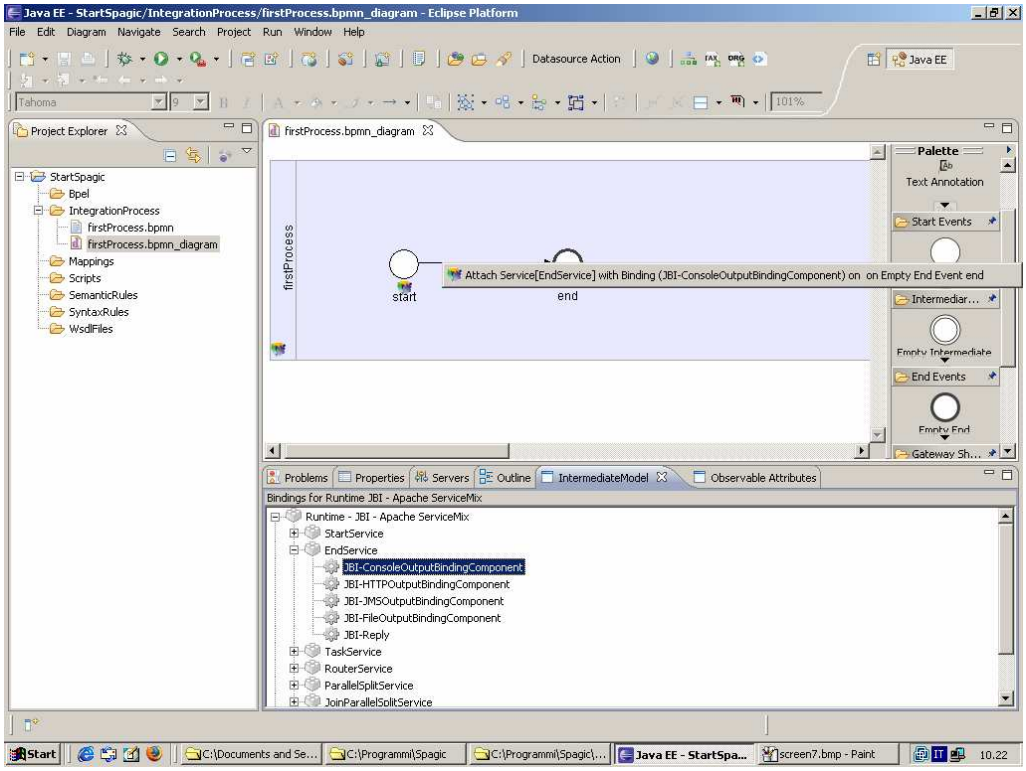
You are ready to configure the HTTP BC of the first step.

Select the *start* step and configure all the properties in the *Properties* tab, in the item *IM-Model.*

In the *firstProcess* sample you need simply to insert the value http://0.0.0.0:8195/firstprocess in the property *Uri*.



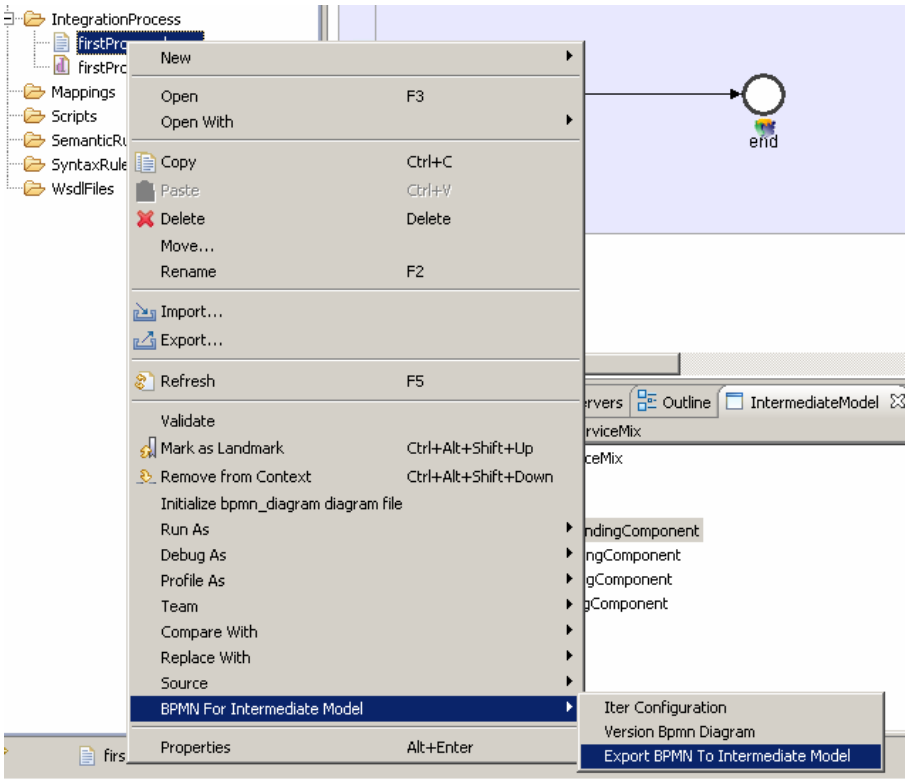
Configure also the final process' step with the binding *EndService->JBI-ConsoleOutputBindingComponent*.



The process is completed: the BPMN diagram was annotated with all the technological information needed to perform deploy and monitoring.

To perform these activities you need to execute an intermediate step that consists of generating an Eclipse model (called *Intermediate Model*) that allows in future using the Spagic features also with editors different from the BPMN editor.

Select the BPMN file (not the diagram, the file that ends with *.bpmn*) and after right click on it, choose *BPMN For Intermediate Model->Export BPMN To Intermediate Model*.

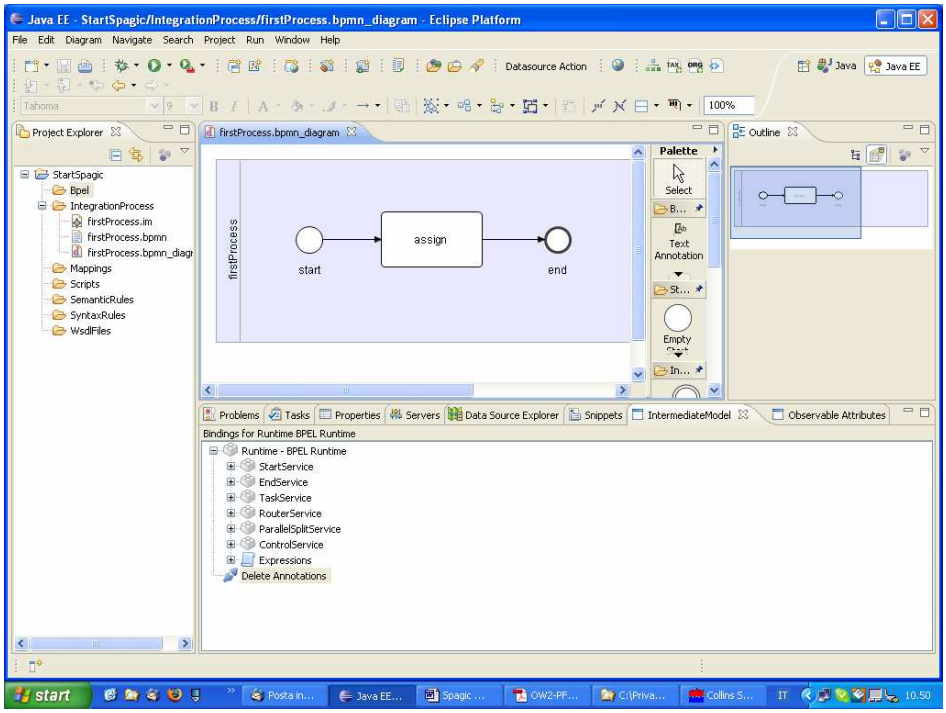


You should see a new file called *firstProcess.im*: this is the file necessary for deploy and publication on meta-db.

Every time you change the BPMN diagram, you must regenerate the Intermediate Model.

