

Spagic 3 Getting Started

Author: Antonella Miele
Gianfranco Boccalon
Rossato Luca

1	Document Goal	3
2	Versions History	3
3	Requirements	3
4	Introduction.....	4
4.1	Spagic All-in-one	5
4.2	Spagic Standalone	5
4.3	Creation of a new Project.....	5
5	Services/Connectors	8
5.1	Configuration of a new service	8
5.2	Deployment.....	13
5.3	Test	14
6	Processes.....	18
6.1	Creation of a new process	18
6.2	Shortcut Start/End connectors association	22
6.3	Deployment.....	23
6.4	Test	24

1 Document Goal

The goal of this document is to provide you with an introduction on using Spagic3 platform looking at a demo application that should allow you to explore some of the most interesting features of the new platform.

2 Versions History

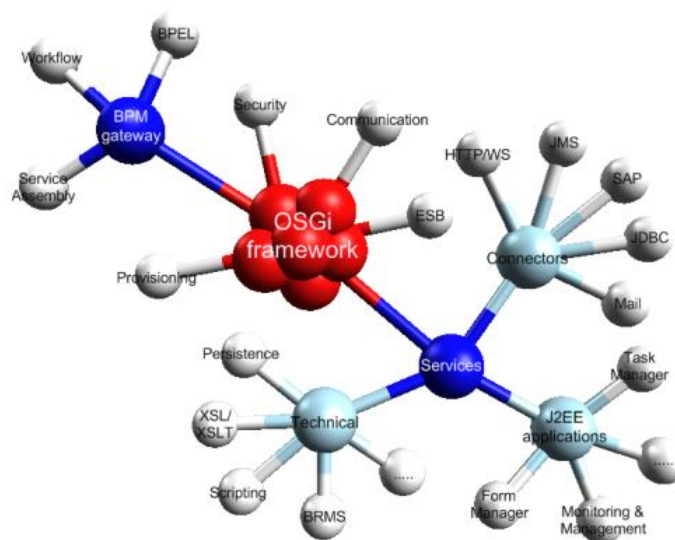
Version/Release n° :	1.0	Date	21/10/2009
Description	First release (English version)		
Version/Release n° :	2.0	Date	10/05/2010
Description	Updates for Spagic 3.0.0		
Version/Release n° :	3.0	Date	28/07/2010
Description	Updates for Spagic 3.1.0		

3 Requirements

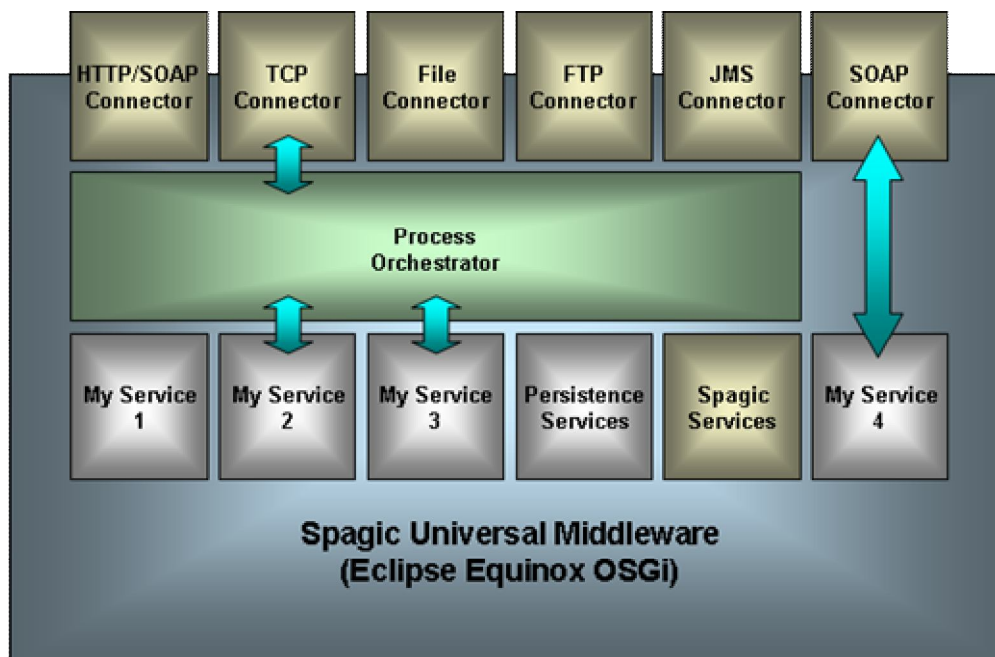
Required Tools	URL for download/Notes	Required for Spagic Studio	Required for Spagic Console
Database	H2 already provided within distribution. Other supported databases: MySQL, Oracle, PostgreSQL	✓	✓
GraphViz	http://www.graphviz.org/	✓	
JDK 1.6	http://java.sun.com/	✓	✓
Apache Tomcat 6.X or later	http://tomcat.apache.org		✓
Mozilla Firefox 2.0.0.x or later	http://www.mozilla.com		✓

4 Introduction

Spagic3 is a major release whose goal is proposing an Enterprise Universal Middleware OSGi based that enables the development of both single services and complex solutions including orchestration processes, workflows with human activities and features like support of rules engines, registries and multi-node distributions.



In Spagic 3 the atomic units of execution are the services that are exposed to the external world through connectors. These services can be deployed and executed alone or composed within complex processes.



Spagic3 Getting Started

In Spagic 3, processes are simply “special services” built by composition of OSGi services that are hosted within an OSGi container (actually Equinox).

The services can simply be configured with a connector (also hosted on the OSGi container) like “My Service 4” in the previous figure, or they can be orchestrated in a complex process by a service called “Process Orchestrator” like “My Service 2” and “My Service 3” in the previous figure. Also this kind of complex processes can be configured with a connector, for the activation of the process.

In this guided tour you will use the following tools:

- Spagic 3 studio
- Spagic 3 console
- Service Manager

Note for Spagic 2 users



The main change in Spagic 3 is the shift of focus from processes to services as atomic unit of execution. Indeed while in Spagic 2 only processes can be deployed and executed, in Spagic 3 deployment and execution are actions that involve also raw services, without necessarily being linked to a process, but just to a connector. In this way services can be used alone or linked to a process, and each service could be reused in different execution models.

4.1 Spagic All-in-one

The simplest way to make the first tour with Spagic is using the package “Spagic 3 All-in-one” that consist of a Web application that includes the Spagic Console and the Spagic Service Manager.

Please refer to the document “*Spagic 3 Download Guide*” for the details of the tools to download.

First of all you need to install the H2 driver within Tomcat (or your preferred servlet container), and then install the Web application *Spagic.war*. Once you will start your Tomcat it will automatically start Spagic Console, Spagic Service Manager and also H2 database creating a Spagic Database instance within it.

4.2 Spagic Standalone

If you want an alternative approach to Spagic All-in-one, you can use “Spagic 3 Standalone” package that consists of several separated applications, it contains the Spagic Service Manager (a standalone Equinox server), the H2 database and the Spagic Console to install into Tomcat (or another servlet container).

Differently from Spagic All-in-one, with the Standalone mode you’ll have to start each Spagic application by your own manually. Also in this mode, remember to install the H2 driver in Tomcat before starting it.

In order to start up the service manager open the command window and execute the *service-manager.bat* (or *.sh*) that is under the Service Manager directory (from now on we will call it *SERVICE_MANAGER*).

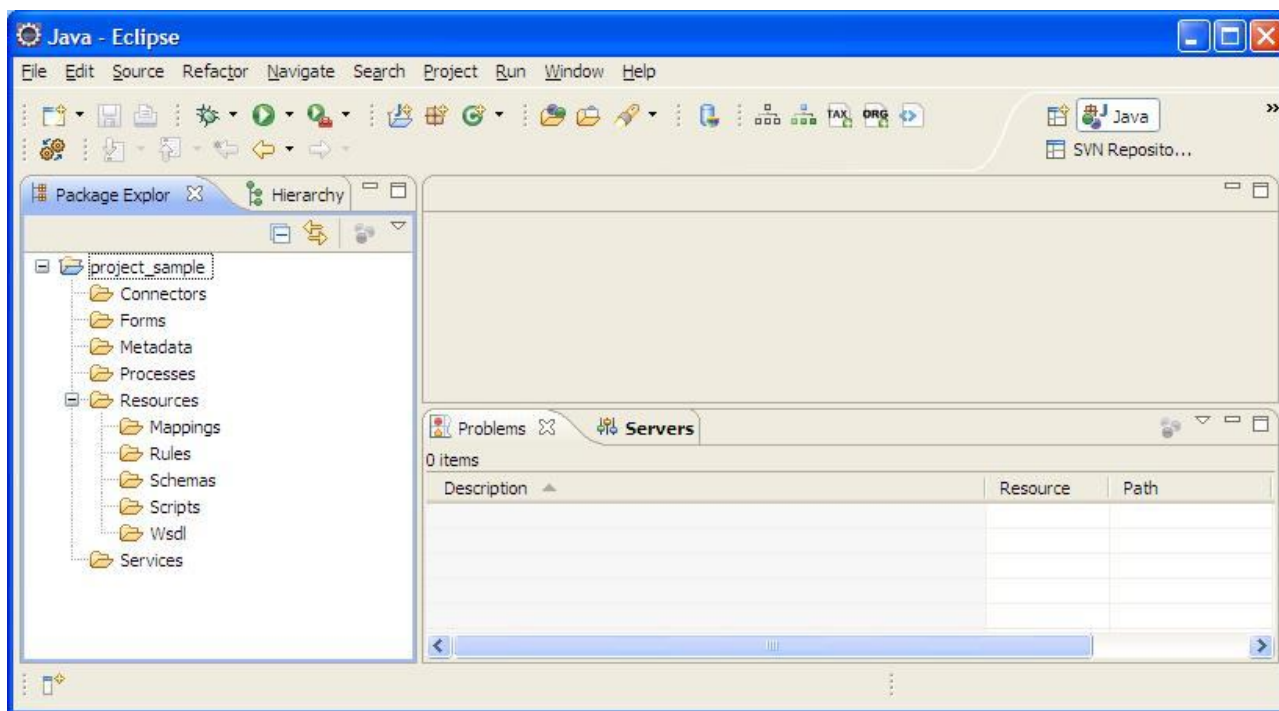
4.3 Creation of a new Project

In this document we’ll use the notation: *WORKSPACE_DIR* as the directory of the Eclipse workspace that you will create with Spagic Studio.

Spagic3 Getting Started

Once created the workspace after the start of Spagic Studio in order to create a new Project, select *File->New->Project...*, choose *Spagic3 Project* under the category *Spagic* and click on *Next*. A Wizard is opened where you can define the name of the project you want to create.

Write "*project_sample*" as the project name, and click on the Finish button.

