eBPM Getting Started

Author: Andrea Zoppello
Antonella Miele
Gianfranco Boccalon
Luca Rossato

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

1 Document Goal

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

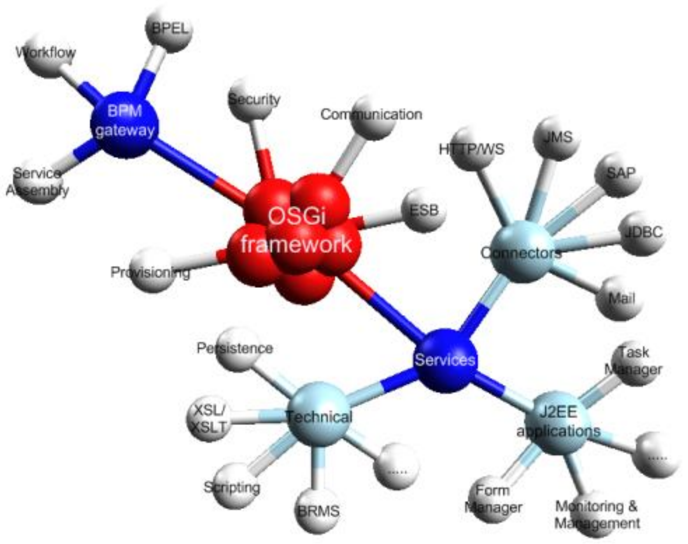
2 Versions History

Version/Release n° :	1.0	**Date**	23/08/2010
Description	First release (English version)		

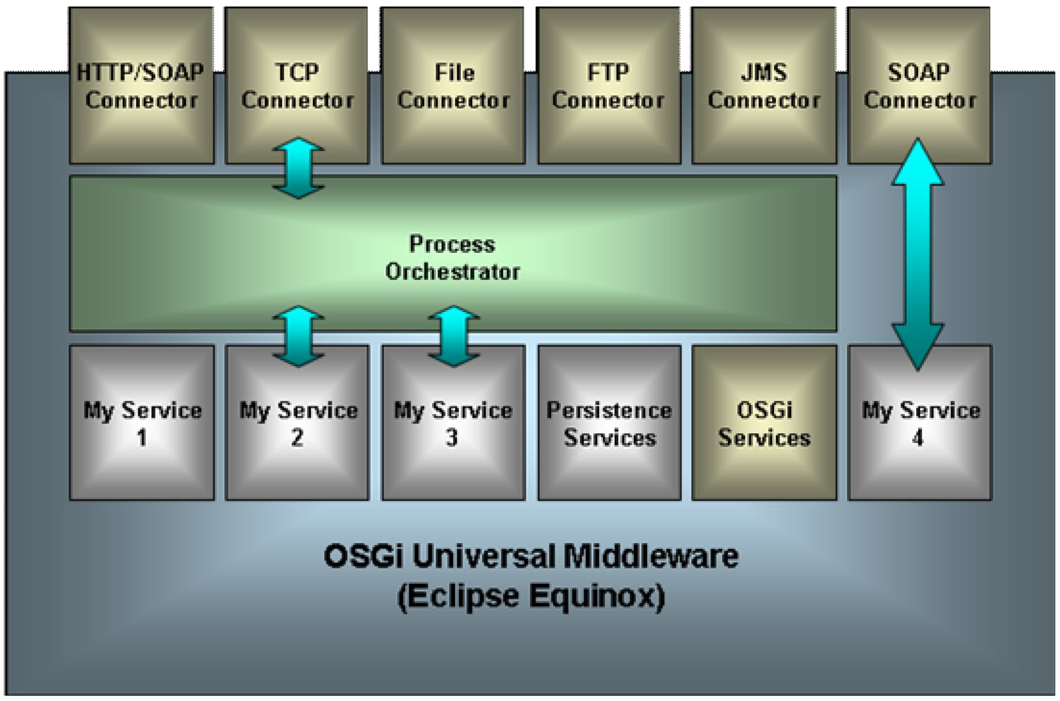
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

3 Introduction

eBPM is a project whose goal is proposing an OSGi Enterprise Universal Middleware 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 eBPM 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.



creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

In eBPM, 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.

Before proceeding with the guided tour we have to explain what is contained in the eBPM package: eBPM is the result of the contribution to Eclipse of most of the core of an open source project called **Spagic** (http://spagoworld.com/xwiki/bin/view/Spagic/).
The contribution work is still in progress: this means that using eBPM without auxiliary tools is quite complex. For this reason, in this tour we’ll use some complementary tools provided by the original project Spagic and the eBPM runtime provided is the basic one with some additional Spagic bundles.
Spagic has been refactored for the eBPM contribution: most of the core bundles has been renamed, some code has been changed to use libraries that are compatible with the EPL license, and the license of all bundles was changed.

In the following table we summarize the features that are included in eBPM, and the ones that are included only in Spagic. As you can see, some features will be ported on eBPM, while others will remain on Spagic (that is the eBPM supported version).

Feature/Tools	**eBPM**	**Spagic**	**Notes**
Service Manager	Yes	Yes	It’s an Eclipse Equinox providing all the runtime bundles.
Connectors	Yes	Yes	
Components	Yes	Yes	
Master/Slave management	Yes	Yes	
InVM connector	Yes	Yes	
Process Orchestrator	-	Yes	Actually the Process Orchestrator is based on JBPM and it’s not releasable with EPL license.
Rules Engine	-	Yes	The Engine is based on JBoss Drools that is EPL compatible, but we have to verify the licenses of all dependencies libraries.
Scripting Engine	Javascript	Javascript/Groovy	The Groovy Engine is EPL compatible, but we have to verify the licenses of all dependencies libraries
JMS Connection factories	-	Yes	The JMS connection factory for ActiveMQ depends on Jencks that is still under Eclipse “triage”
Service distribution	-	Yes	Based on ECF (http://www.eclipse.org/ecf/)

creative commons COMMONS DEED
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

			Distributed EventAdmin. Must add the ECF Distributed EventAdmin to the eBPM runtime.
Designer environment	-	Yes	The designer is called **Spagic Studio**: we are evaluating a partial/full contribution to the eBPM project.
Monitoring Console	-	Yes	The full monitoring environment is the **Spagic Console**. In the eBPM roadmap we foresee to provide a "light" monitoring view directly within the Designer environment.
Form Management	-	Yes	The application that interacts with the Process orchestrator to show to the user the available tasks is the **Spagic TaskList**. In eBPM to interact with the Process Orchestrator you can use the "Workflow API".

3.1 Software 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		✓

3.2 eBPM Package

The eBPM package consists of several separated applications, it contains:

- The **eBPM Service Manager** (a standalone Equinox server)
- The designer environment **Spagic Studio** (a customized Eclipse IDE).