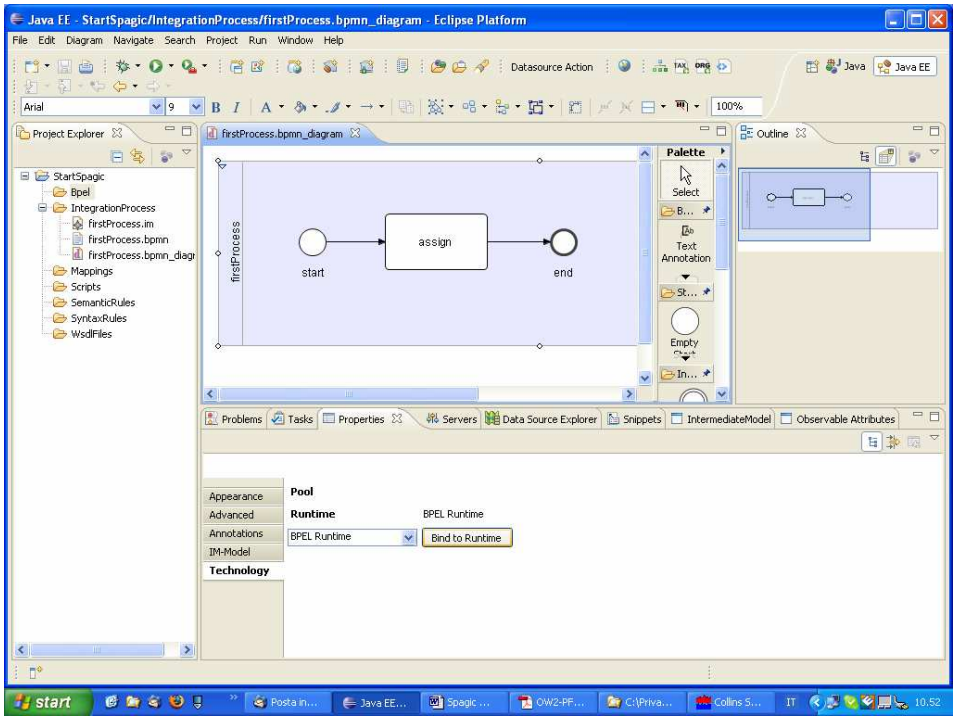
6.2.2 BPEL Technology

Modify the previous process adding a new step. Select, from the component palette located at the right of the editor, a *Task* element and put it to the editing area; assign it the name “assign”. Link the start element with the task element and the task element with the end element.



Select the pool *firstProcess* (it should be unique) and in the view *Properties* select the Tab *Technology*: here you should found a combo containing the supported technologies.

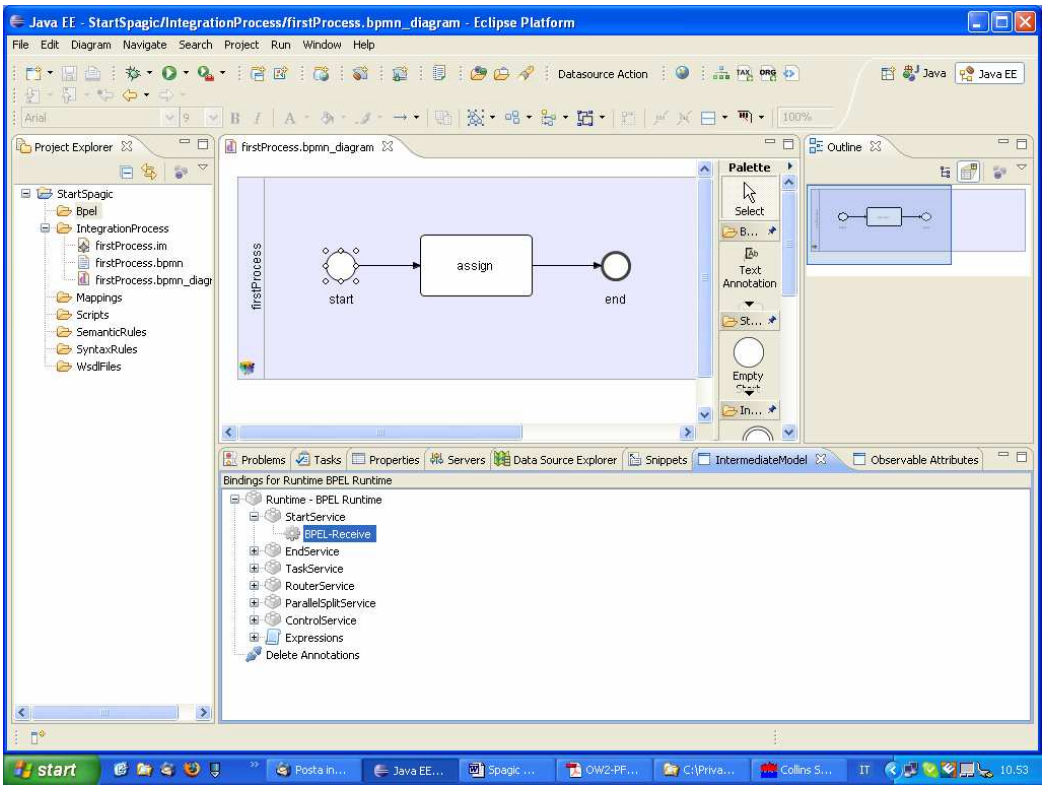
Choose "*BPEL Runtime*" and select *Bind to Runtime*: now your process has the BPEL technology associated, and you are ready to use the services provided by this specific technology.



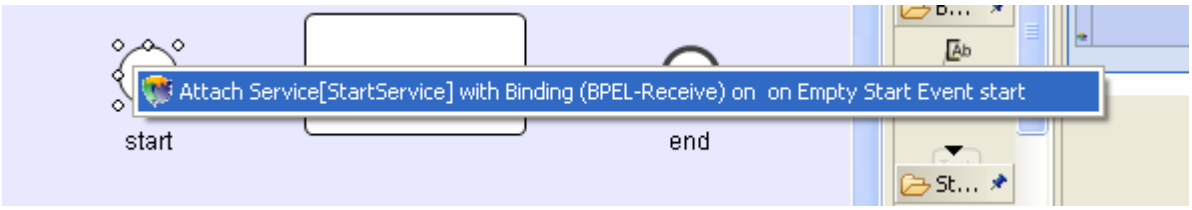
In the next steps you have to specify which specific technological services are bound to the steps of your BPMN diagram.

To do this, select the view *IntermediateModel*: you should see the list of all services available for BPEL.

Open the *StartService* category and drag the service *BPEL-Receive* on the step *start*.



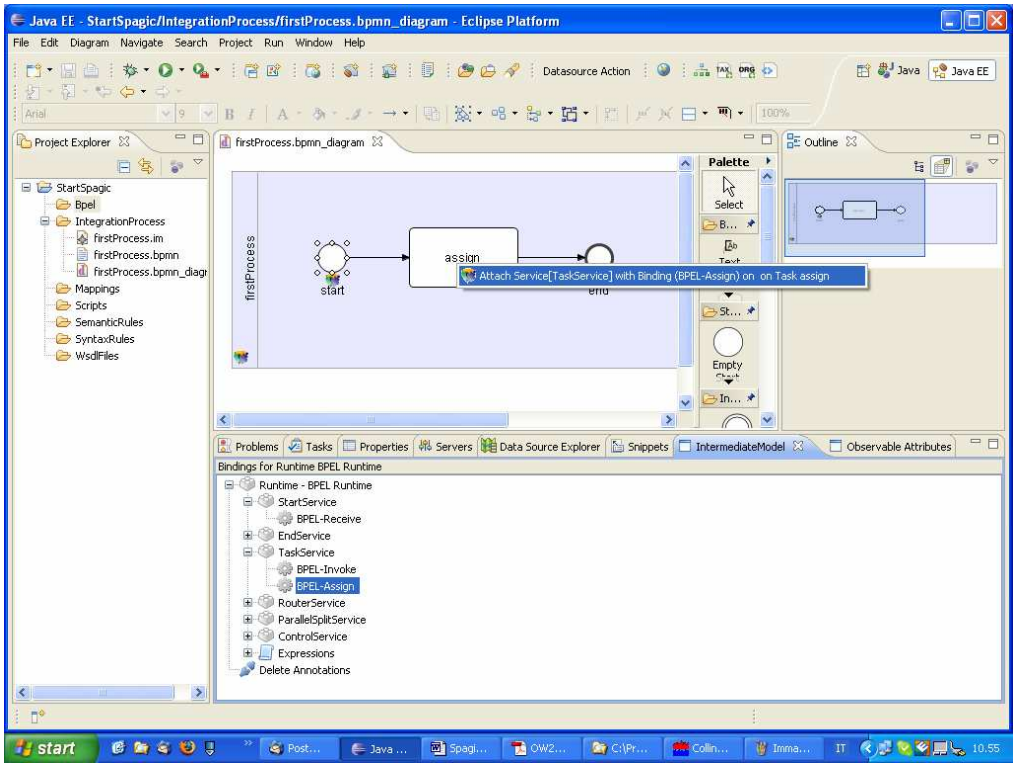
Spagic Studio asks if you want to associate this BPMN step with the BPEL service (also called *Binding*) *BPEL-Receive* that is a receive activity.