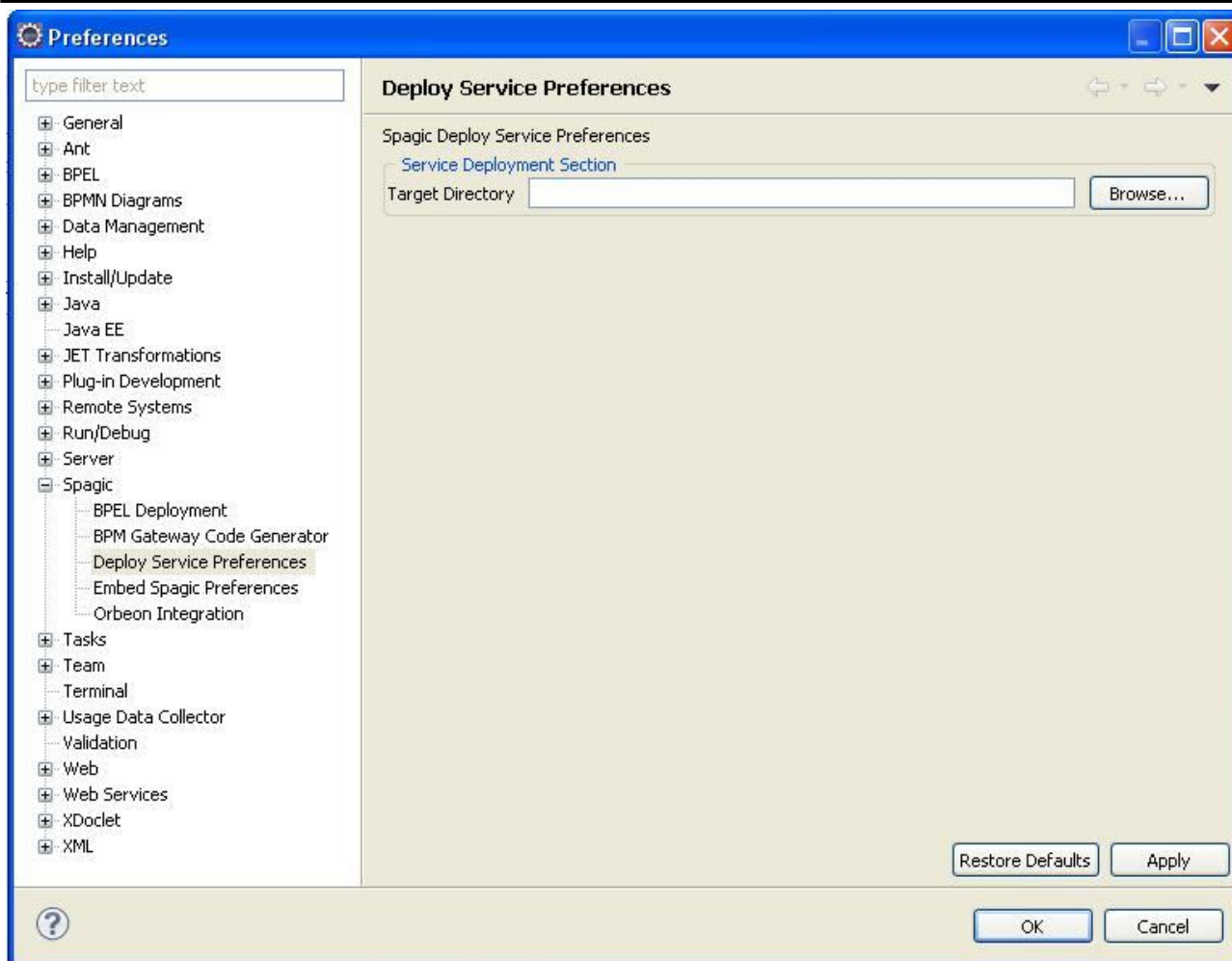


Within the project main folder you will find different subfolders where you can place the components of your project: connectors, processes, services, forms, metadata and resources, further divided in other several subfolders.

Before to start the design of your project components you need to setup the *Target Directory* inside your Spagic Studio. The Target Directory is a folder where Spagic Deploy Manager will copy all files necessary for the execution of Spagic processes and services. By default the Target Directory is set to "C:\Scrappy", if you want to change it go in Windows-> *Preferences-> Spagic->Deploy Service Preferences* and change the value of the *Target Directory* field.

If you change the Target Directory value in Spagic Studio you need to do it also for Spagic Service Manager, if you're using the Spagic All-in-one package you must open the Spagic.war file, go into WEB-INF folder and edit the file web.xml. Seek the param-name spagic.home and change the param-value with the new Target Directory you choosed. If you are using the Spagic Standalone you simply need to change the value of -Dspagic.home parameter of Spagic3.ini file inside the *SERVICE_MANAGER* directory.

Spagic3 Getting Started



5 Services/Connectors

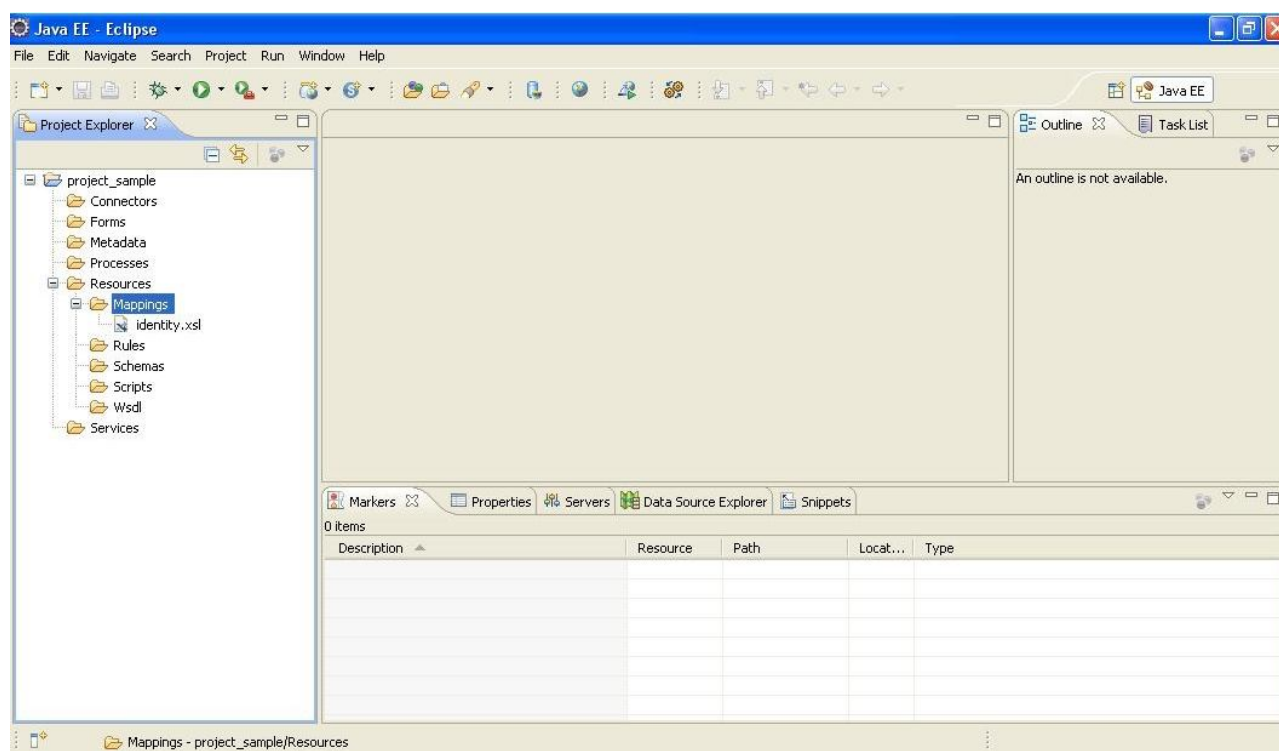
5.1 Configuration of a new service

In the folder *Services* we can place all the services definitions we need.

In our example we will configure a service that performs a XSLT transformation. For simplicity we provide all the resources used in this demo, within the folder *sample_resources* in the “Spagic3 All-in-one” package.

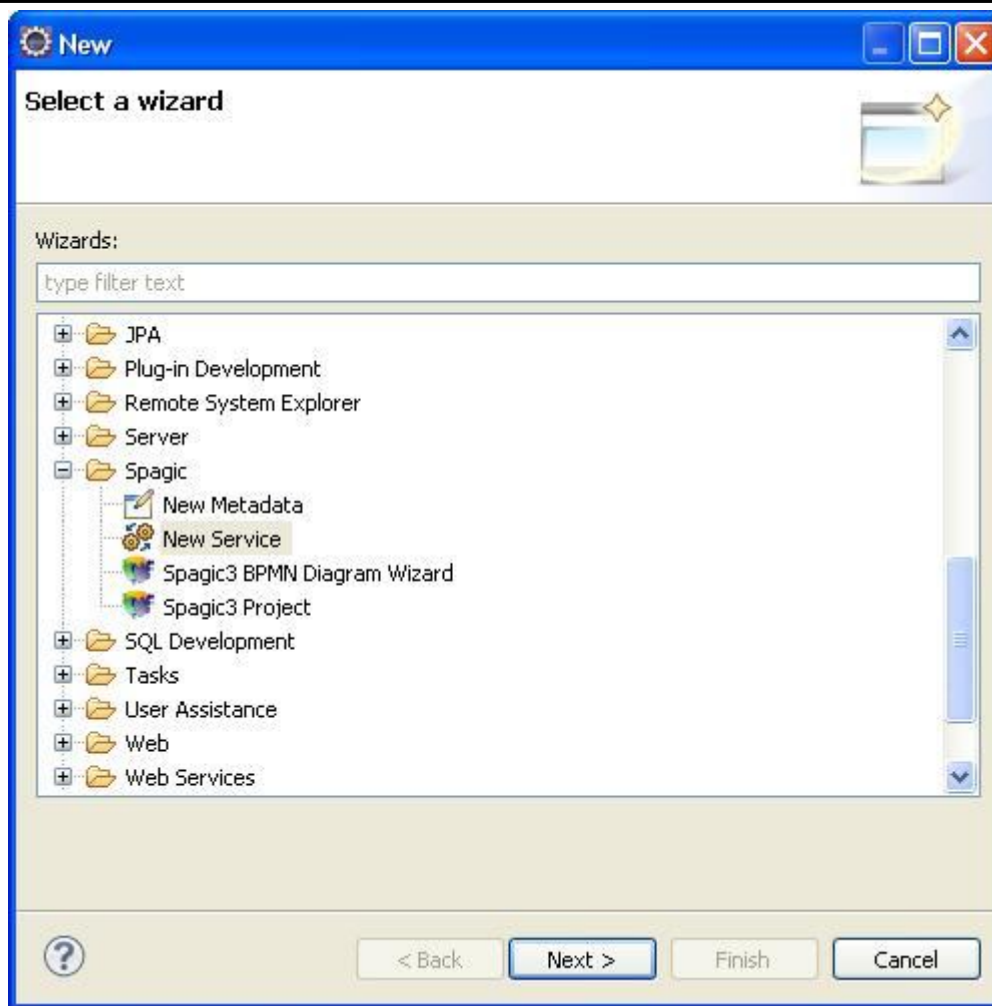
First, we need the resources necessary to configure the services we are creating: in our sample the resource used by the service is an xsl file. So we place this file in the folder *Mappings*, under the *Resources* folder.

Placing all the resources used by services in project under the *Resources* folder before configuring the service, is more comfortable for the feature of drag and drop of files, available in several service configuration in Spagic 3.