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

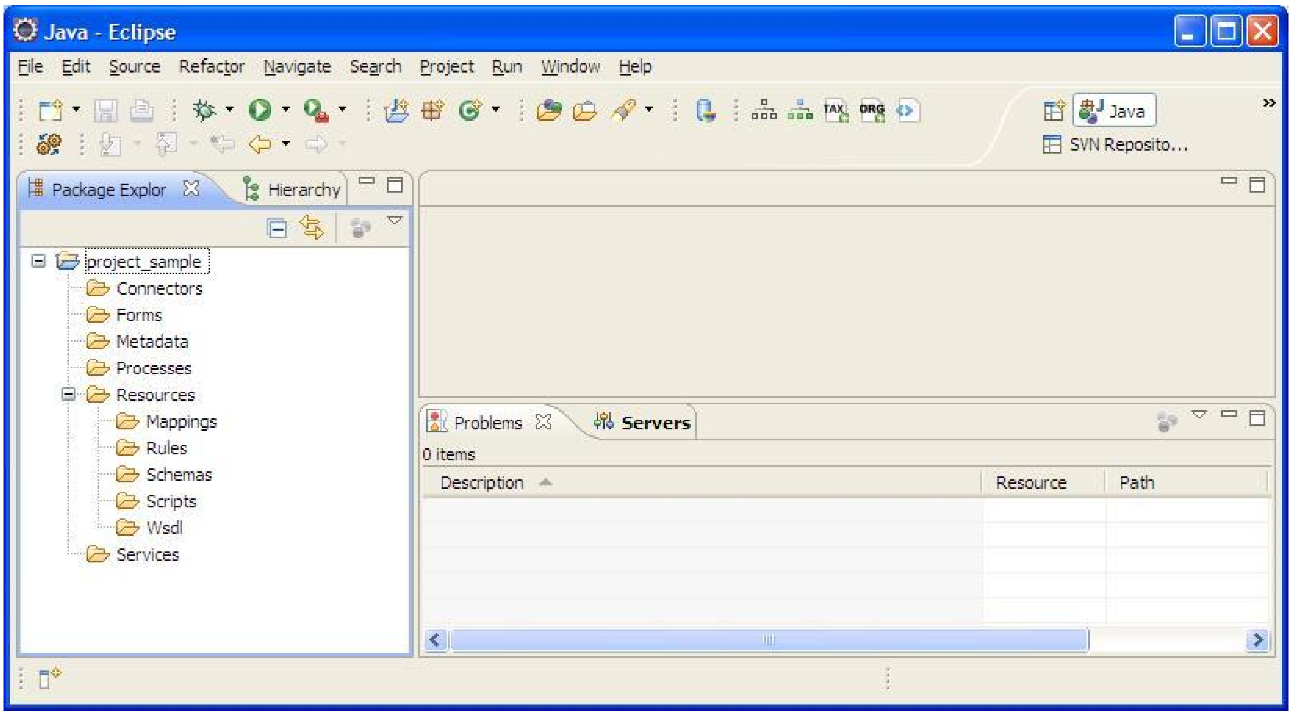
- The monitoring application **Spagic Console** to install into Tomcat (or another servlet container).

3.3 Start the Service Manager

In order to start up the service manager, open the command window and start the eBPM executable that is within the Service Manager directory (from now on we will call it *SERVICE_MANAGER).*
You can also optionally open an OSGi console, towards eBPM, opening a telnet connection on the port 9999. Use the command “help” to have the list of available commands.
Then start the Tomcat containing the console: remember to install the H2 driver in Tomcat before starting it.

3.4 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.
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.

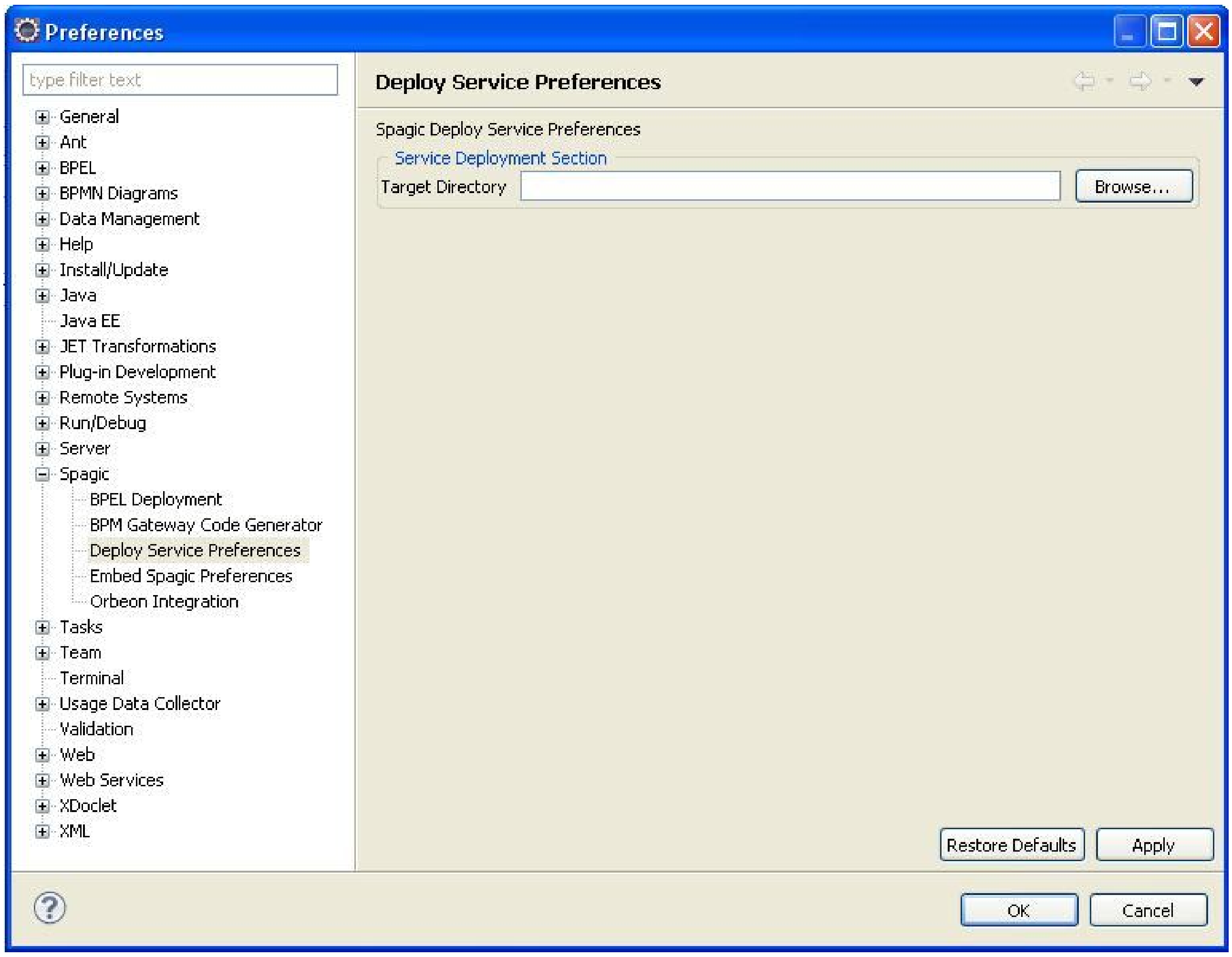


Within the project main folder you will find different subfolders where you can place the components of your project: **connectors**, **processes**, **services**, **forms**, **metadata** and **resources**, further divided in other several subfolders.
Before to start the design of your project components 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 the folder ***SERVICE_MANAGER\default***. If you want to

creative commons COMMONS DEED
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

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: you simply need to change the value of **-Dspagic.home** parameter of eBPM.ini file inside the *SERVICE_MANAGER* directory (by default it's commented).



creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

4 Services/Connectors

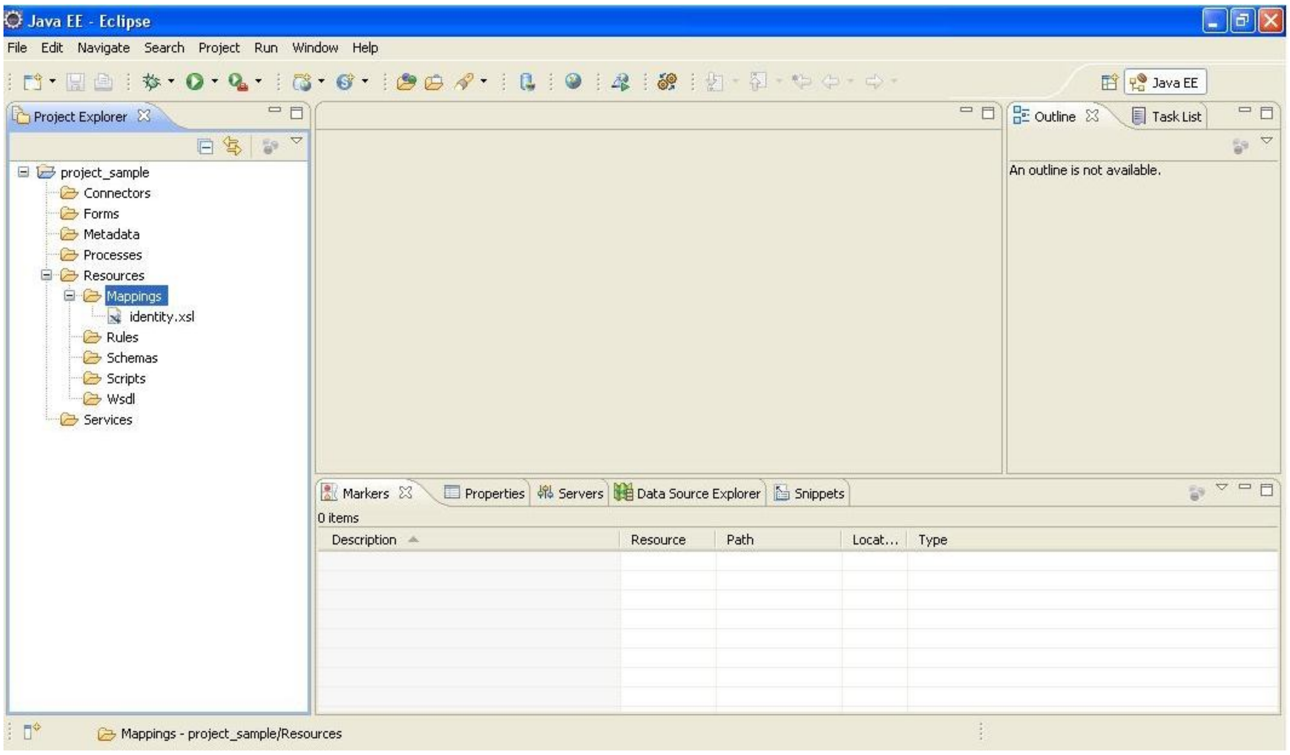
4.1 Configuration of a new service

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 eBPM 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.

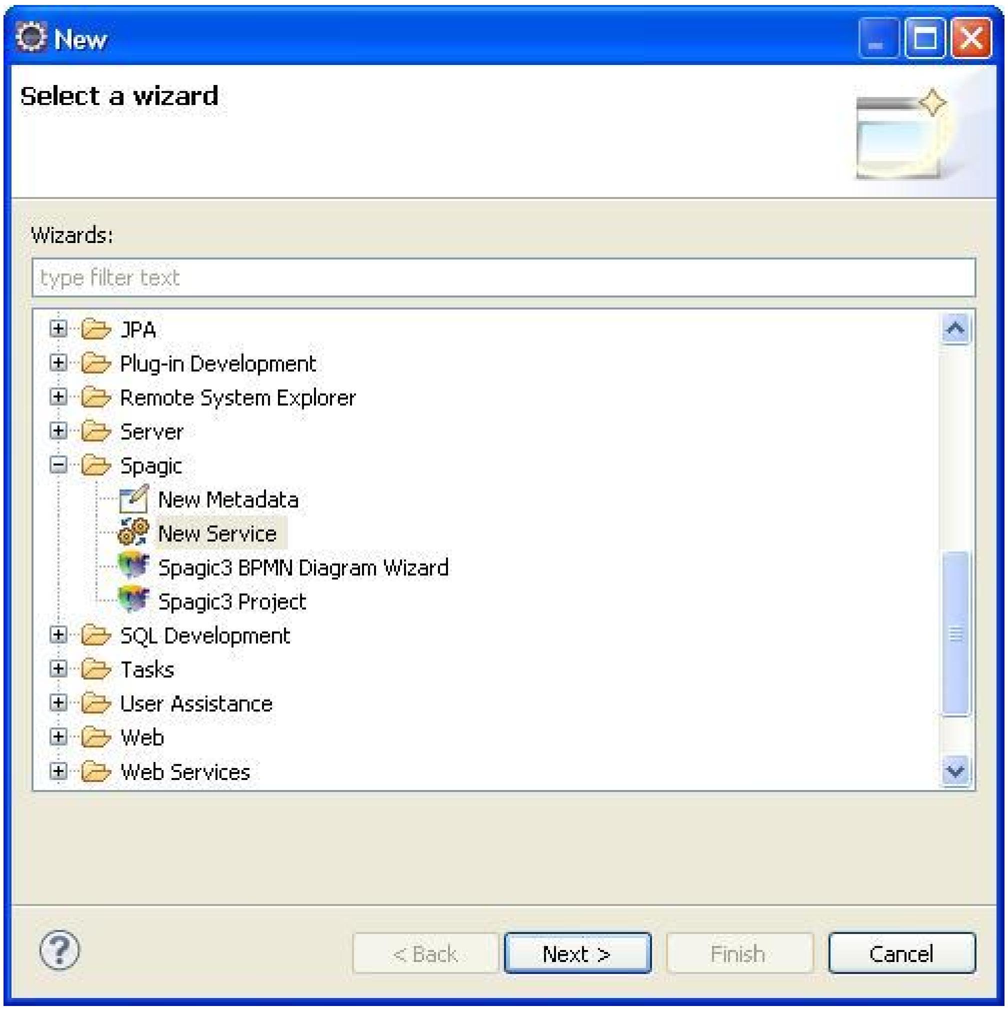
Placing all the resources used by services in project under the *Resources* folder before configuring the service, is more comfortable for the feature of drag and drop of files, available in several service configuration in Spagic 3.



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*.