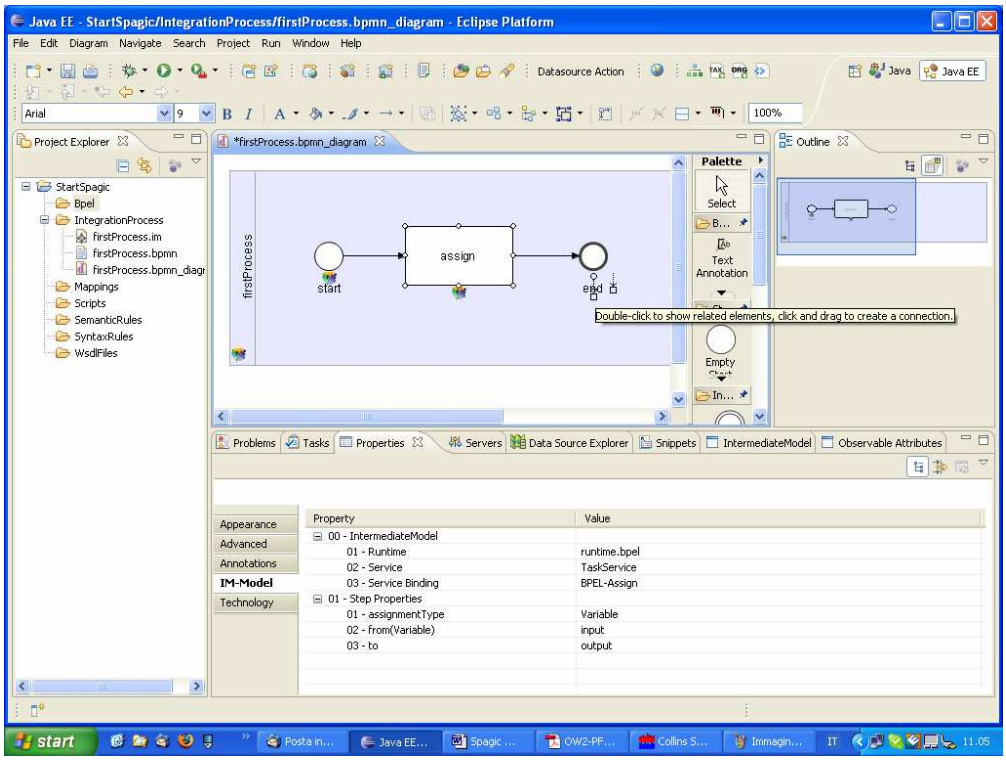
To display all the properties, select the *start* step, then the *Properties* tab, in the item *IM-Model.*

All properties have been configured.

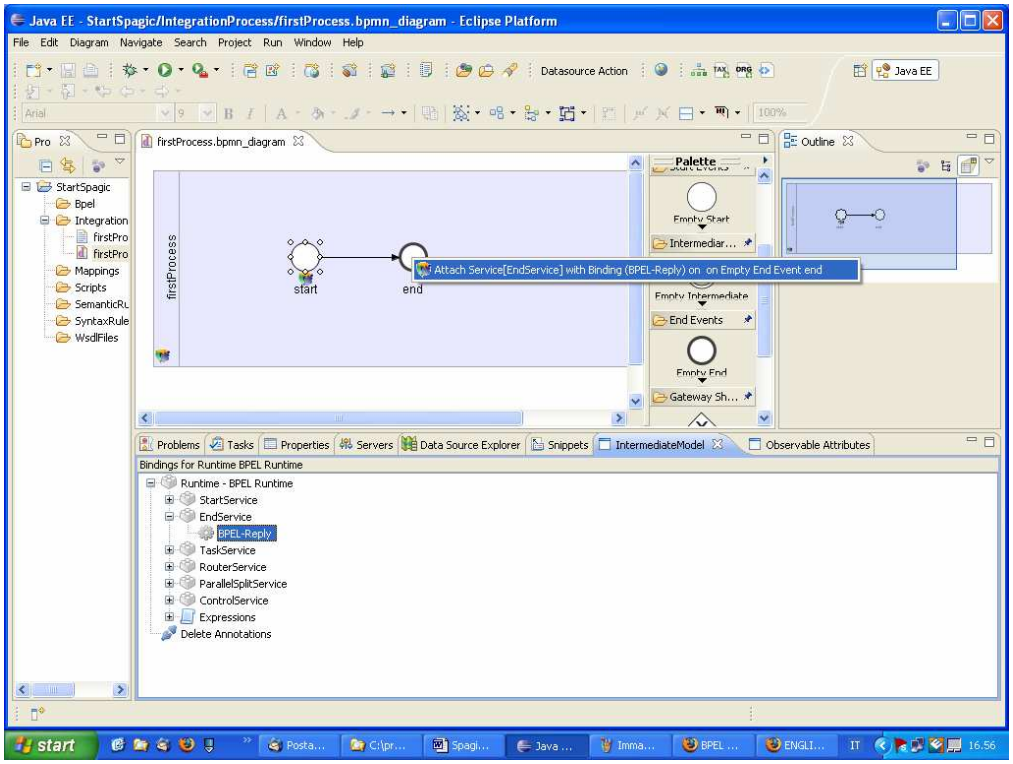
Configure the assign step with the binding *TaskService->BPEL –Assign*.



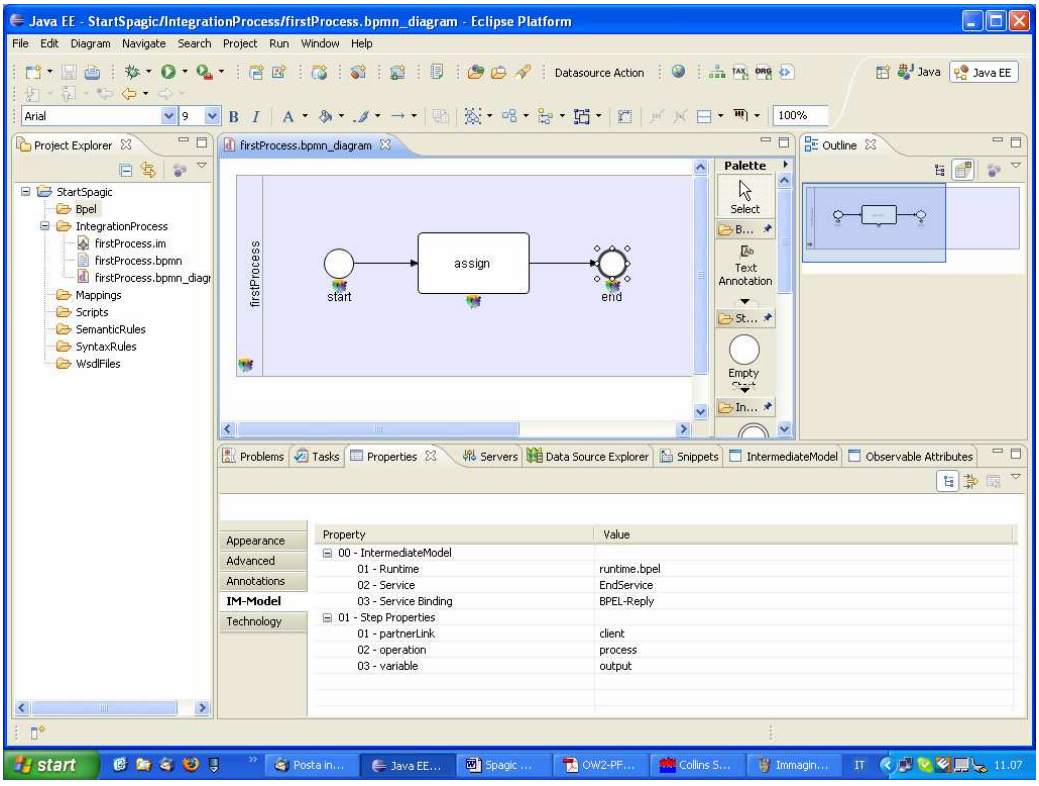
You have to configure the *assign* step: select it and configure all the properties in the *Properties* tab, in the item *IM-Model*. Set the property *assignmentType* with value *Variables*, the property *from(Variable)* with value *input*, the property *to* with value *output*.