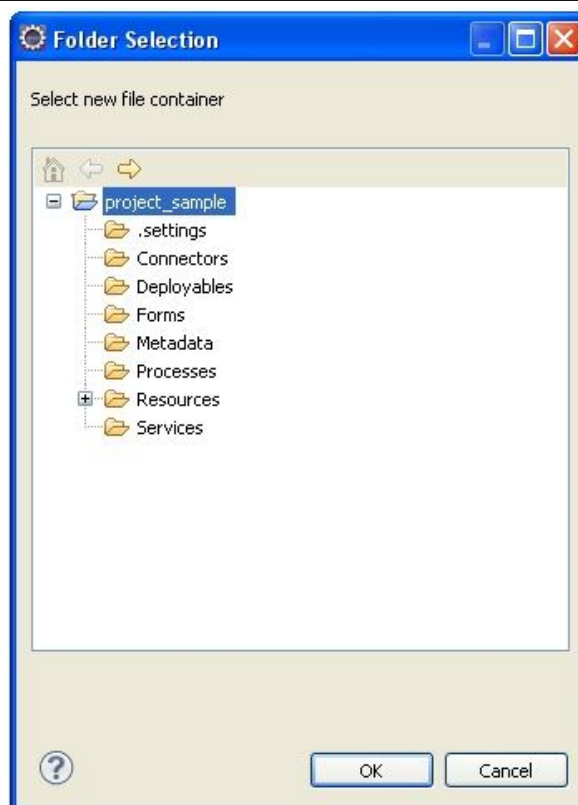


After this we can configure the service: let's go under the project and choose File → New → Other.

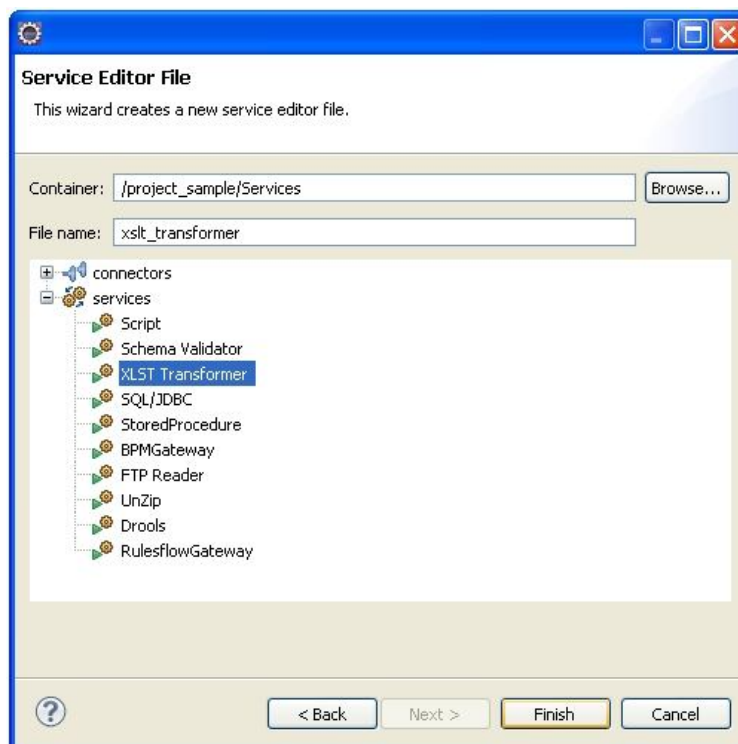
The list of wizards appears. Let's go under the Spagic → New Service, and click *Next*.



In the next window there is the list of all service and connectors available. In the text box "Container" you can specify the placement of the new service, typically under the Services folder of your project. If is not already present, click the Browse button. The wizard of folder selection appears: click on the folder of your project, in our case the *project_sample* project and select the subfolder Services and click OK.



In the next Wizard go to *Service* and click on it, select *XSLT Transformer*, and finally set the name of the service in the "File name" field. For this example call it *xslt_transformer* and then Click *Finish*.

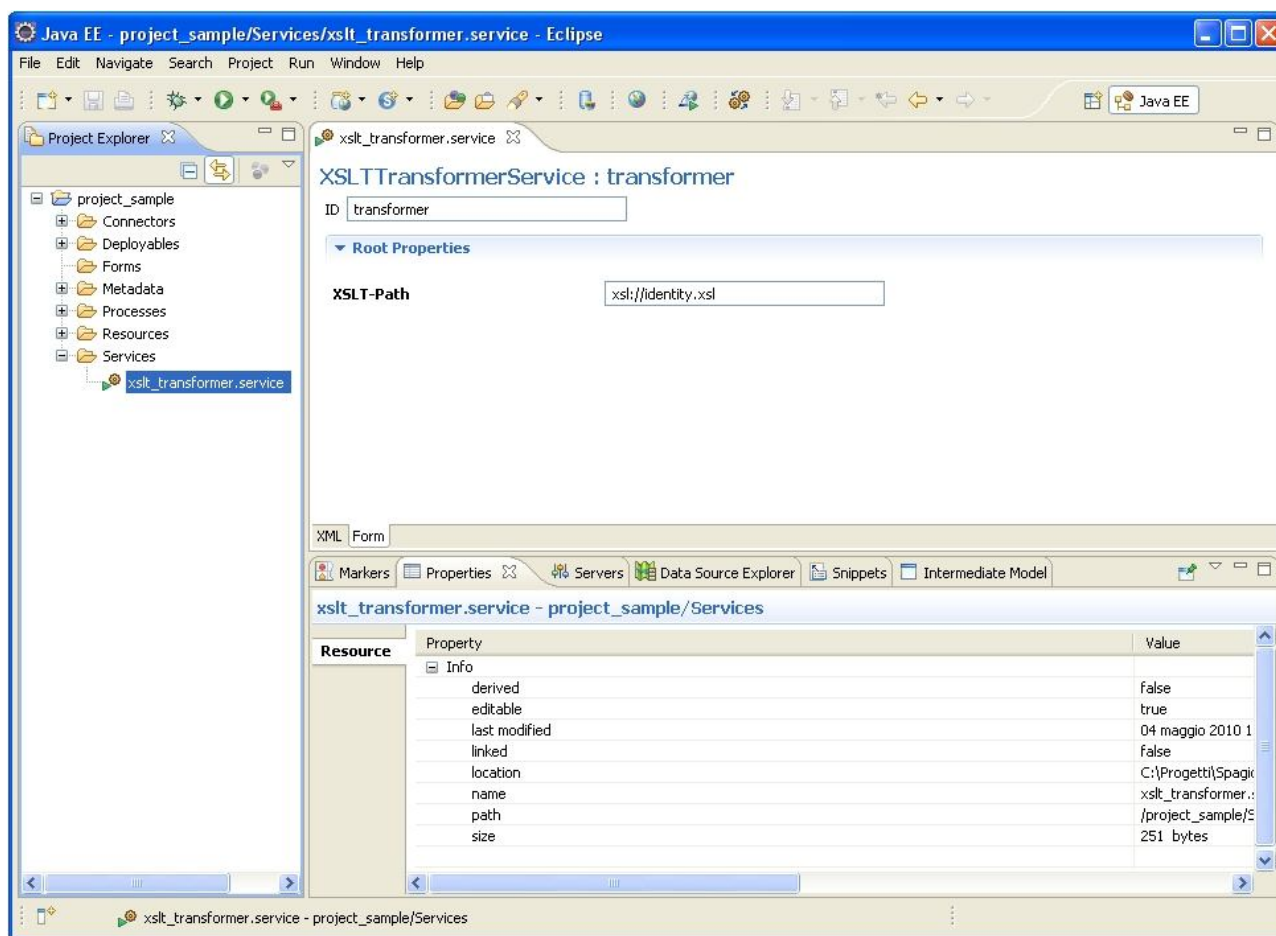


Spagic3 Getting Started

Now a new instance of service is created in the *Service* folder whose name is *xslt_transformer.service*. Automatically, the wizard service configuration appears and you can set all the values of the properties of a service, filling the fields in the configuration wizard.

All services have a unique identifier that allows us to reuse them; this identifier is briefly named **ID**.

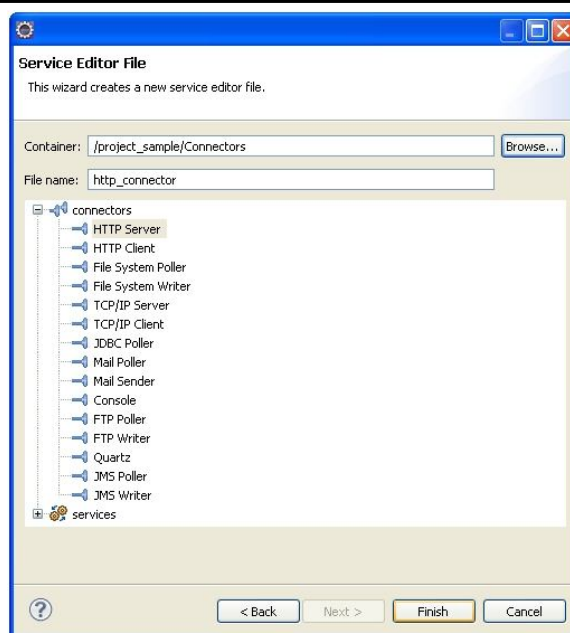
In our case, we have to set an ID for the service and give it an xslt-Path. Set the ID filling the namesake field with “*transformer*” and set the xsl file as path making the drag and drop of the file *identity.xsl* we previously put in the *Mappings* folder, in the field *XSLT-Path*. In this way, our service will be configured.



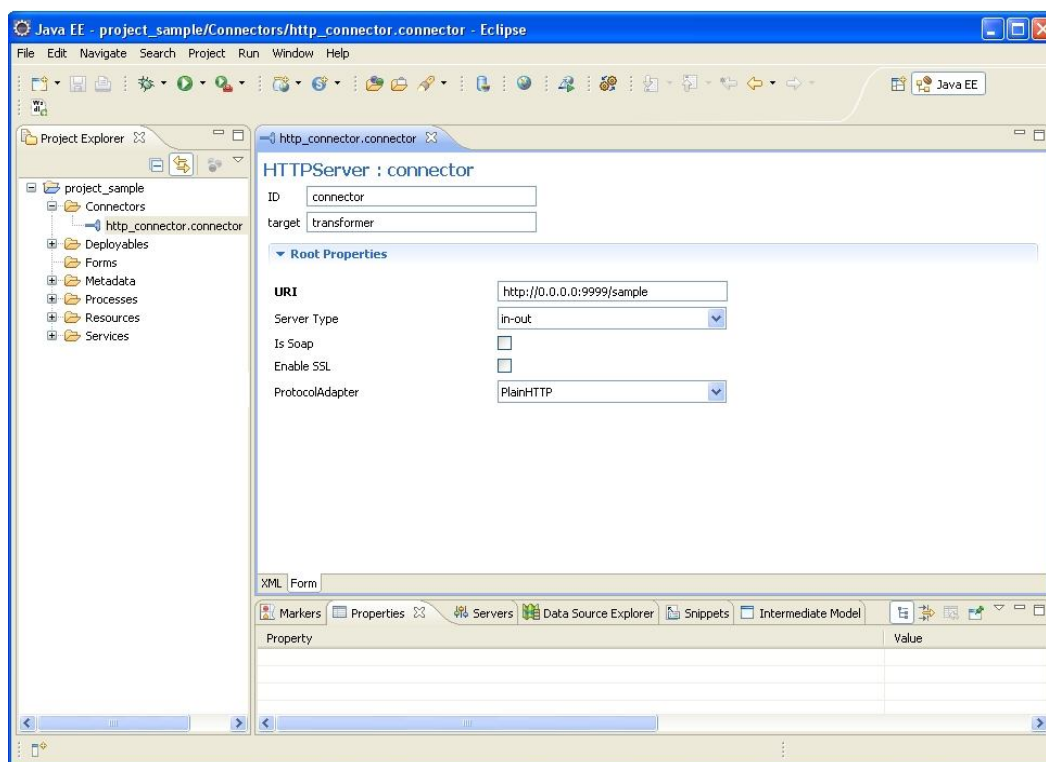
To expose the service and allow it to be called from external applications, we have to link it to a connector. So, let's create a new instance of connector. In our example our connector is an HTTP Server. To do this, right click on *File* → *New* → *Other* and select *Spagic* → *New Service*.