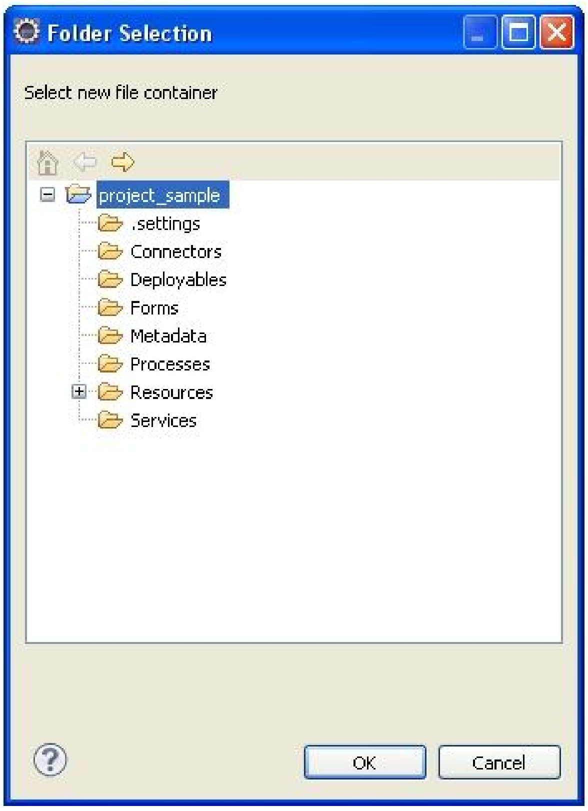
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/



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.

creative commons COMMONS DEED

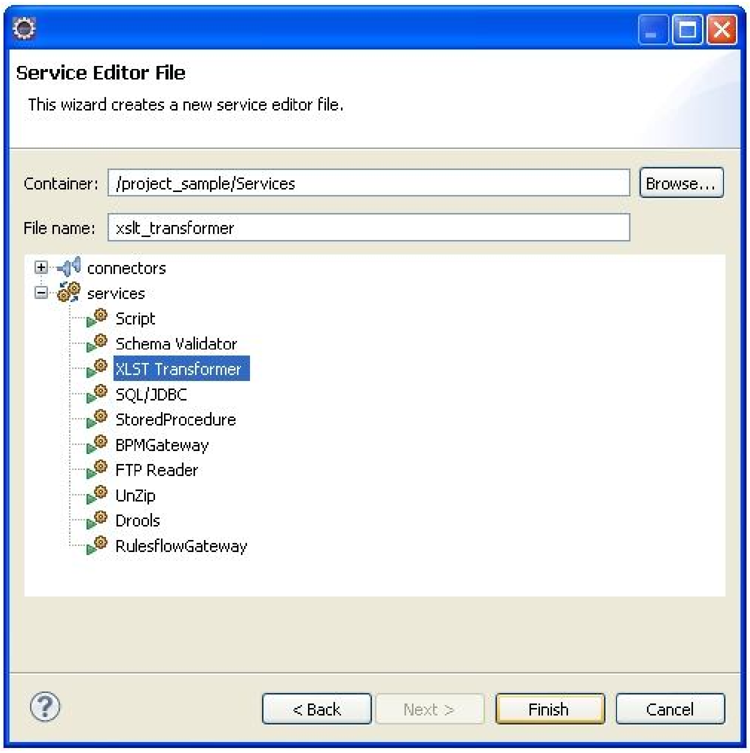
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/


Folder Selection

Select new file container

project_sample
.settings
Connectors
Deployables
Forms
Metadata
Processes
Resources
Services

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*.


Service Editor File

This wizard creates a new service editor file.

Container: /project_sample/Services

File name: xslt_transformer

connectors
services
Script
Schema Validator
XLST Transformer
SQL/JDBC
StoredProcedure
BPMGateway
FTP Reader
UnZip
Drools
RulesflowGateway

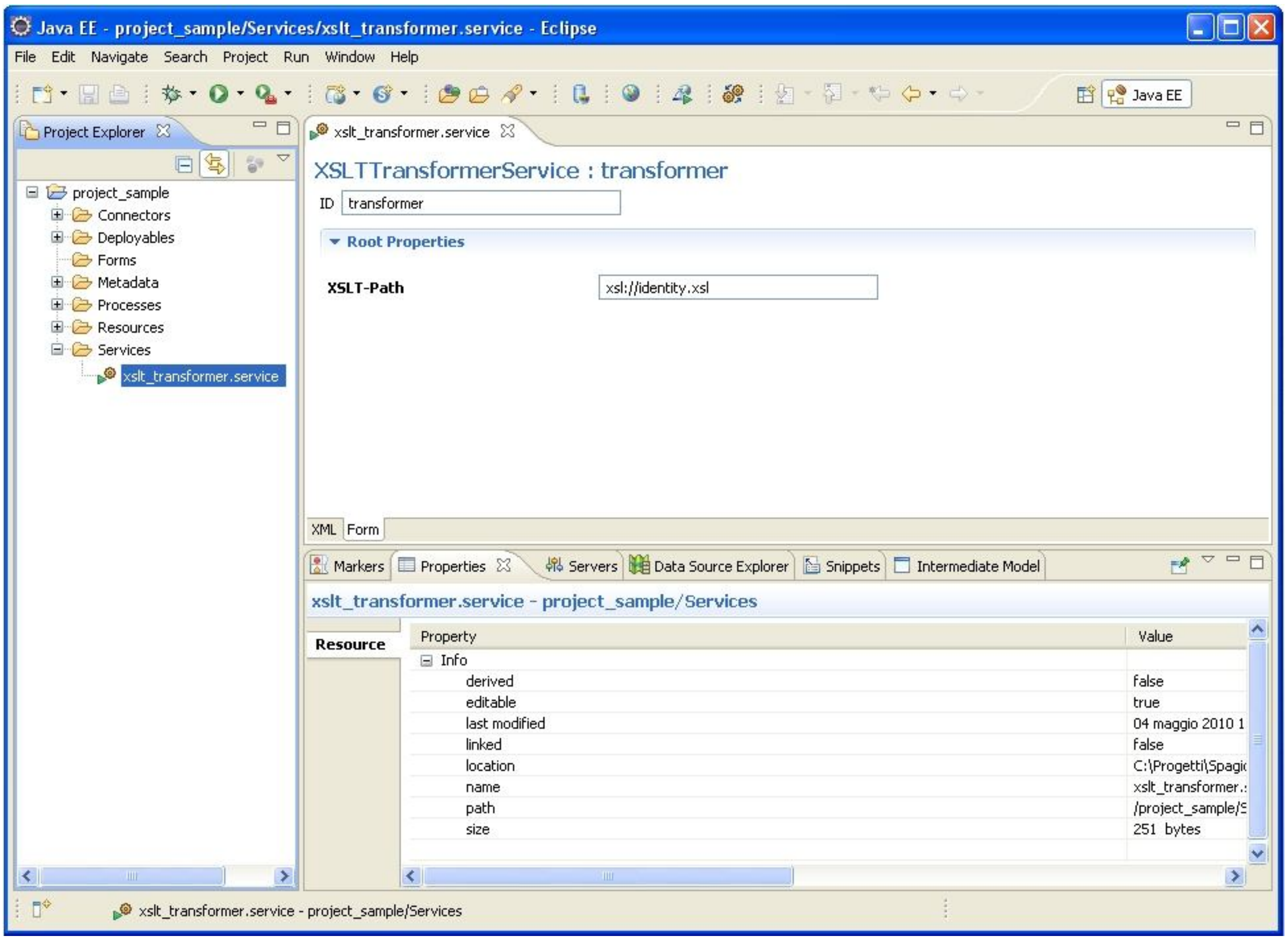
creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

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.