Configure also the final process' step with the binding *EndService->BPEL-Reply*.



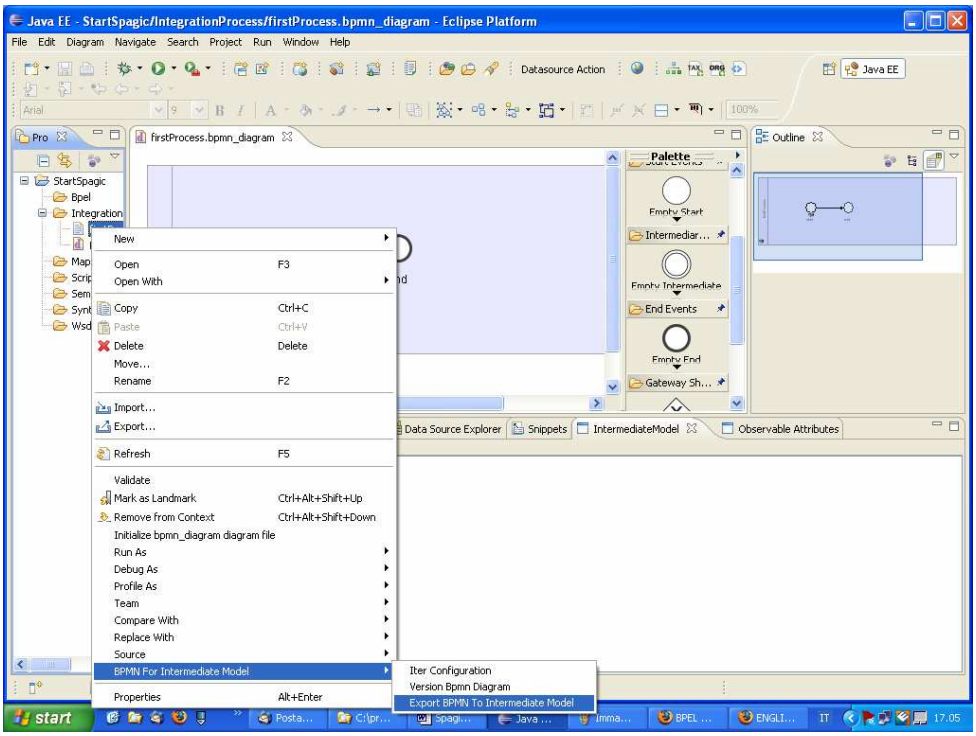
The last step has been configured.



The process is partially completed: the BPMN diagram was annotated with some technological information needed to perform deploy and monitoring.

To perform these activities you need to execute an intermediate step that consists of generating an Eclipse model (called *Intermediate Model*) that allows in future using the Spagic features also with editors different from the BPMN editor.

Select the BPMN file (not the diagram, the file that ends with *.bpmn*) and after right click on it, choose *BPMN For Intermediate Model->Export BPMN To Intermediate Model*.



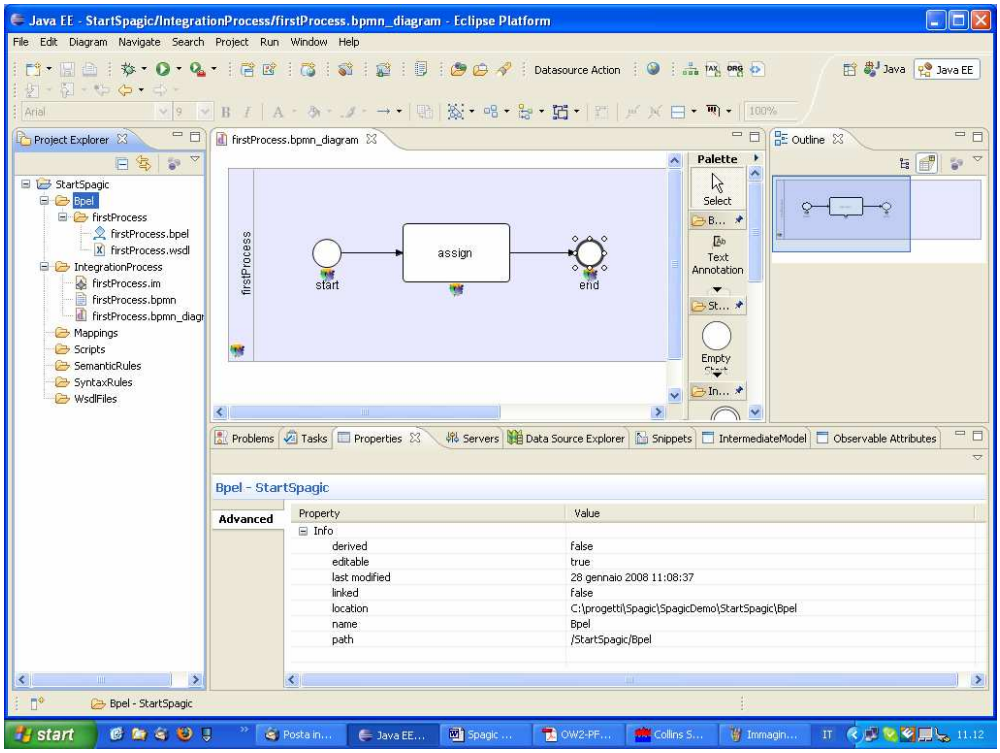
You should see a new file called *firstProcess.im*: this is the file necessary for deploy and publication on meta-db.

Every time you change the BPMN diagram, you must regenerate the Intermediate Model.

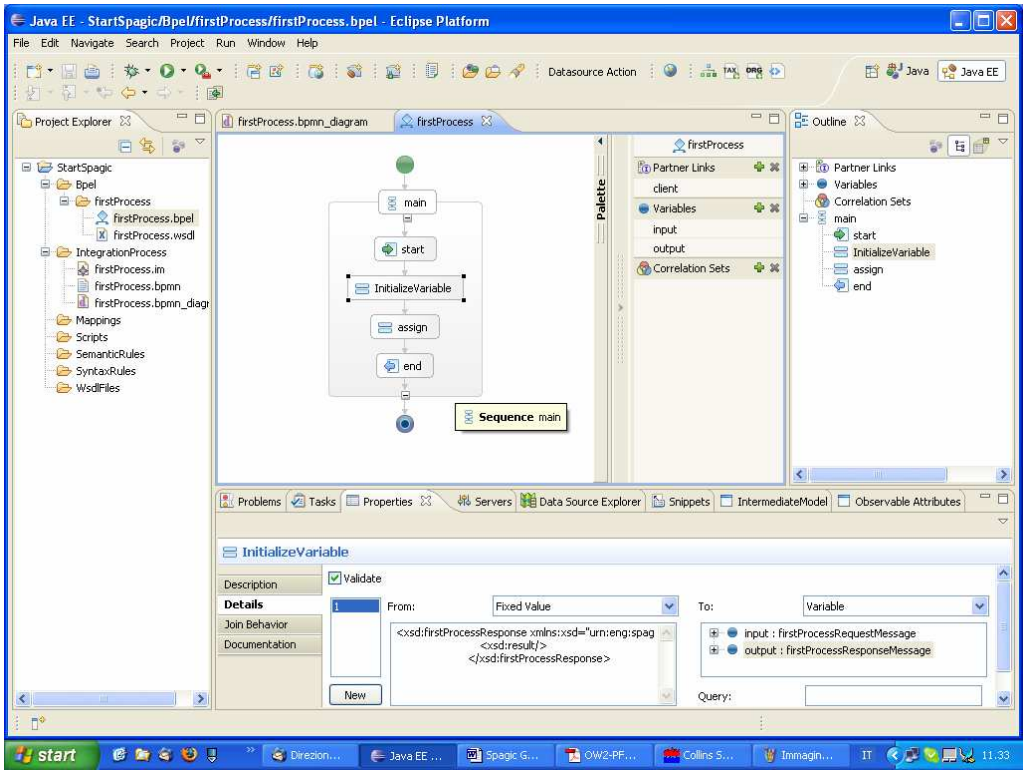
6.2.3 Complete with further technical definitions

The BPEL process is partially completed: the BPMN diagram was annotated with all the technological information needed to perform monitoring but we have to add further technical definitions. For this purpose, we preferred to use a specific designer: BPEL Designer.