In the wizard set the placement of the connector under the directory *WORKSPACE_DIR/project_sample/Connectors*. If it is not already present, click on the Browse button and set it manually, in the folder selection window. Then go in the list and click on *Connectors*, in the list below select the connector *HTTP Server*. Finally set the name of the connector writing “*http_connector*” in the box name and click *Finish*.

Spagic3 Getting Started



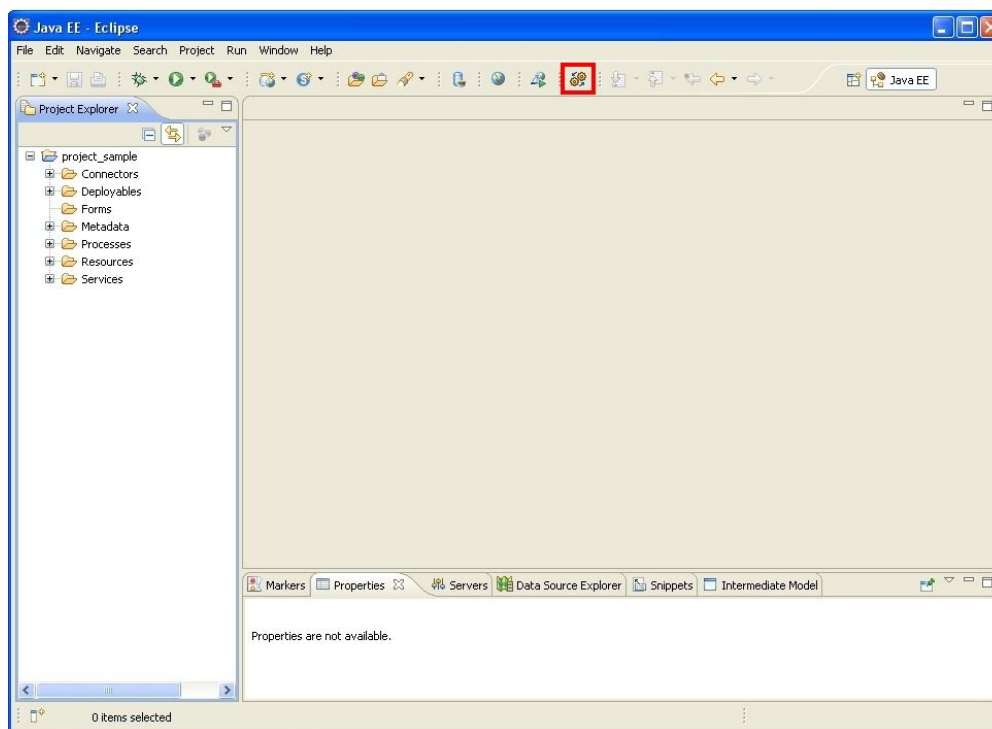
So a new instance of connector is created in the *Connector* folder whose name is *http_connector.connector*. Automatically, the wizard connector configuration appears, where you can set all the values of the properties of a connector, filling the fields in the configuration wizard. Set the ID filling the namesake field with “connector” and set the URI file with “http://0.0.0.0:9999/sample” and select in the field Server Type “in-out”. To link the service to the connector, just drag and drop of the *xslt_transformer* service we have previously configured and place it in the *target* field. The ID of the service will appear in the target field. In this way, our connector will be configured.



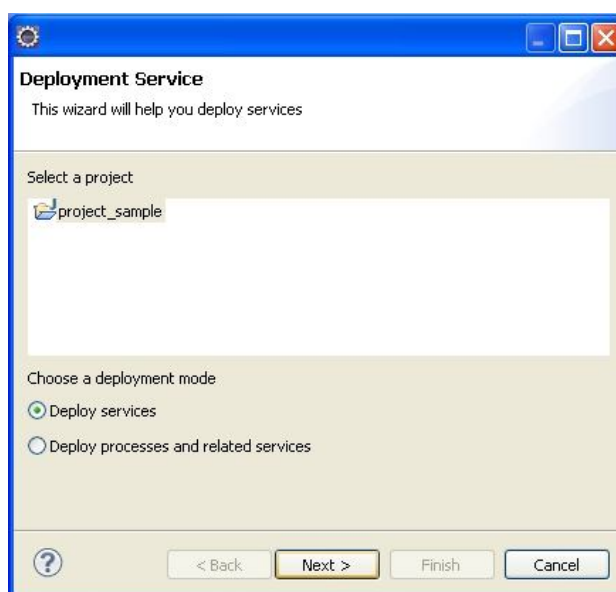
5.2 Deployment

Before to deploy your service you need to start up the service manager. If you are using the All-in-One distribution the service manager is already running inside Tomcat while if you use The Standalone mode and you need to start it manually.

To deploy a service or a process, we have to call the *Deployment Wizard* and choose the services and processes you need to deploy. Click in the icon of deployment service that is in the main icon bar of Spagic Studio 3.



In the wizard, there are all workspace projects. Select the project *project_sample* and choose “Deploy services”.



Spagic3 Getting Started

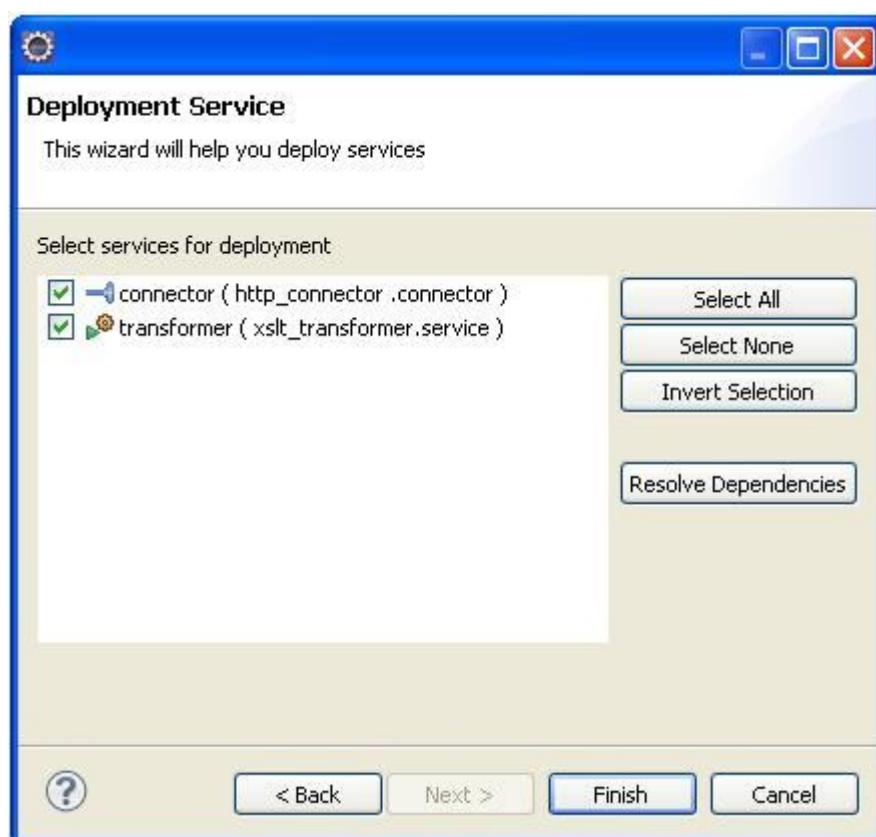
In the next window check the connector of the service you want to deploy.

To deploy the services (and the related resources) you can proceed in two ways:

- Manually, selecting all the services you want to deploy.
- Selecting the first connector of the flow of control (in our case the HTTP Server) and click the *Resolve Dependencies* button. The service or process linked will be automatically checked.

When all the items you want to deploy are selected, click *Finish*. The services will be published in the *Target Directory* folder.

The Deployment wizard is a tool that copies the resources and the services definition (.connector and .service files) in the Server target directory: it's useful because, for example, the resources must be copied before the services definition, and this order is automatically managed by the wizard.

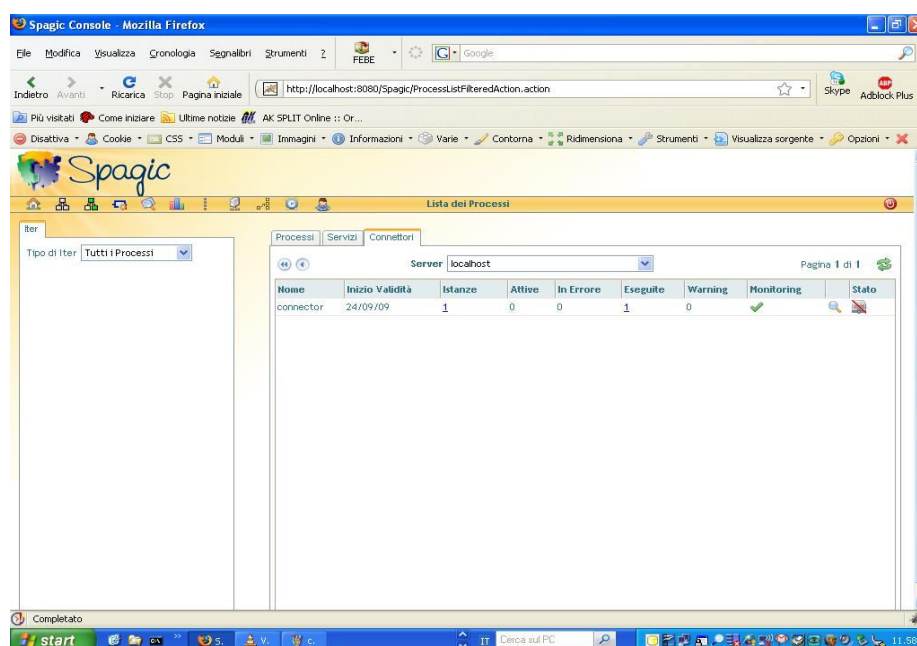
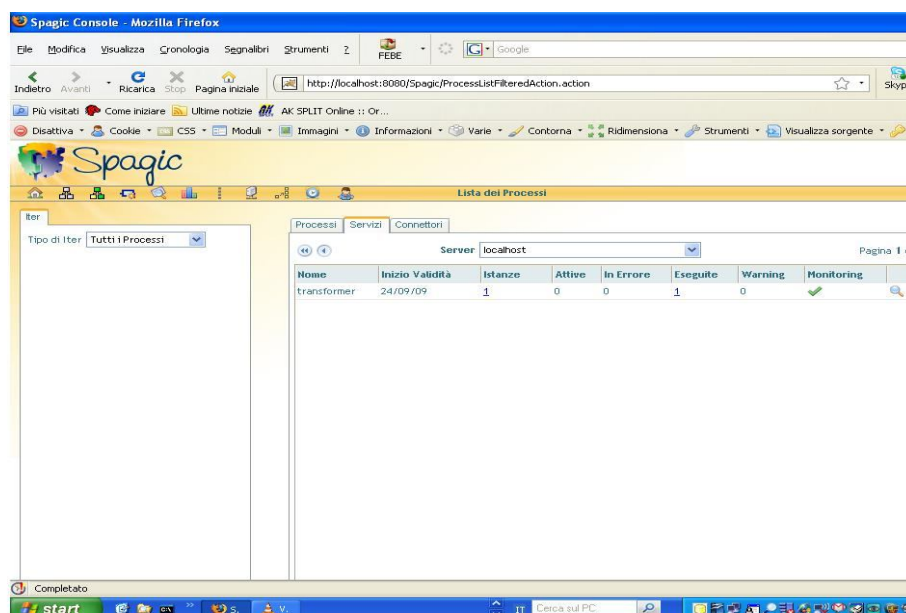