Java EE - project_sample/Services/xslt_transformer.service - Eclipse
File Edit Navigate Search Project Run Window Help
Java EE
Project Explorer
project_sample
Connectors
Deployables
Forms
Metadata
Processes
Resources
Services
xslt_transformer.service
XML Form
Markers Properties Servers Data Source Explorer Snippets Intermediate Model
Resource
xslt_transformer.service - project_sample/Services

XSLTTransformerService : transformer

ID: transformer

Root Properties

XSLT-Path: xsl://identity.xsl

xslt_transformer.service - project_sample/Services

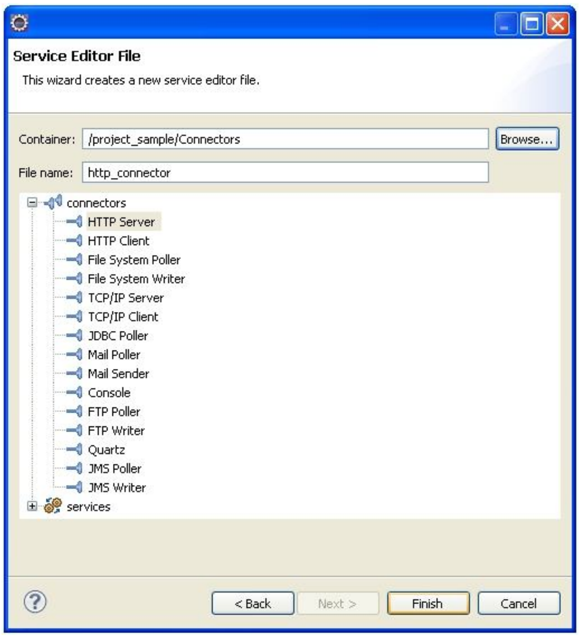
Property	Value
Info	
derived	false
editable	true
last modified	04 maggio 2010 1
linked	false
location	C:\Progetti\Spagic
name	xslt_transformer.:
path	/project_sample/S
size	251 bytes

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.

creative commons
COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

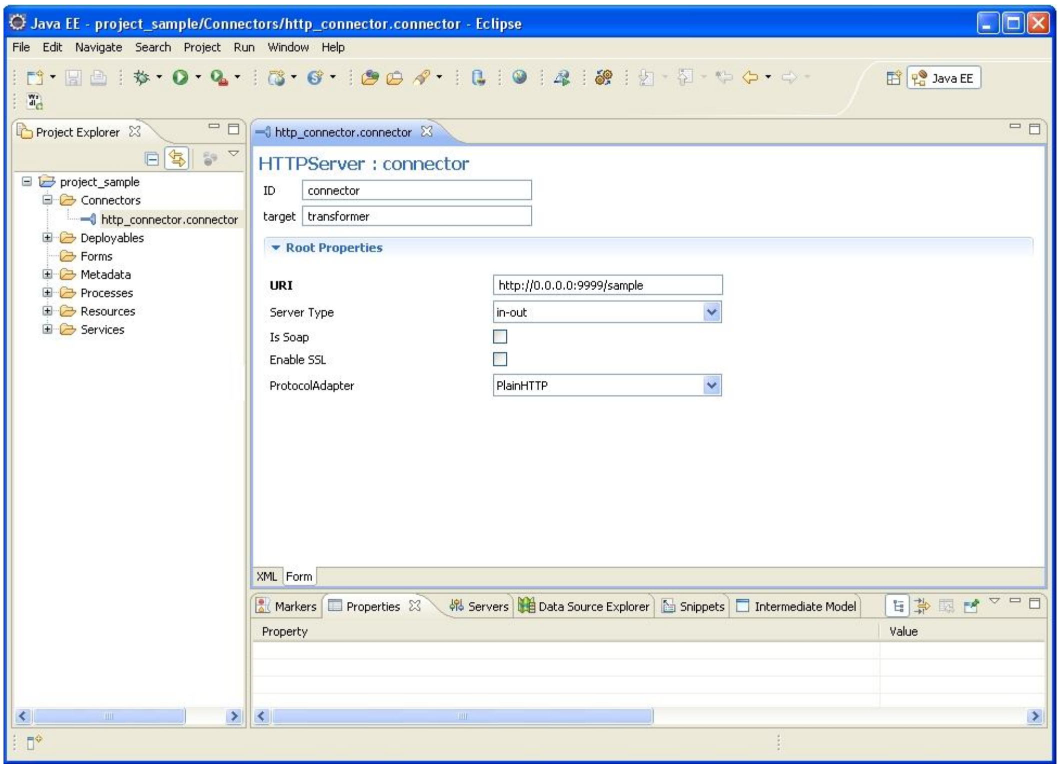

Service Editor File

This wizard creates a new service editor file.

Container: /project_sample/Connectors

File name: http_connector

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://0.0.0.0:10000/sample” and select in the field Server Type “in-out”. To link the service to the connector, just drag and drop 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.


Java EE - project_sample/Connectors/http_connector.connector - Eclipse

HTTPServer : connector

ID: connector

target: transformer

Root Properties

URI: http://0.0.0.0:9999/sample

Server Type: in-out

Is Soap: ☐

Enable SSL: ☐

ProtocolAdapter: PlainHTTP

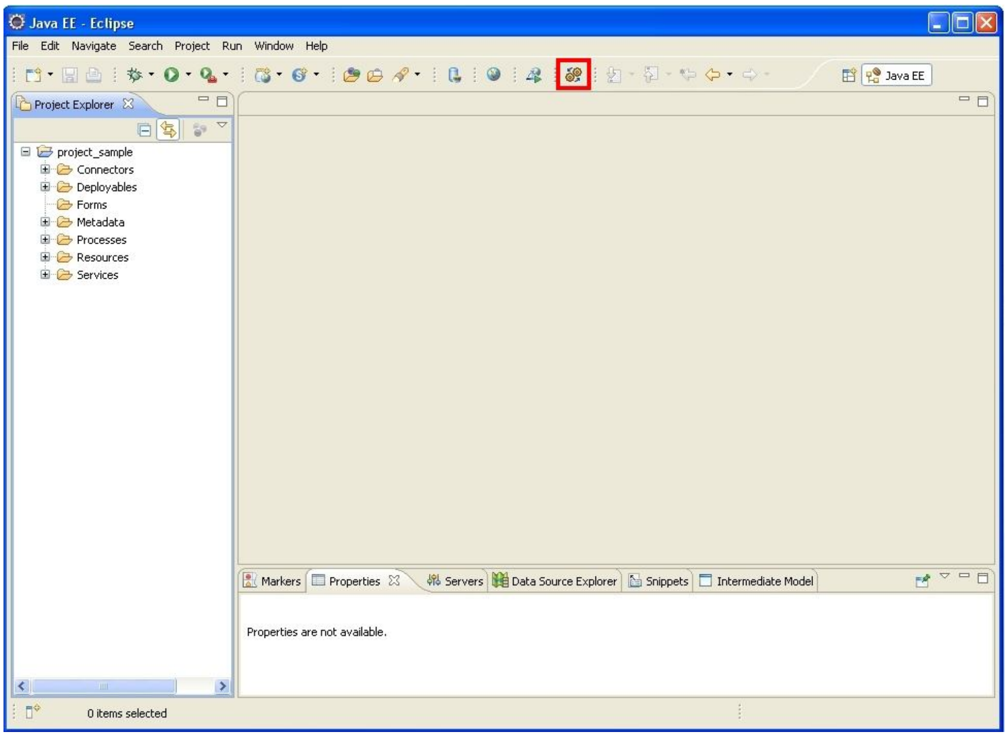
creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

4.2 Deployment

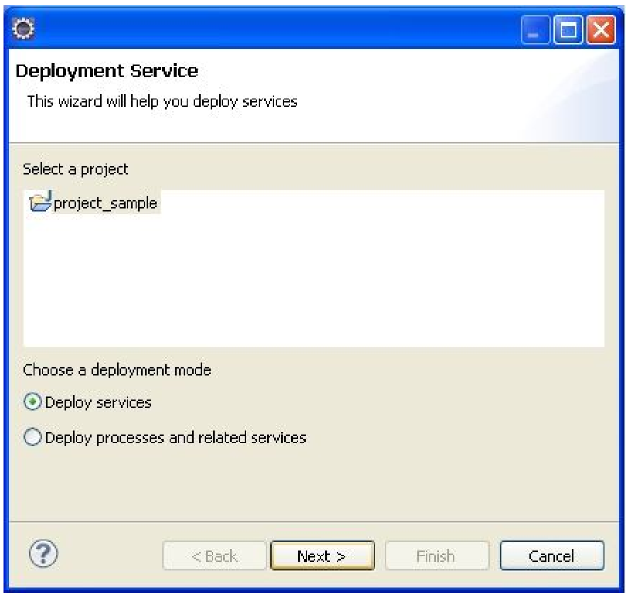
Before to deploy your service you need to start up the service manager (see 3.3-Start the Service Manager).