In the editor area, select the *.im* file, open the context menu and select *Intermediate Model->Generate BPEL Model.*
Into the folder Bpel, the new folder *firstProcess* was generated, containing the files *firstProcess.bpel* and *firstProcess.wsdl*

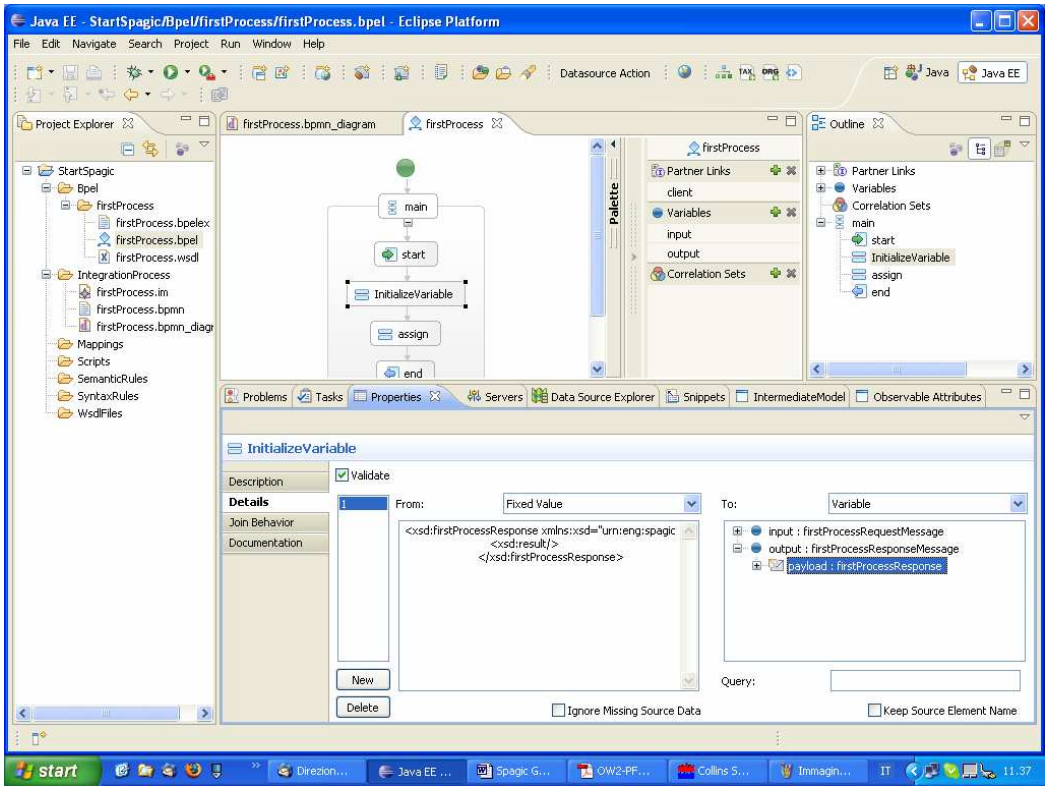


Select the *firstProcess.bpel*, open the context menu, select *Open With -> Business Process Editor*.

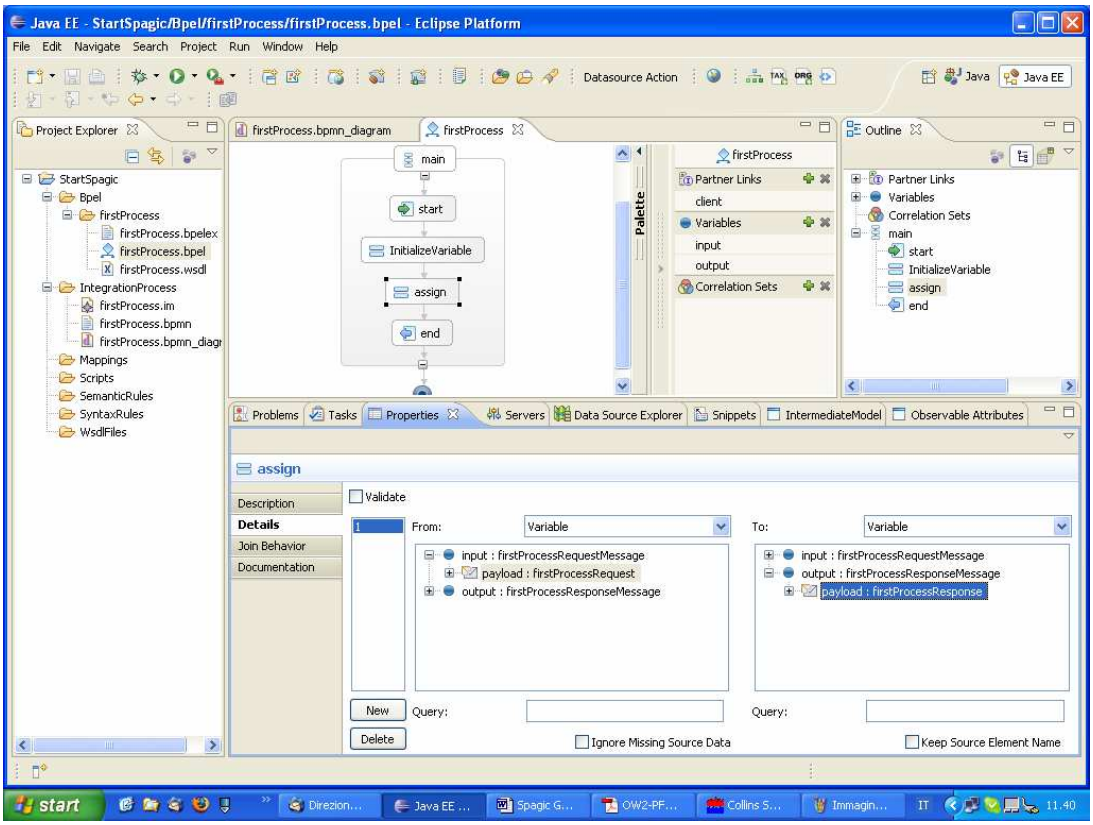


To complete the bpel process execute the following steps:

1. Select the *InitializeVariable* activity and then the tab *Properties* : specify the payload for the *output* variable;



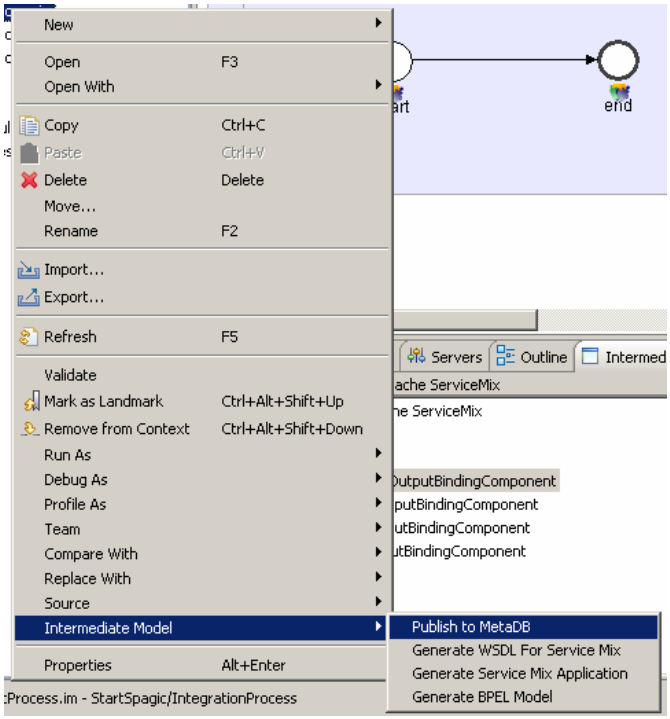
2. Select the *assign* activity and then the tab *Properties* : specify the payload for the *input* and *output* variables