5.3 Test

After deploying services and connectors we can see all items in the Spagic console. For installation and use of Spagic Console, please refer to the document "*Spagic3 Console.pdf*". In Spagic Console, clicking on the icon Process/Services List we have the list of all the processes, services and components deployed on the service manager.

Spagic3 Getting Started

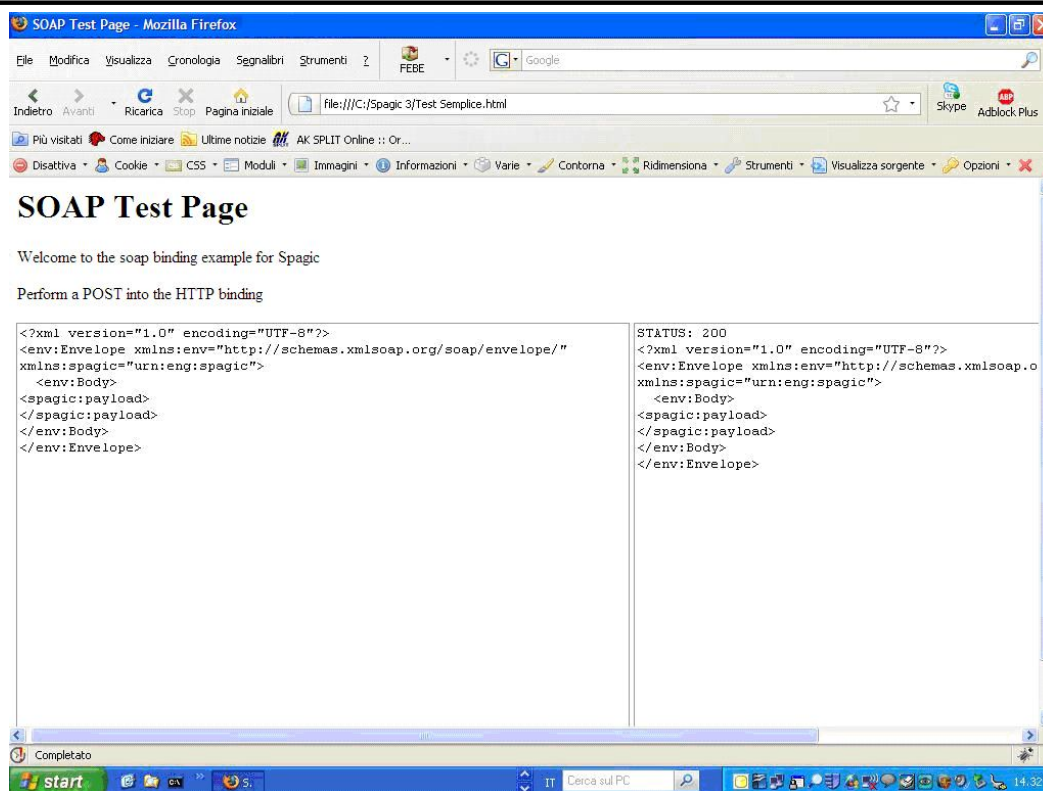
Clicking on the tab *Services*, you can see the service *transformer* we have deployed, and in the tab *Connectors* you can see the connector of the service deployed.




Then, let's execute the service we've deployed. Take the file "*Test Page.html*" that is in the directory *sample_resources/html* and open it.

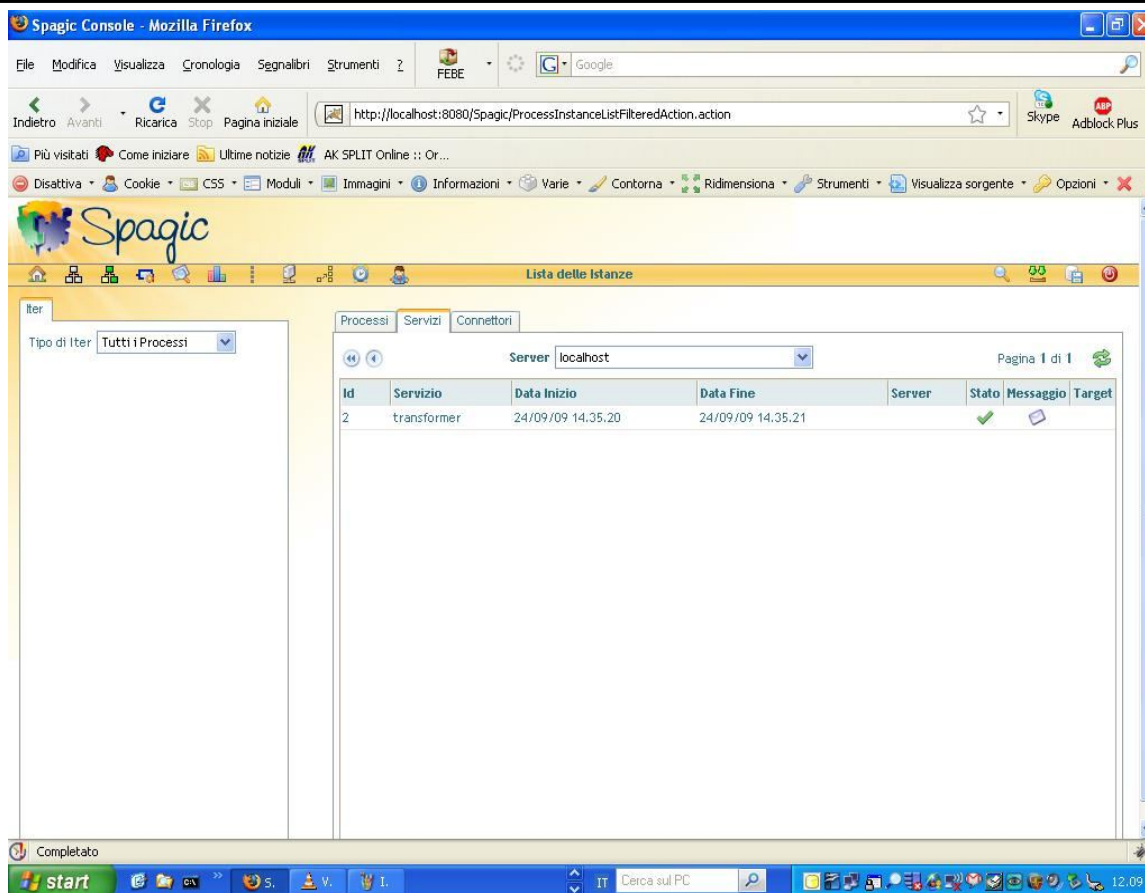
This page allows us to send some XML to an HTTP endpoint, and to see the result. Click on send, and in the window on the right will appear the xml text transformed by the xml transformation.

Spagic3 Getting Started



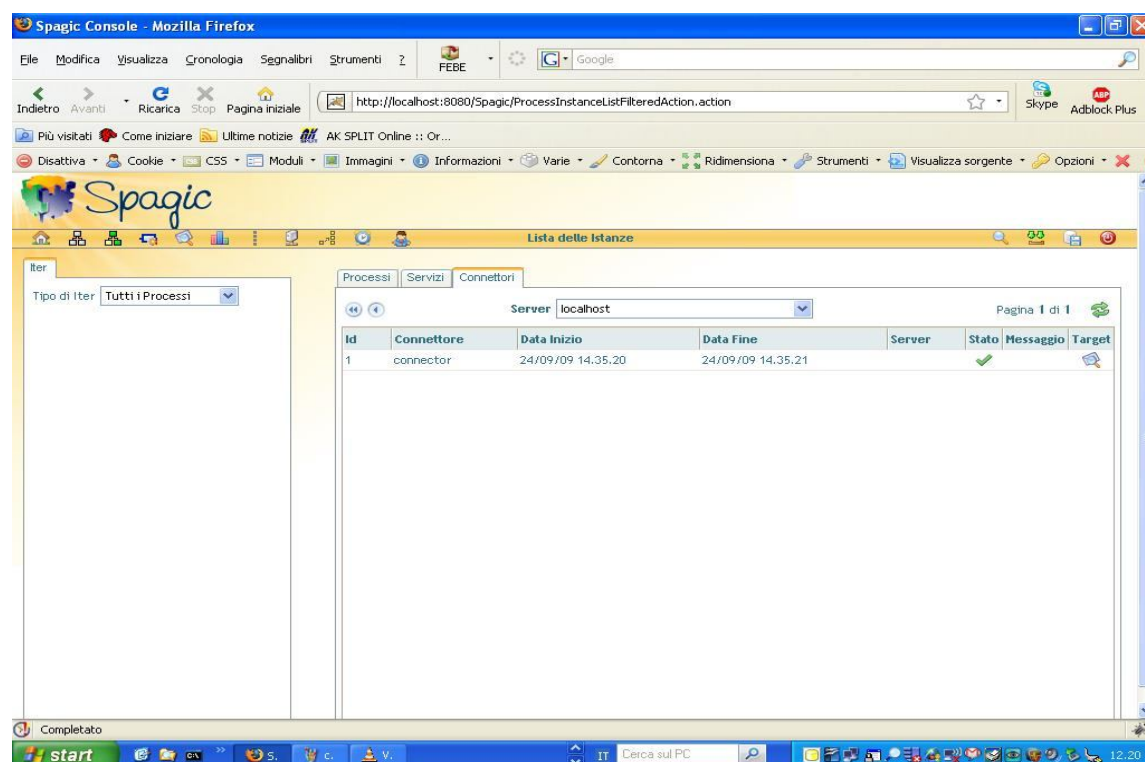
If you take a look in the Instances List () you can find the instances of service and the connector we have deployed.