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.



Properties are not available.

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



Deployment Service

This wizard will help you deploy services

Select a project

project_sample

Choose a deployment mode

(●) Deploy services

( ) Deploy processes and related services

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

creative commons COMMONS DEED

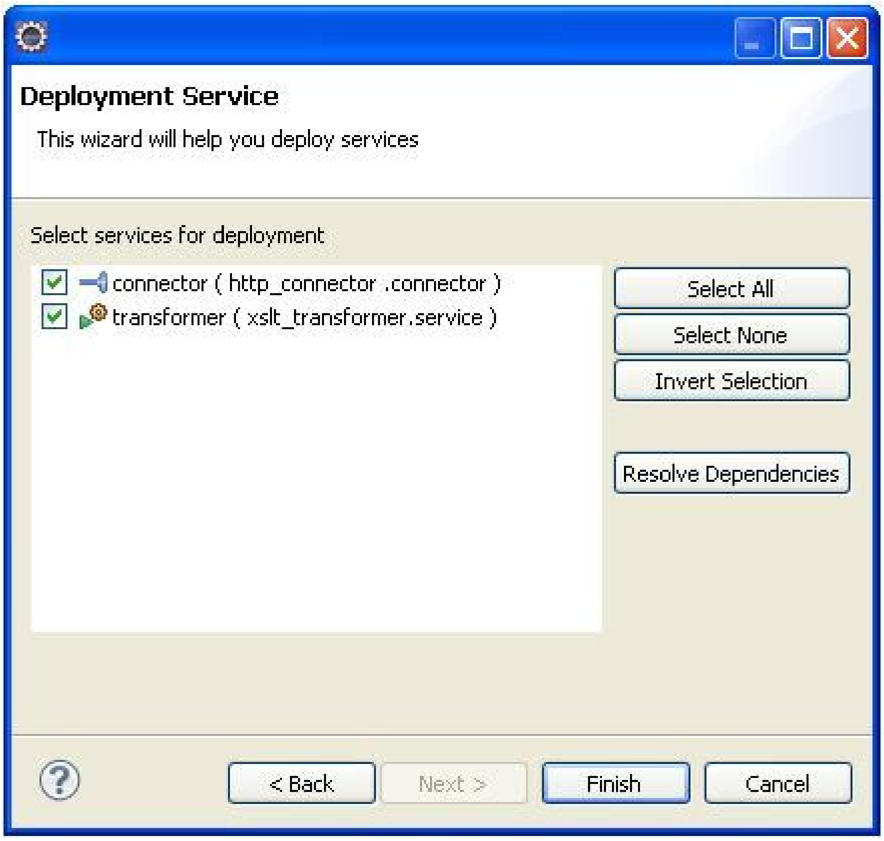
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

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.


Deployment Service

This wizard will help you deploy services

Select services for deployment

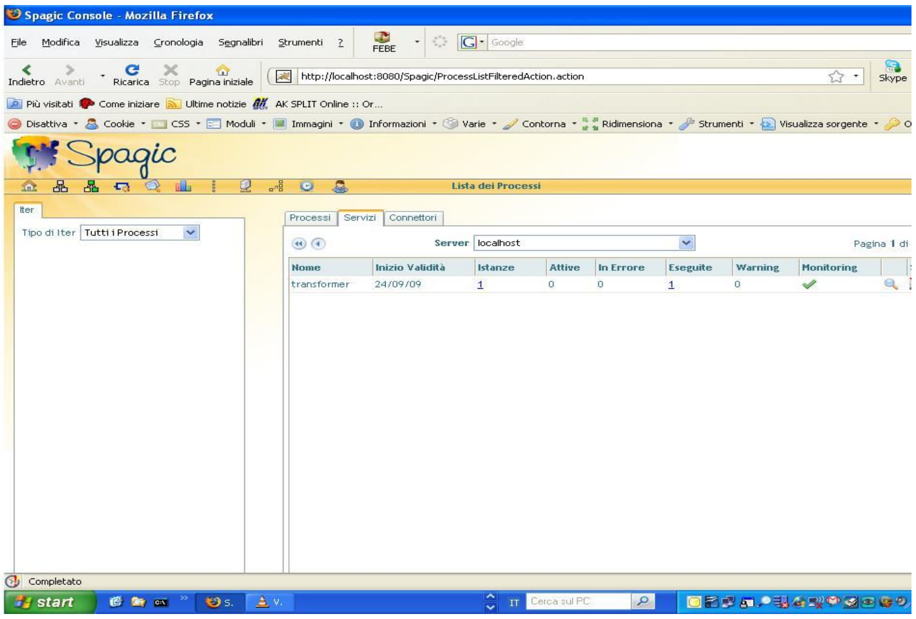
- ✓ connector (http_connector .connector)
- ✓ transformer (xslt_transformer.service)

4.3 Test

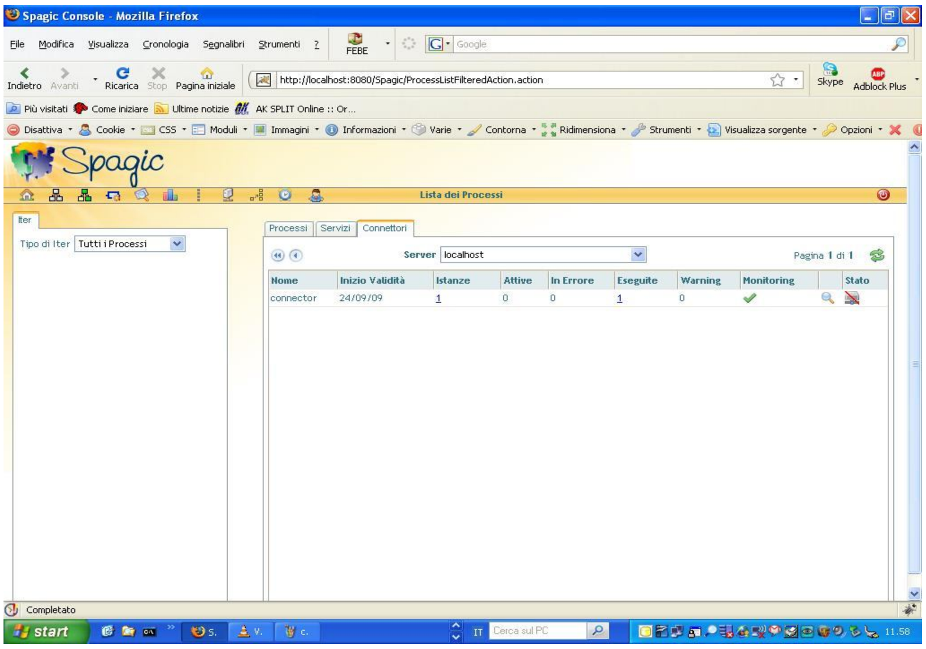
After deploying services and connectors we can see all items in the Spagic console. For installation and use of Spagic Console, please refer to the document "*Spagic3 Console.pdf*". In Spagic Console, clicking on the icon Process/Services List we have the list of all the processes, services and components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

creative commons COMMONS DEED
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

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.