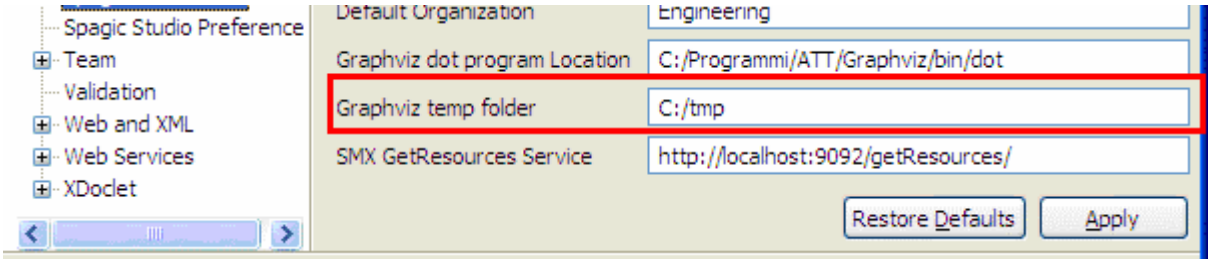
6.3 Publish the process

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, select the *.im* file, open the context menu and select *Intermediate Model->Publish to MetaDB.*

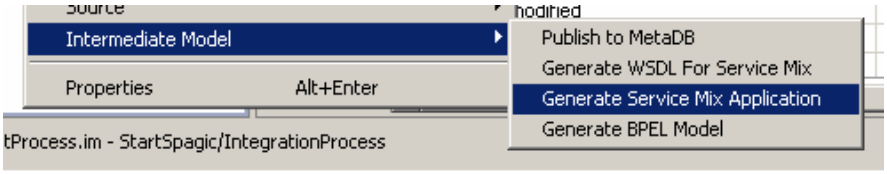


If you have some errors, verify that you installed *graphviz* tool and you created the temporary directory that should be configured in the Spago Preferences, in the item *Graphviz temp folder*.



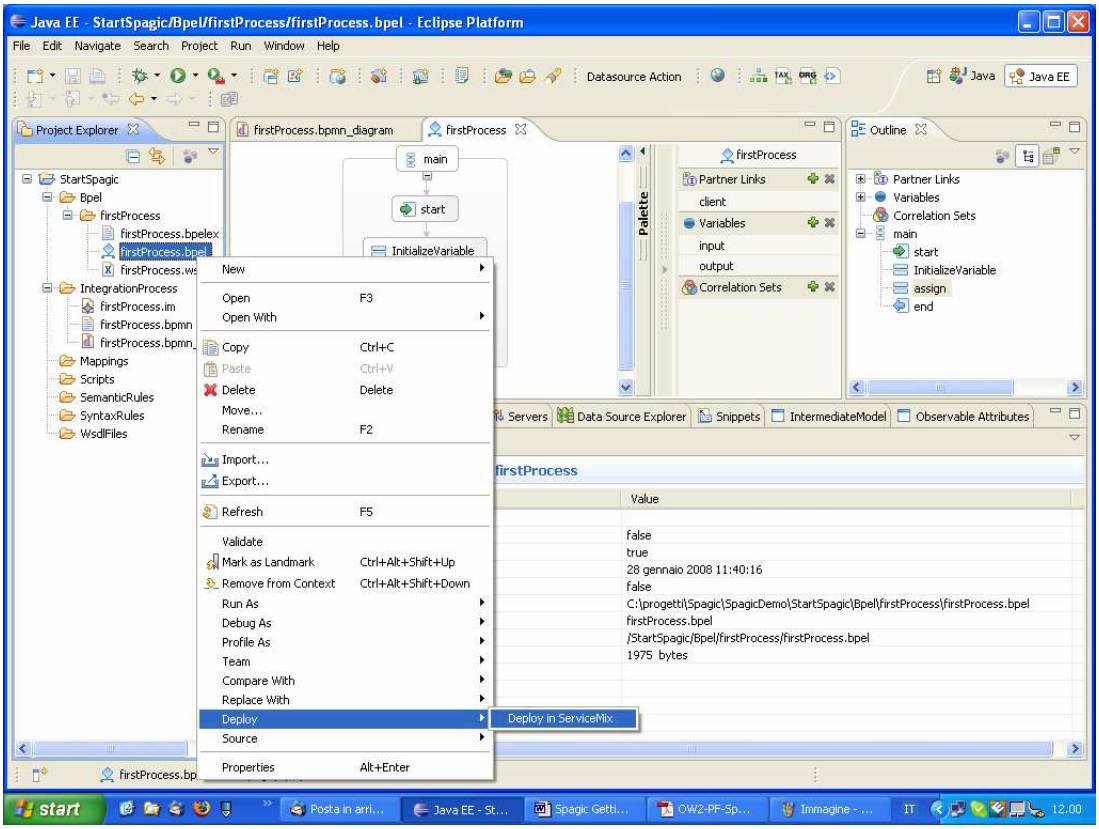
6.4 Deploy the process

To deploy the process implemented with **JBI technology**, in the editor area, open the context menu after selecting the .im file and select *Intermediate Model->Generate Service Mix Application.* The *firstprocess_v_0-sa.zip* will be created in the project *Deployables* folder.



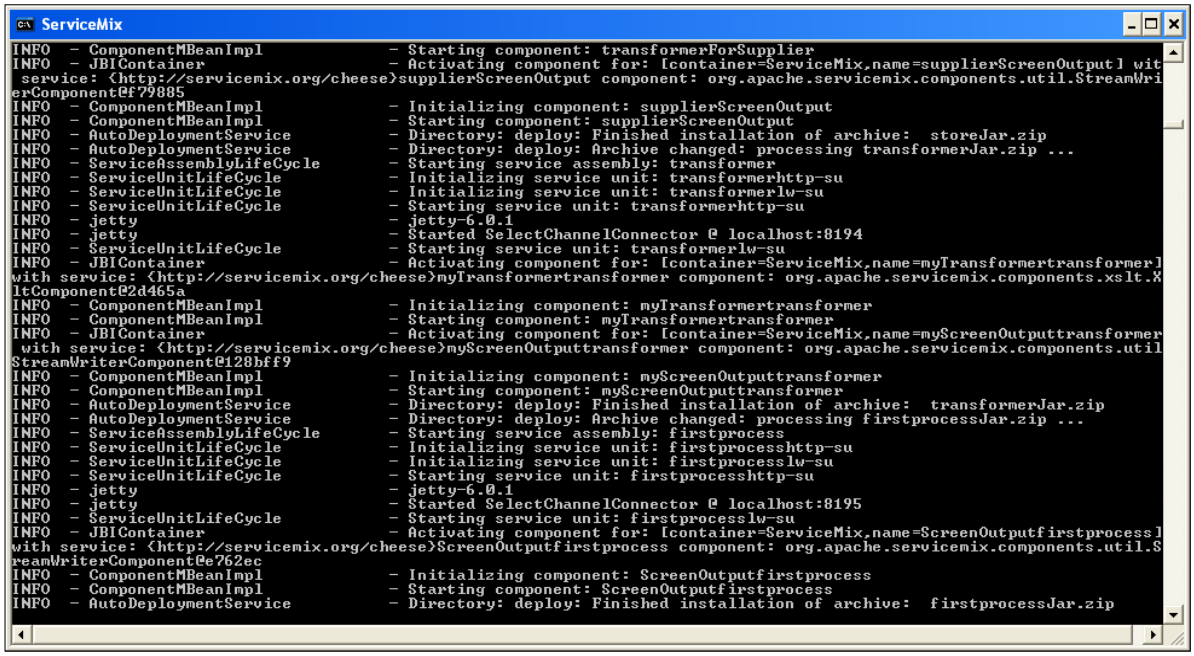
The label *_v_0* is assigned to the process because it's now possible to specify the process version: you have not specified this information, so the default version (0) is assigned.

To deploy the process implemented with **BPEL technology**, in the editor area, open the context menu after selecting the .bpel file and select *Deploy->Deploy in Service Mix.*



A pop-up window will be displayed, insert the process uri: *http://localhost:8214/firstProcess* and then click the *Ok* button. The *firstprocess-sa.zip* will be created in the BPEL folder.

Start ServiceMix launching the command *bin\servicemix.bat* (or .sh) *from* apache-*servicemix-{SERVICEMIX_VERSION}*.
After ServiceMix was started, copy the *.zip* generated in the *hotdeploy* folder and verify in the ServiceMix console that the process was deployed without errors.