Spagic3 Getting Started



The screenshot shows the Spagic Console interface in a Mozilla Firefox browser window. The address bar displays the URL `http://localhost:8080/Spagic/ProcessInstanceListFilteredAction.action`. The page title is "Spagic" and the main heading is "Lista delle Istanze". On the left, there is a sidebar with a "Tipo di Iter" dropdown set to "Tutti i Processi". The main content area has tabs for "Processi", "Servizi", and "Connettori". The "Processi" tab is active, showing a table of instances for the "transformer" service on the "localhost" server. The table has columns: Id, Servizio, Data Inizio, Data Fine, Server, Stato, Messaggio, and Target. One instance is listed with Id 2, starting at 24/09/09 14:35:20 and ending at 24/09/09 14:35:21, with a status of "✓".

Id	Servizio	Data Inizio	Data Fine	Server	Stato	Messaggio	Target
2	transformer	24/09/09 14:35:20	24/09/09 14:35:21	localhost	✓		



The screenshot shows the Spagic Console interface in a Mozilla Firefox browser window. The address bar displays the URL `http://localhost:8080/Spagic/ProcessInstanceListFilteredAction.action`. The page title is "Spagic" and the main heading is "Lista delle Istanze". On the left, there is a sidebar with a "Tipo di Iter" dropdown set to "Tutti i Processi". The main content area has tabs for "Processi", "Servizi", and "Connettori". The "Connettori" tab is active, showing a table of instances for the "connector" service on the "localhost" server. The table has columns: Id, Connettore, Data Inizio, Data Fine, Server, Stato, Messaggio, and Target. One instance is listed with Id 1, starting at 24/09/09 14:35:20 and ending at 24/09/09 14:35:21, with a status of "✓".

Id	Connettore	Data Inizio	Data Fine	Server	Stato	Messaggio	Target
1	connector	24/09/09 14:35:20	24/09/09 14:35:21	localhost	✓		

6 Processes

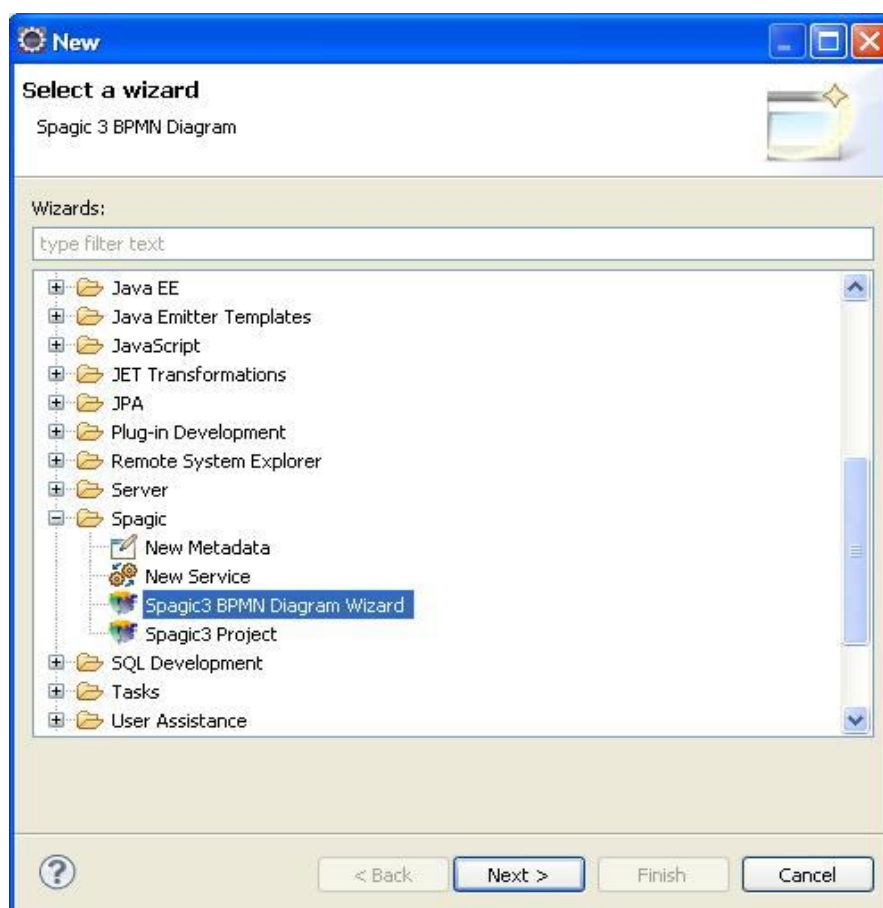
6.1 Creation of a new process

The steps for creating a new process in Spagic 3 are:

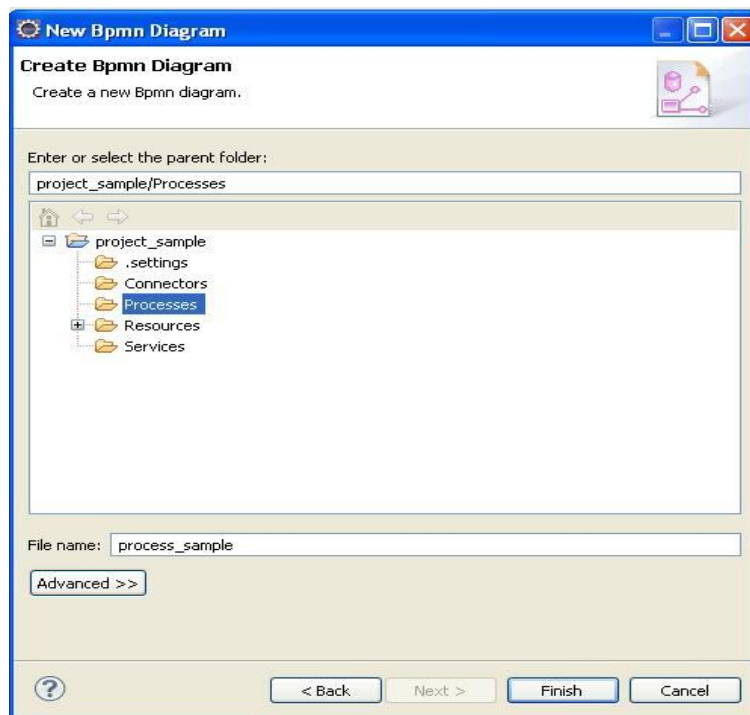
- Create a new BPMN process with Spagic Studio
- Configure all the services that will be used by the process steps. Each service is configured in a declarative way.
- Define the connectors that will be used to activate the process. Also this step is done in a declarative way.

Before to start the process design we need to configure Graphviz in order to generate processes graphs. Go to Windows->Preferences->Spagic and set Graphviz Dot Program Location(the path to graphviz dot program) and Graphviz Temp folder (the path for graphviz temporary folder, it must be created manually).

Now to create a new Process, do right click on *File* → *New* → *Spagic* → *Spagic3 Bpmn Diagram Wizard*.



Set the name and location of the process you want to create. Give the name “*process_sample*” and place it under the folder *Processes*.

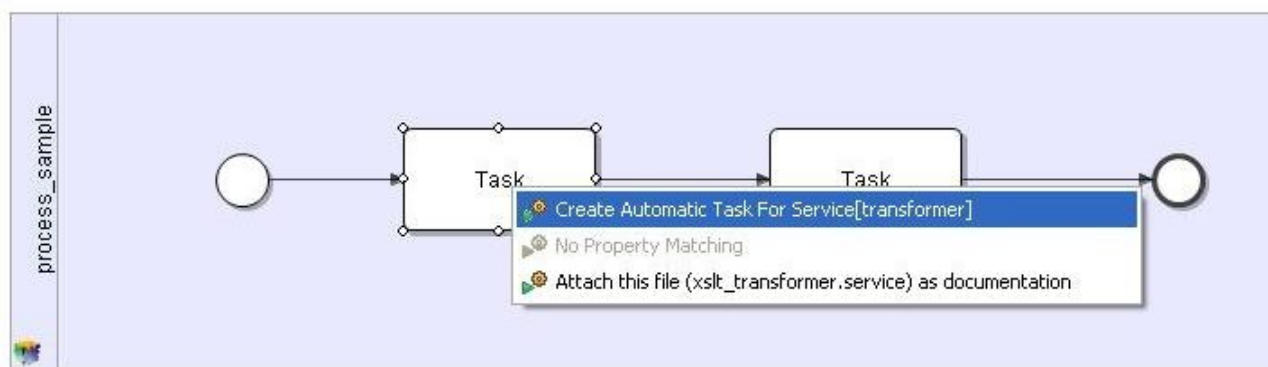


Let's model a process: from the Palette on the right select the *Empty Start*, an *Empty End* and a *Task*. Replicate the item Task twice. Then link them with the Flow Connector.

The first task is choosing the process technology: select the Pool properties and in the tab *Technology* choose *JBPM runtime*.

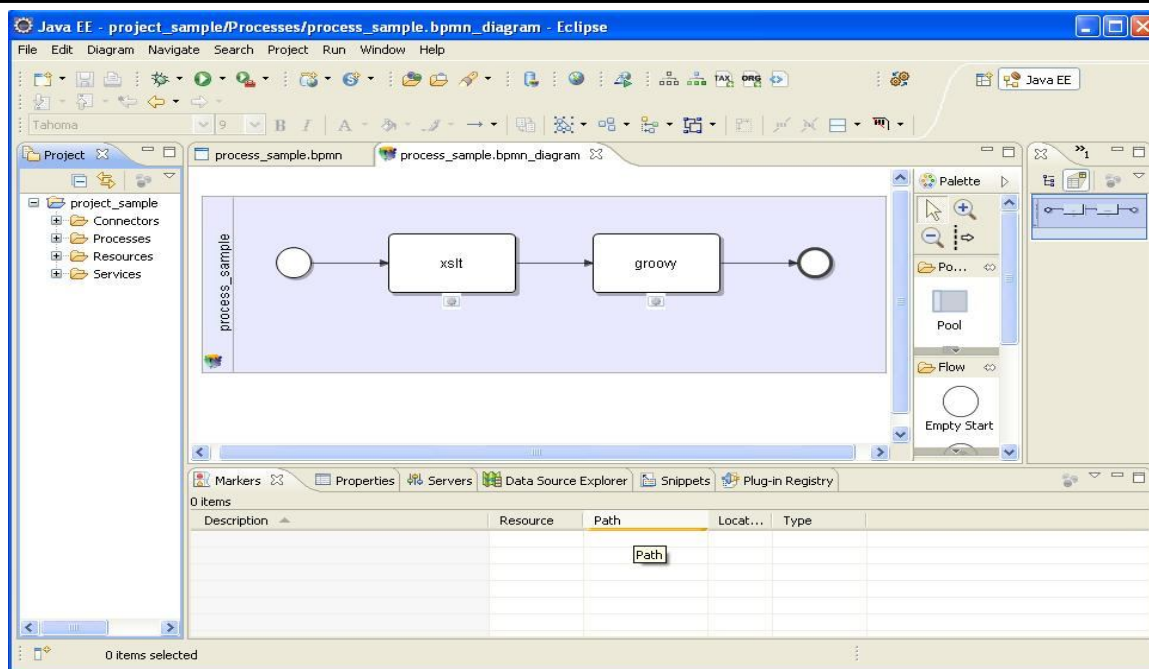
To attach the services to the steps of the process you just have to drag and drop the services to the steps of the process, and choose "Create Automatic Task for Service". The service will be attached to the task.