Nome	Inizio Validità	Istanze	Attive	In Errore	Eseguite	Warning	Monitoring
transformer	24/09/09	1	0	0	1	0	✓

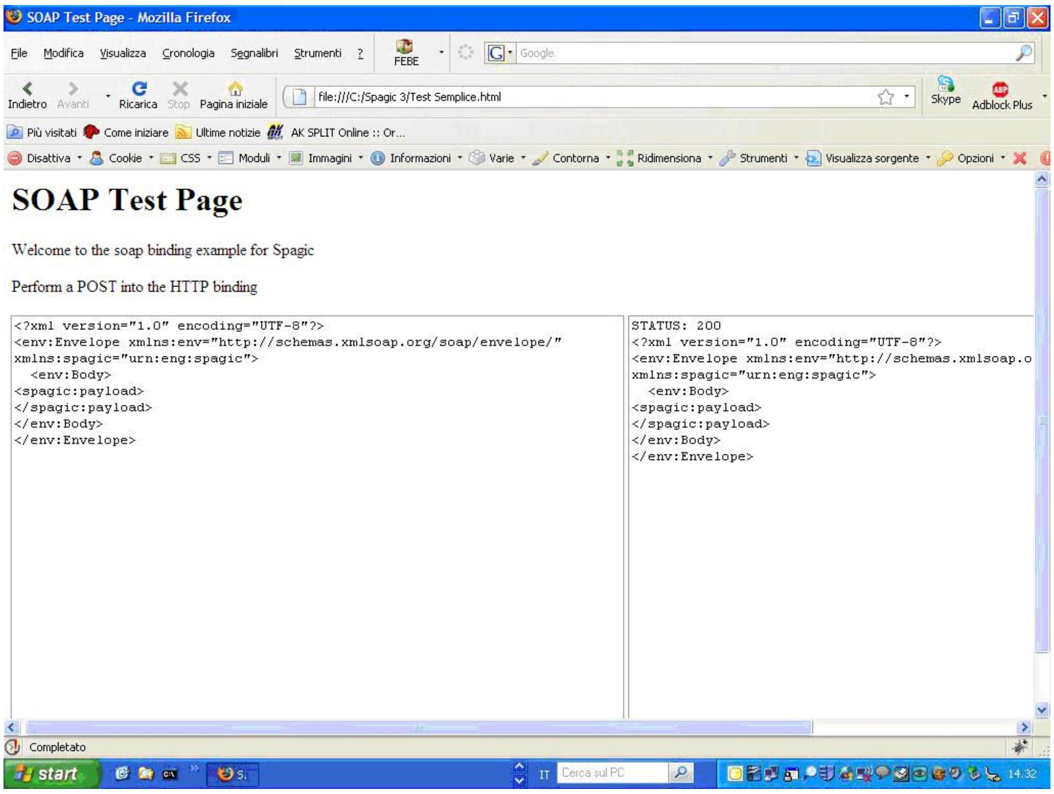


Nome	Inizio Validità	Istanze	Attive	In Errore	Eseguite	Warning	Monitoring	Stato
connector	24/09/09	1	0	0	1	0	✓	

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.

creative commons
COMMONS DEED
Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/


SOAP Test Page - Mozilla Firefox
file:///C:/Spagic 3/Test Semplice.html
Completato

SOAP Test Page

Welcome to the soap binding example for Spagic

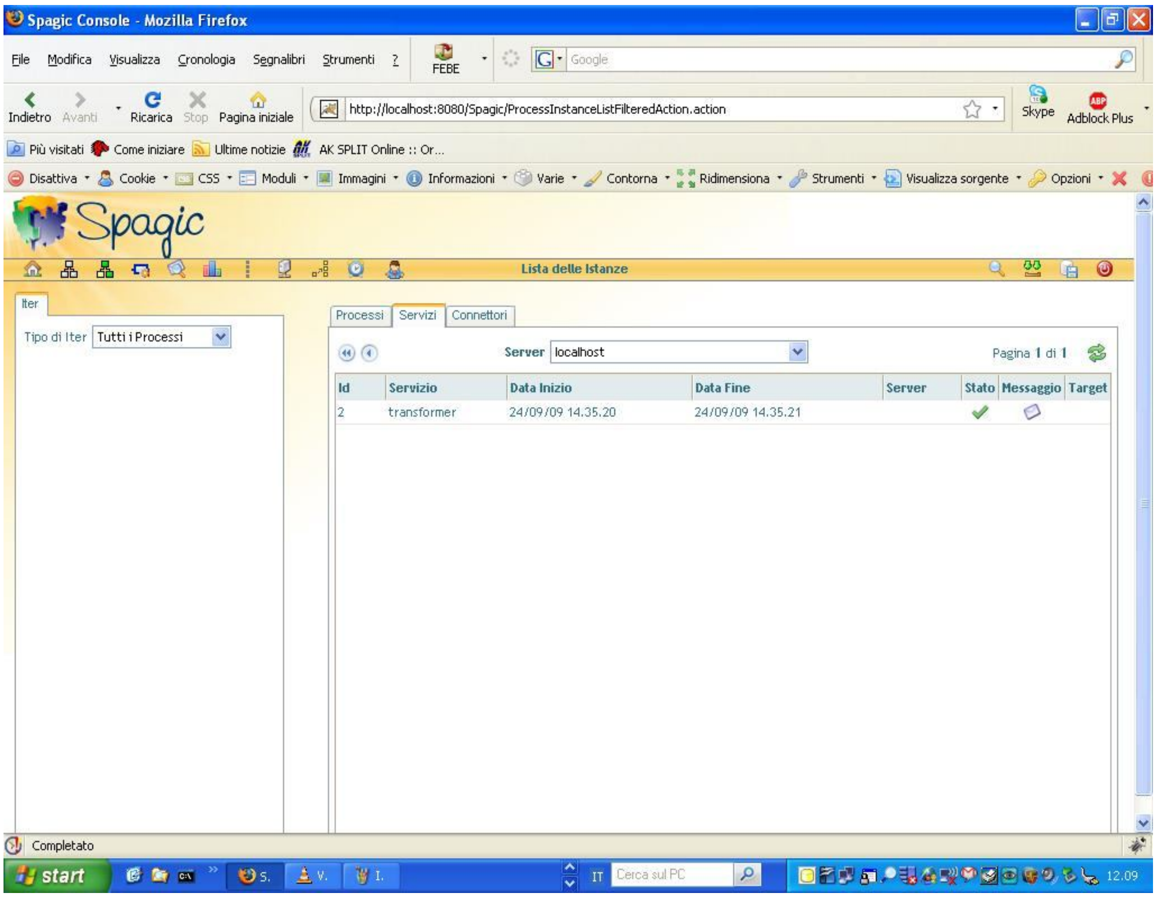
Perform a POST into the HTTP binding

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:spagic="urn:eng:spagic">
  <env:Body>
<spagic:payload>
</spagic:payload>
</env:Body>
</env:Envelope>
```