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-clients*".
Use the page *client.html* to launch the JBI process or the page *clientBpel.html* to launch the BPEL process.

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

8 Monitor the Process

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

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

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

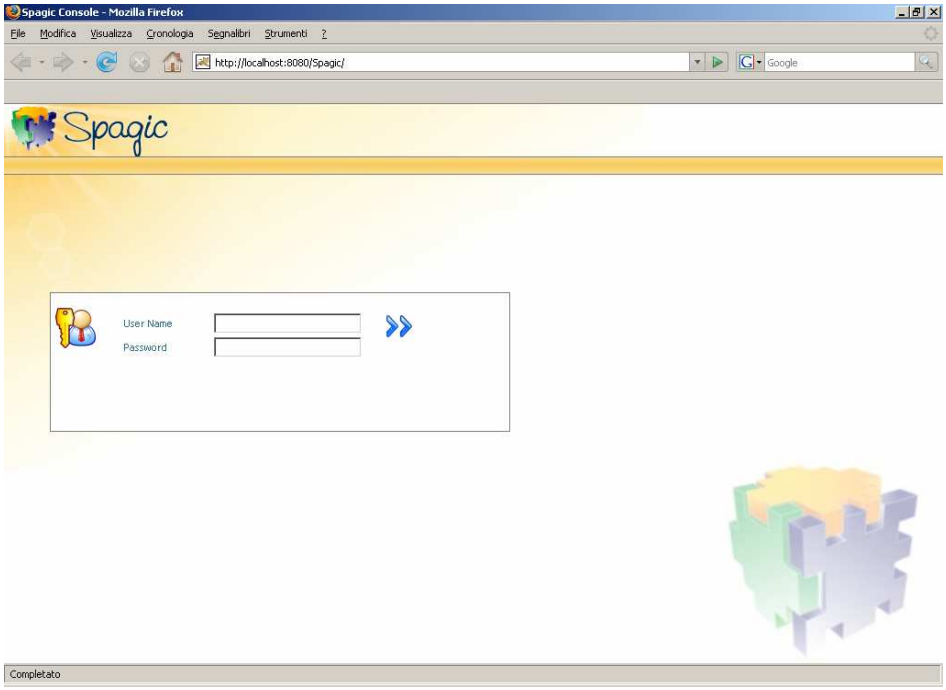
Start the Tomcat where you installed the Spagic Console.

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 s*pagic* and the password s*pagic* and then click the button .



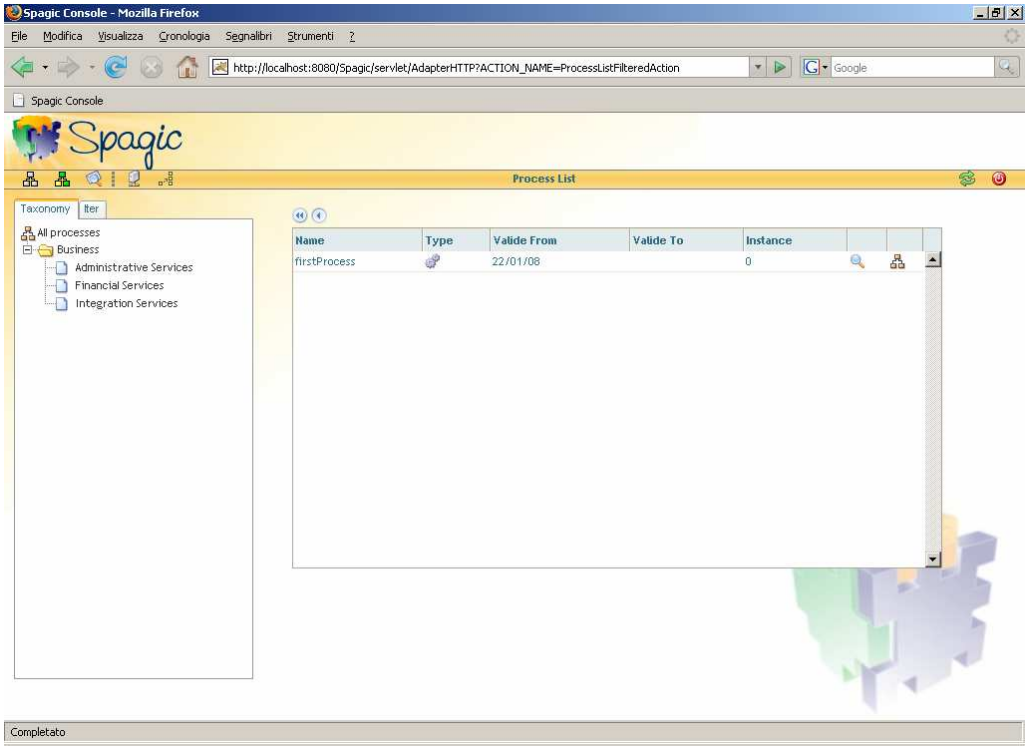
The next page contains the application menu.

8.2 Process List

Select the icon **Process List** () to view all processes and theirs configuration, properties, and static flow diagram. The list contains the following attributes:

- *Name*: process name.
- *Type*: process technology.
- *Valide From*: start validity date.
- *Valide To*: end validity date.
- *Instance*: number of process instances.
-  : displays the process detail containing the *description*, *version*, *organization*, *service name*.
-  : displays the process graph.

The process list should contain only *firstProcess*: the process just deployed and launched.

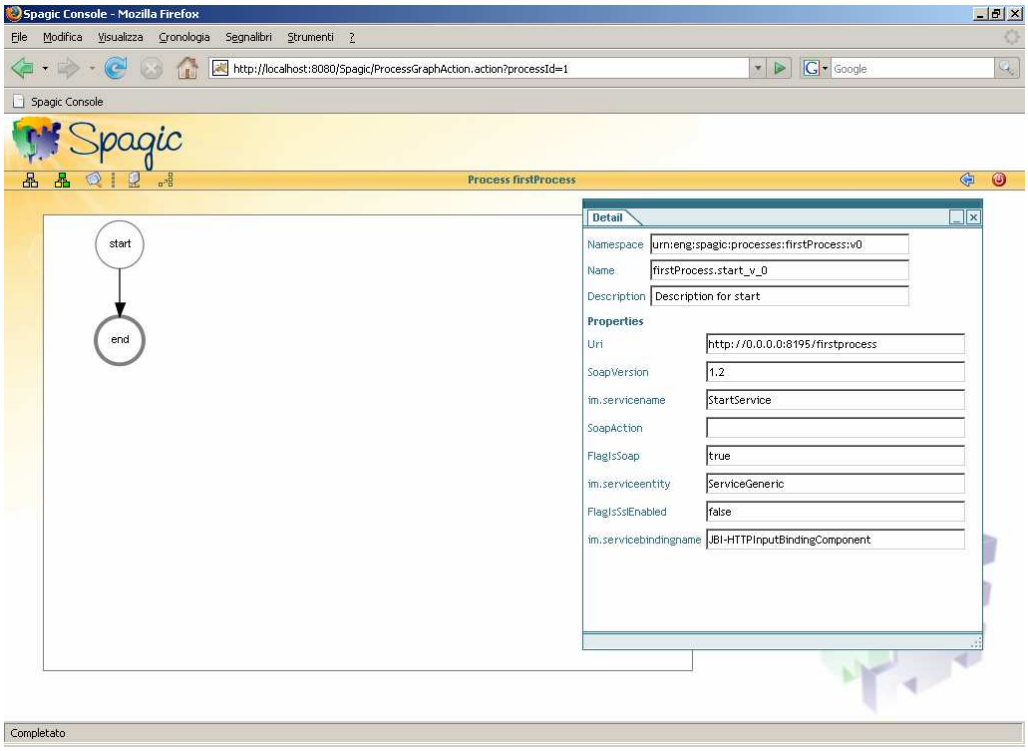


You can perform a search by *Taxonomy* (if you choosed to install jUDDI or freebXML) or by *Iter* using the section in the left side of the page.