For this example, we can drag and drop the service *xslt_transformer* to the task of the process we are drawing.



Finally the process is complete.

Spagic3 Getting Started



The BPMN Diagram Wizard created automatically also the service needed in order to execute our process, it is under Services folder and it's called `process_sampleOrchestration.service`.

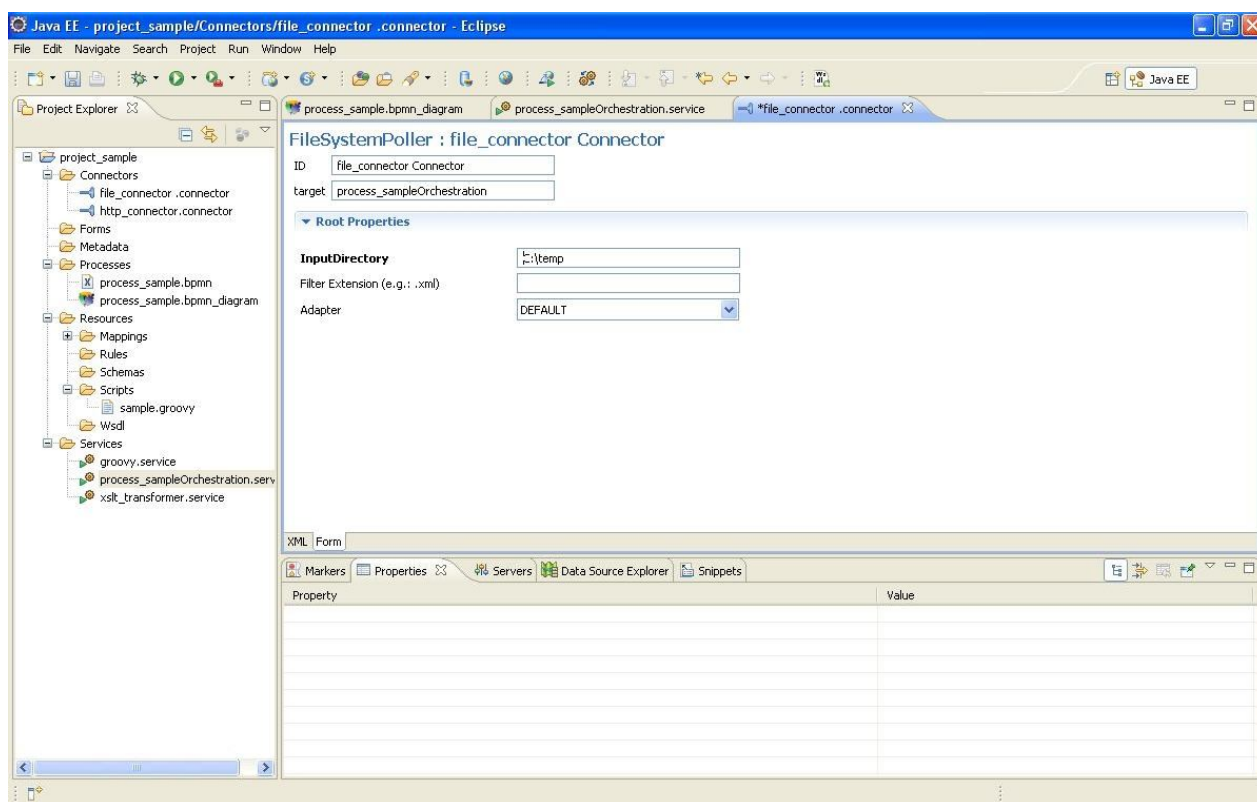
Then we have to create two connectors that can activate the process, and write its result somewhere.

Let's create the input connector, in our case a file poller. To do this, right click on **File** → **New** → **Spagic** → **New Service**. In the wizard set the placement of the connector under the directory `WORKSPACE_DIR/project_sample/Connectors`. Then go in the list and select **Connectors** → **File System Poller**. Finally set the name of the connector writing "`file_connector`" in the box name and click **Finish**.



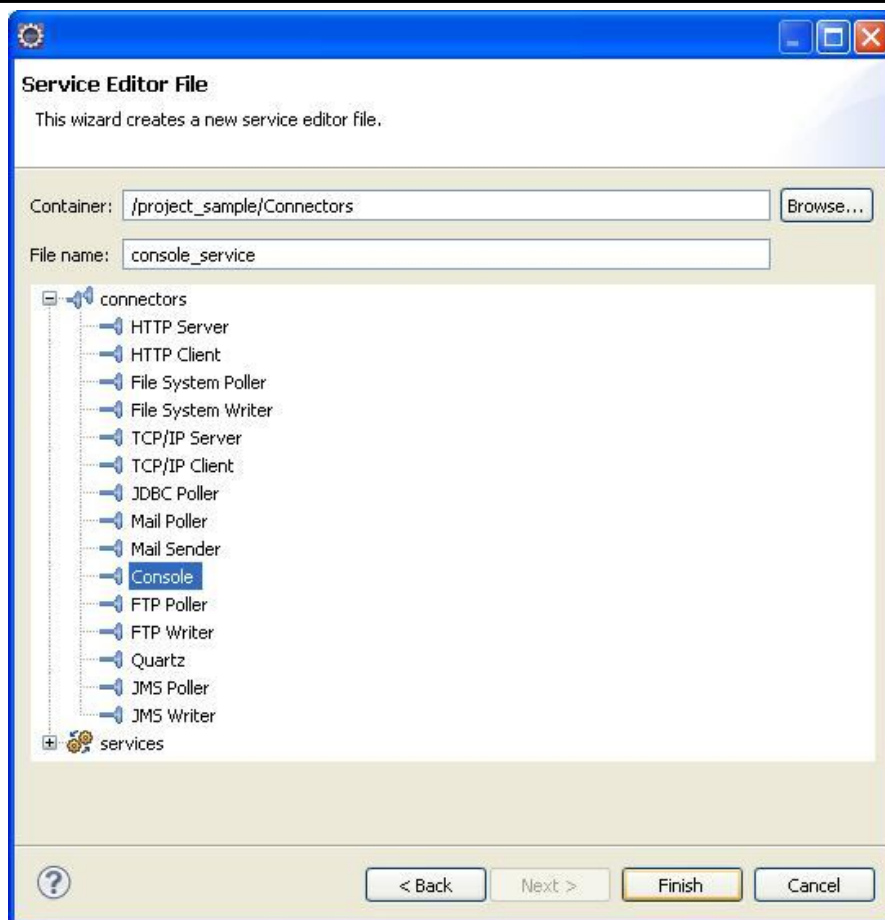
Spagic3 Getting Started

In the Service editor of the *file_connector* we have to define the ID, the directory to check in, and the target. In target we have to drag the definition (the .service) of the process *process_sampleOrchestration*. Leave the default ADAPTER



Then we have to create the output connector, in our case a Console connector. To do this, right click on *File* → *New* → *Spagic* → *New Service*.

In the wizard set the placement of the connector under the directory *WORKSPACE_DIR/project_sample/Connectors*. Then go in the list and click on *Connectors*, in the list below select the connector *Console*. Finally set the name of the connector writing "*console_service*" in the box name and click Finish.

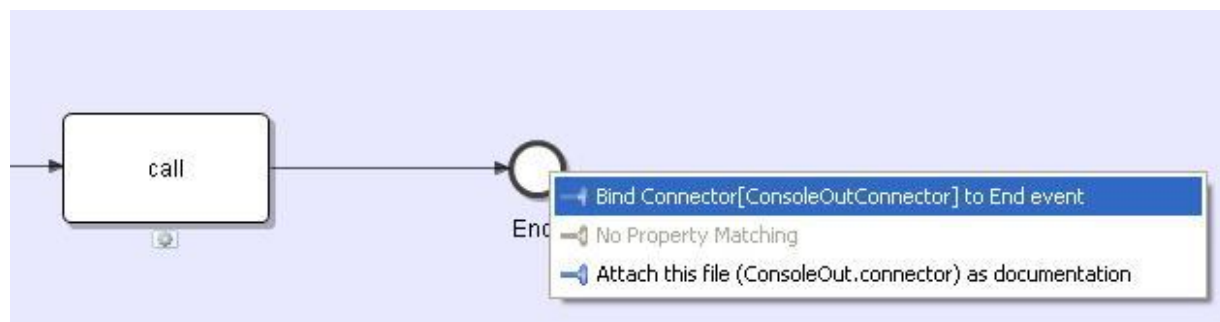


This connector has no parameters, except the ID. Then we have to drag and drop the service connector to the field target of the service process, to make it the output connector of our process.

6.2 Shortcut Start/End connectors association

Once you get more familiar with Spagic target-definition mechanisms, you will be happy to discover that there is a fastest way to associate process Start and End connectors than explained in the previous paragraph.


Once created the start or end connector you can associate it to the process simply by drag and drop it on the process step. Once released it a popup menu will display, you simply need to click on "Bind Connector[connectorID] to Start/End event". In the image below an example of an end connector association is shown.



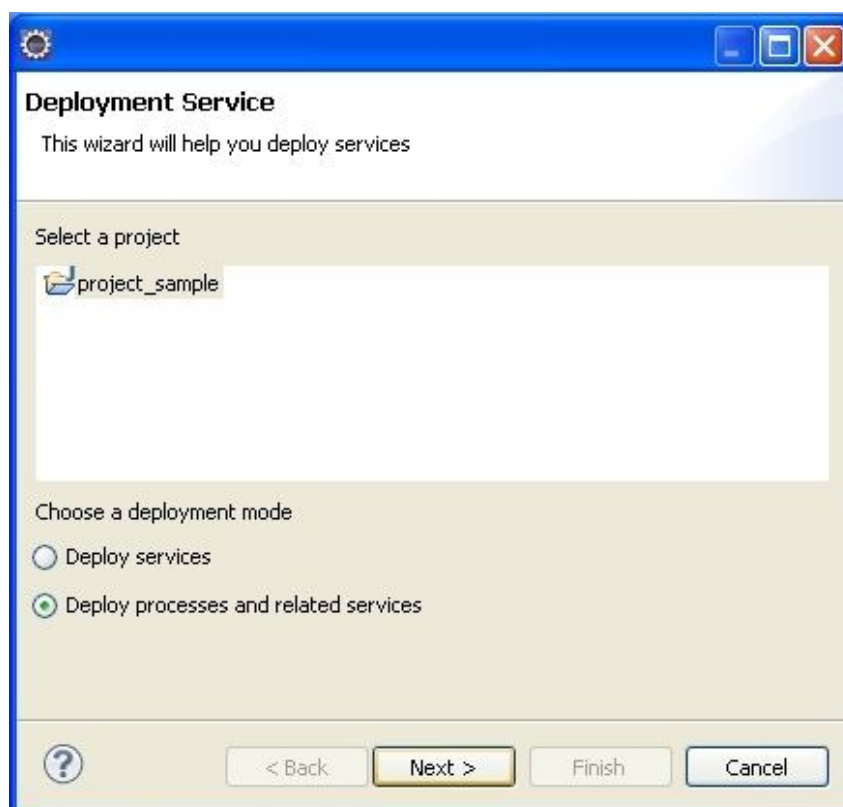
This works only for connectors, if you try to drag and drop a service or another file, the association menu is disabled.

6.3 Deployment

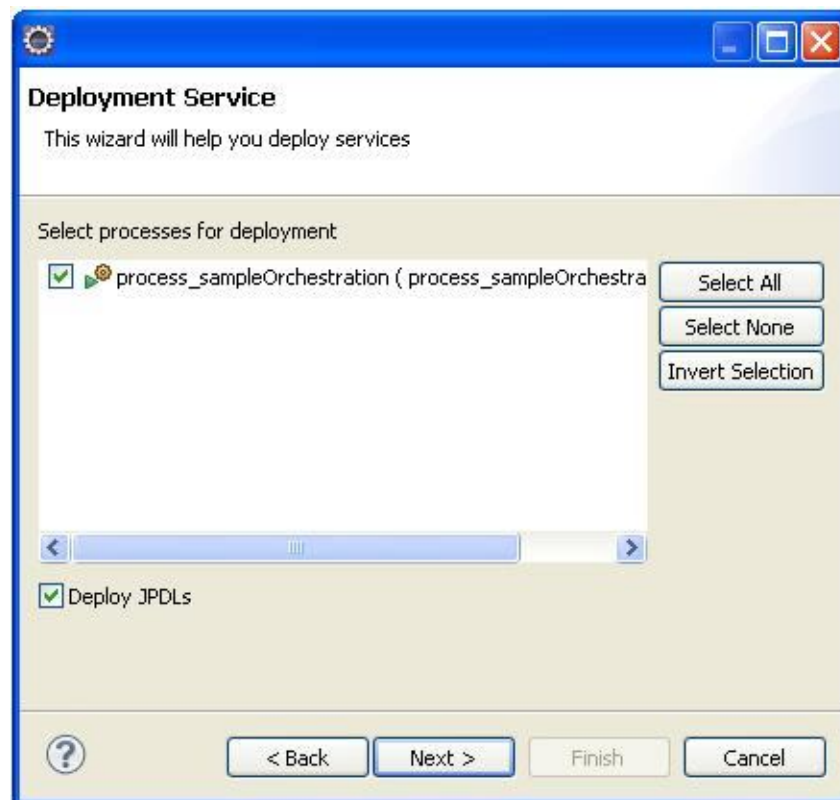
Before to deploy your process you need to start up the service manager. If you are using the All-in-One distribution the service manager is already running inside Tomcat while if you use the Standalone mode you need to start it manually. To deploy the process you have to call the deployment Wizard and choose the process and all the services related to it.

Click on the deployment service icon that is in the main icon bar of Spagic 3 .

In the wizard, there are all projects that are in workspace. Select the project *prova_project* and choose “Deploy services”. Then click to Next.



To deploy the process, check to the name of the service and check also “Deploy JPDLS”: in this way the process will be published on the MetaDB and will be deployed on the service manager.



The deployment wizard opens a new window where all the services and connectors that are used by the process are automatically selected, so we can simply click *Finish* if we want to deploy the process and all the services used.

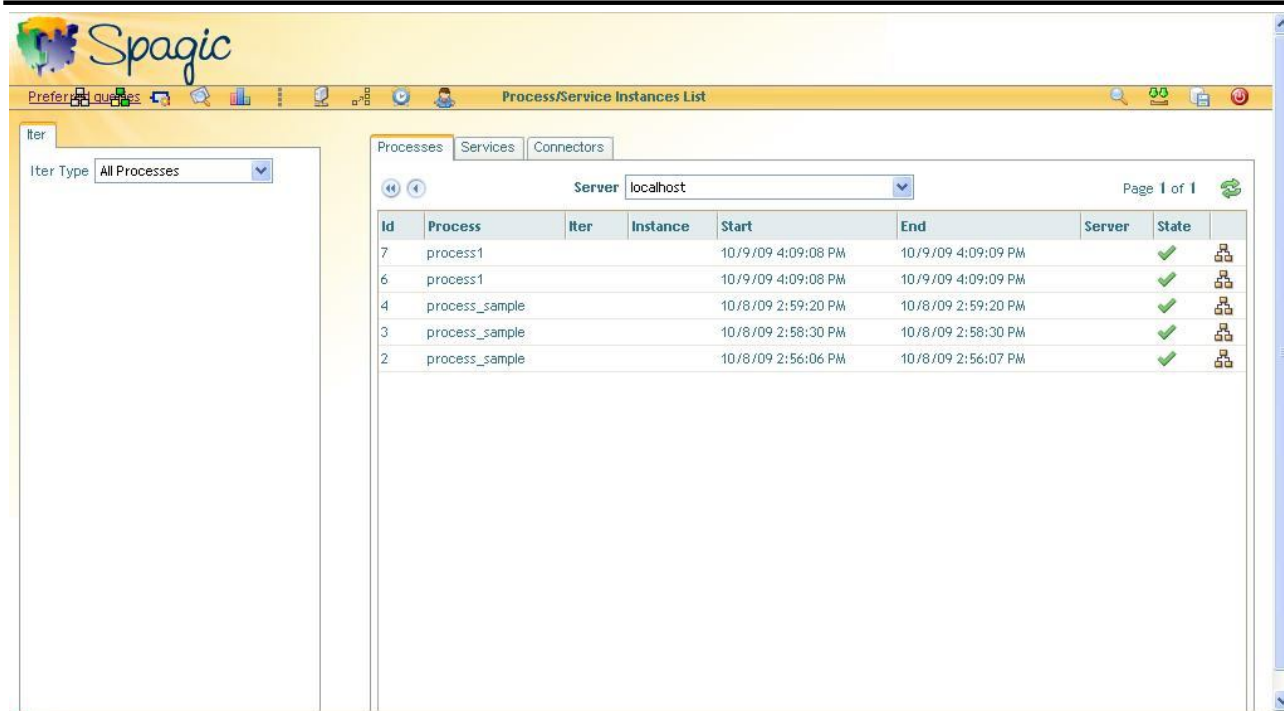
So the process of deployment starts: it appears the window in which is defined the name of the process and we have to click OK, then the deployment continues by itself.

6.4 Test

After the deployment of the process, we can test the whole process. We have the process whose beginning is implemented by a File Poller. So, we have to place an xml file under the folder [C:/test](#).

When the file is placed in the folder, the file poller will pick it up and send it to the main task that is the service *xslt_transformer*. After being transformed by the service that supports the task, the message will be displayed in the console. All the process, services and connectors will be monitored by the Spagic Console. It is remarkable that when you execute a process, and decide to monitor it in Spagic Console the services used by the process are monitored only within the process; you will not find their instances in the tabs *Services* and *Connectors List*.

Spagic3 Getting Started

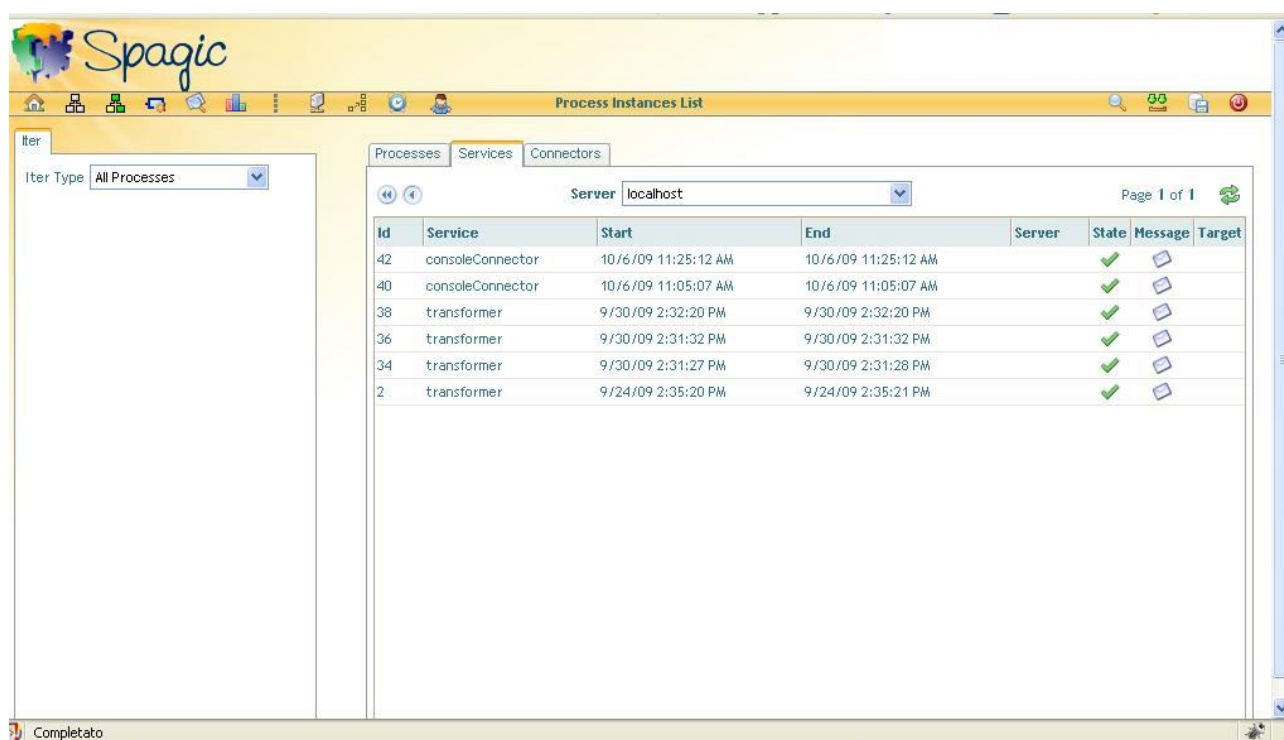


Spagic3 Process/Service Instances List

Iter Type: All Processes

Server: localhost

Id	Process	Iter	Instance	Start	End	Server	State
7	process1			10/9/09 4:09:08 PM	10/9/09 4:09:09 PM		✓
6	process1			10/9/09 4:09:08 PM	10/9/09 4:09:09 PM		✓
4	process_sample			10/8/09 2:59:20 PM	10/8/09 2:59:20 PM		✓
3	process_sample			10/8/09 2:58:30 PM	10/8/09 2:58:30 PM		✓
2	process_sample			10/8/09 2:56:06 PM	10/8/09 2:56:07 PM		✓



Spagic3 Process Instances List

Iter Type: All Processes

Server: localhost

Id	Service	Start	End	Server	State	Message	Target
42	consoleConnector	10/6/09 11:25:12 AM	10/6/09 11:25:12 AM		✓		
40	consoleConnector	10/6/09 11:05:07 AM	10/6/09 11:05:07 AM		✓		
38	transformer	9/30/09 2:32:20 PM	9/30/09 2:32:20 PM		✓		
36	transformer	9/30/09 2:31:32 PM	9/30/09 2:31:32 PM		✓		
34	transformer	9/30/09 2:31:27 PM	9/30/09 2:31:28 PM		✓		
2	transformer	9/24/09 2:35:20 PM	9/24/09 2:35:21 PM		✓		