8.2.1 Process Graph

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

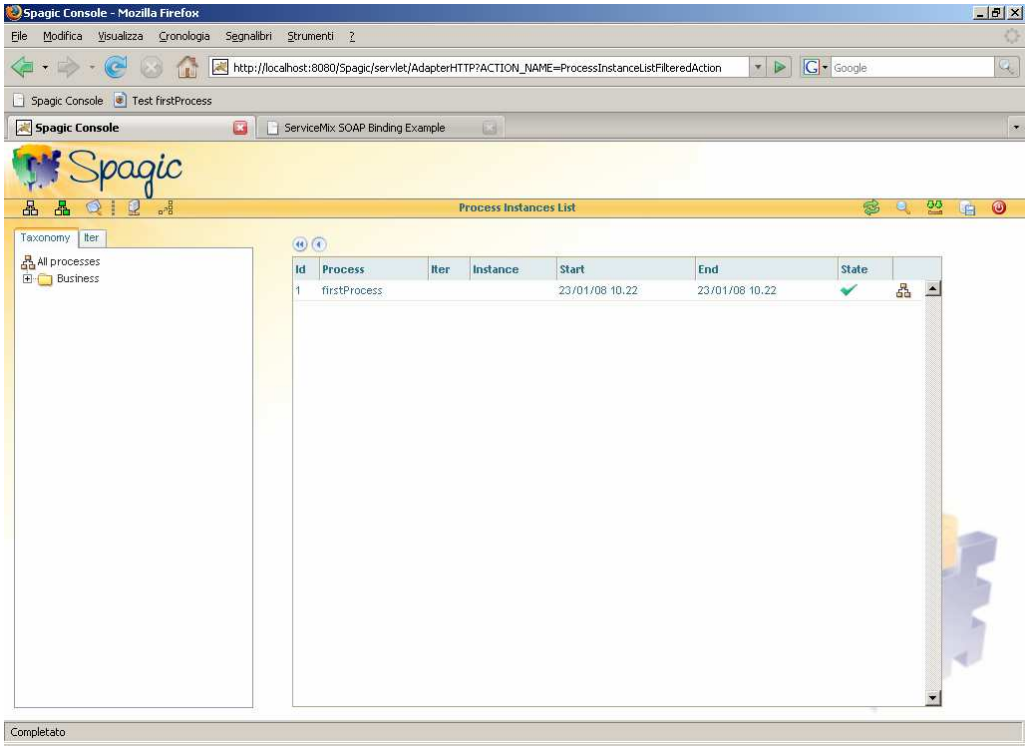
Selecting the elements on the graph it's possible to display a window containing *Name,* step *Description*, *Service* and its *Properties*.



8.3 Process Instances List

Selecting the icon **Process Instances List** () the process instances list is showed. The list contains:

- *Id*: identifier of process instance.
- *Process*: process name.
- *Iter*: name of the iter (optional) associated to the process instance.
- *Instance*: attributes identifying the iter instance (optional).
- *Start*: start execution time.
- *End*: end execution time.
- *State*: instance state. It can be *active* (yellow √), *executed* (green √) or *fault* (red ✕).
-  : visualizes the process execution detail.



It's possible to perform a search by T*axonomy* or by I*ter* using the section in the left side of the page.

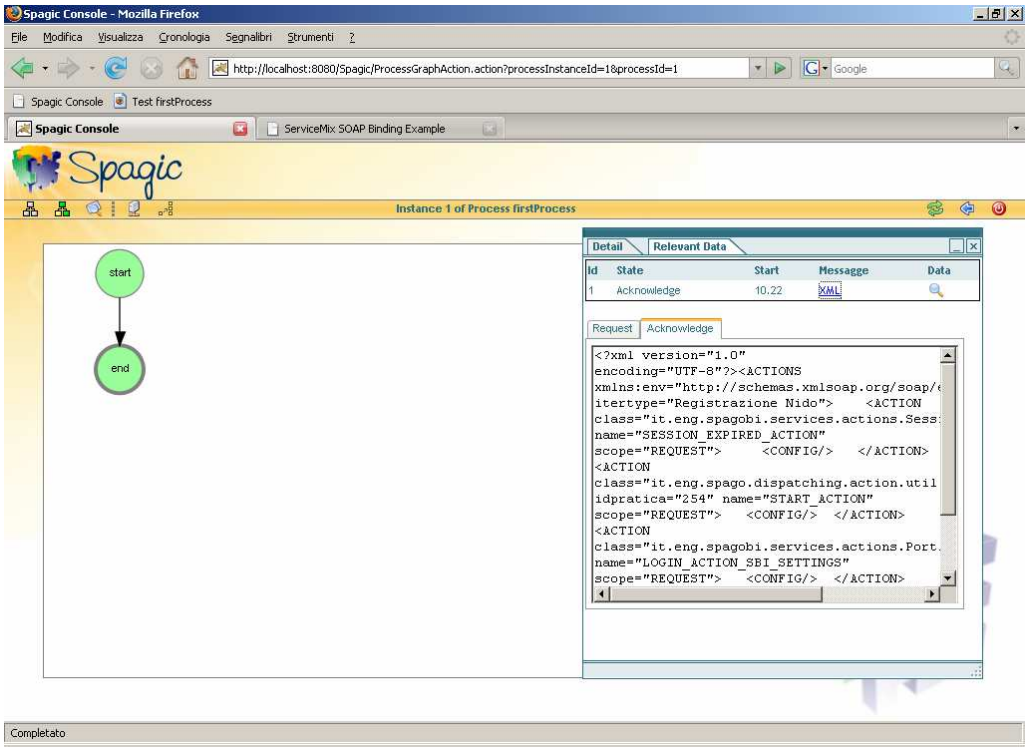
8.3.1 Process Instance Graph

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*.

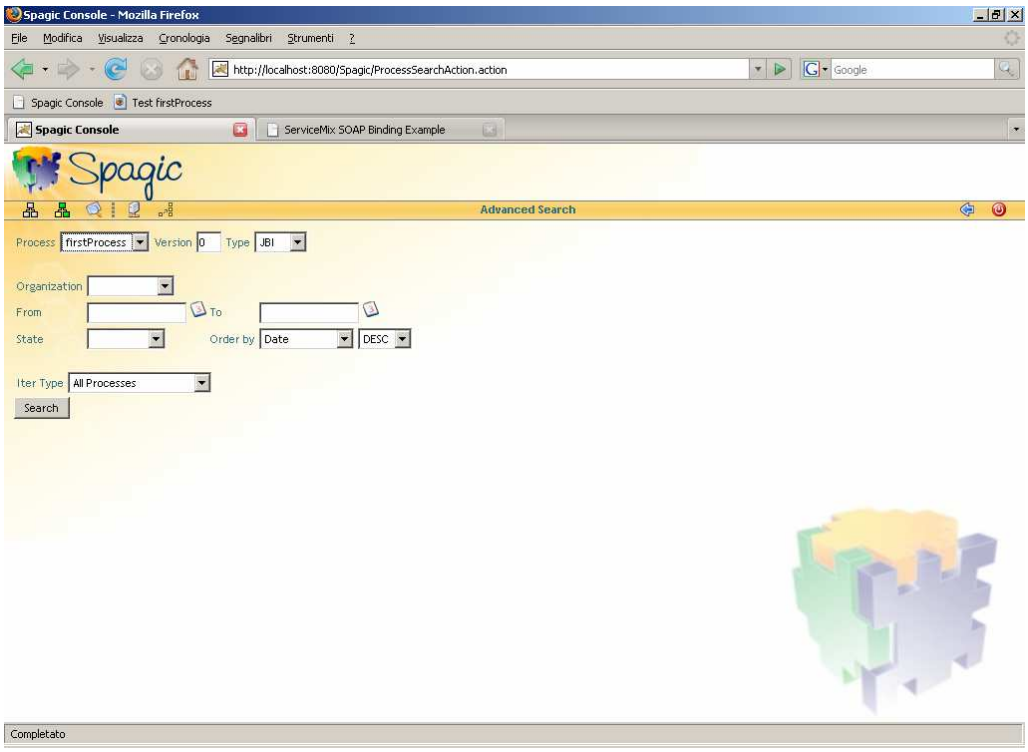


8.3.2 Advanced Search

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*: process name.
- *Version*: process version.
- *Type*: process technology.
- *Organization*: process organization if the process was published on a service registry (jUUDI or freebXML).
- *From* and *To*: start and end validity process dates.
- *State*: it can be *Executing, Error, Executed.*
- *Order by*: order criteria.
- *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 JBI Components.doc*: detail document about ServiceMix and Spagic JBI components.
2. *Spagic Console.doc*: detail document about *Spagic Console* monitoring application.
3. *Spagic Studio User Guide*.doc: detail document about *Spagic Studio* environment.