```
STATUS: 200
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.o
xmlns:spagic="urn:eng:spagic">
  <env:Body>
<spagic:payload>
</spagic:payload>
</env:Body>
</env:Envelope>
```

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

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

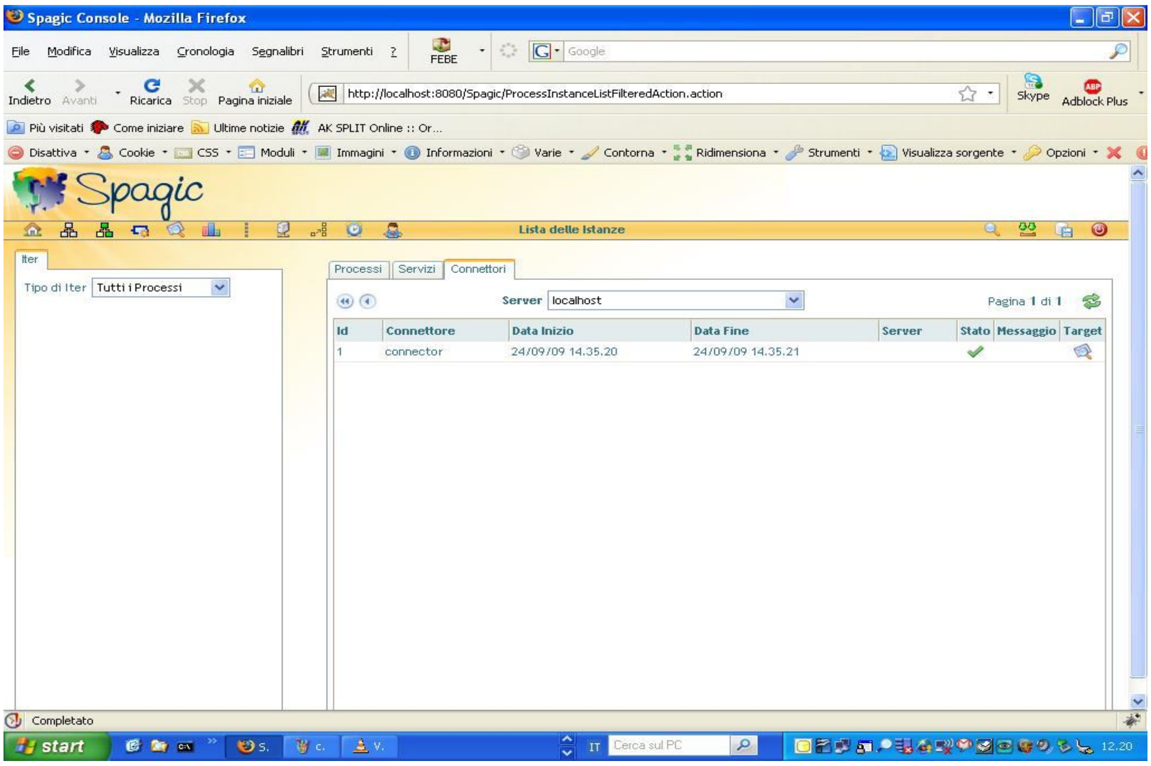

Spagic Console - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti ?
Indietro Avanti Ricarica Stop Pagina iniziale
Spagic
Lista delle Istanze
Iter
Processi
Servizi
Connettori
Completato

http://localhost:8080/Spagic/ProcessInstanceListFilteredAction.action

Tipo di Iter: Tutti i Processi

Server: localhost — Pagina 1 di 1

Id	Servizio	Data Inizio	Data Fine	Server	Stato	Messaggio	Target
2	transformer	24/09/09 14.35.20	24/09/09 14.35.21		✓		


Spagic Console - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti ?
Indietro Avanti Ricarica Stop Pagina iniziale
Spagic
Lista delle Istanze
Iter
Processi
Servizi
Connettori
Completato

http://localhost:8080/Spagic/ProcessInstanceListFilteredAction.action

Tipo di Iter: Tutti i Processi

Server: localhost — Pagina 1 di 1

Id	Connettore	Data Inizio	Data Fine	Server	Stato	Messaggio	Target
1	connector	24/09/09 14.35.20	24/09/09 14.35.21		✓		

creative commons
COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

5 Processes

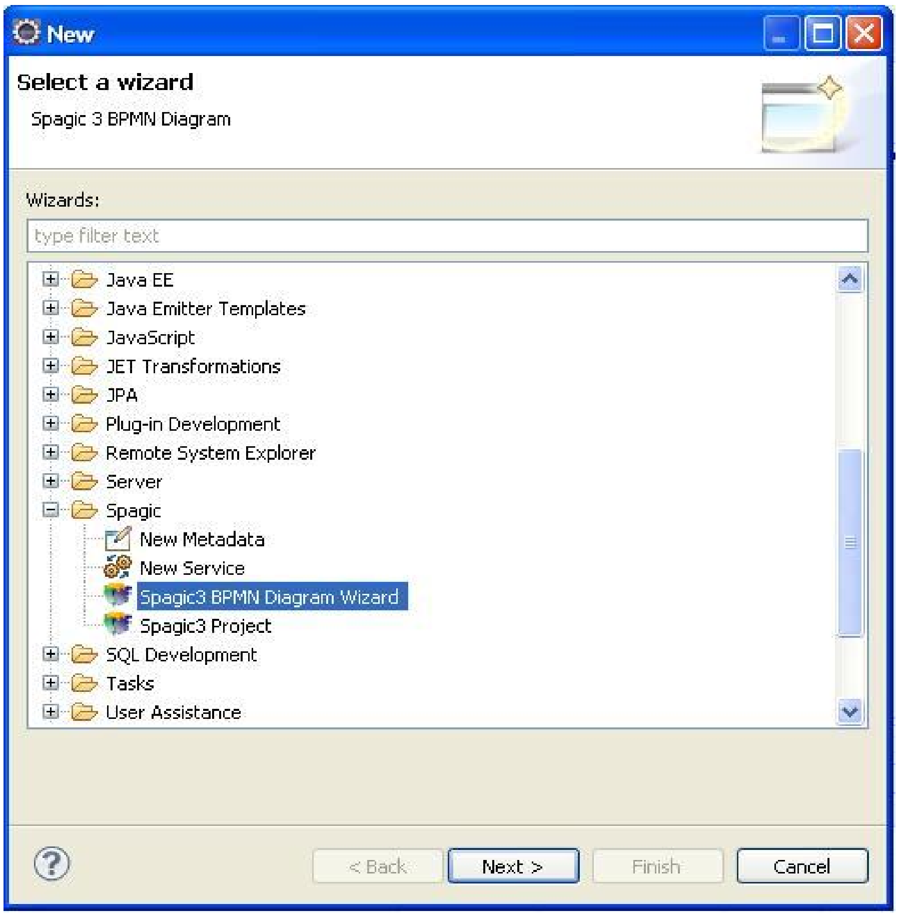
5.1 Creation of a new process

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 install **Graphviz** and configure **Spagic Studio** to know the location of the Graphviz executable. 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*.


New

Select a wizard

Spagic 3 BPMN Diagram

Wizards:

type filter text

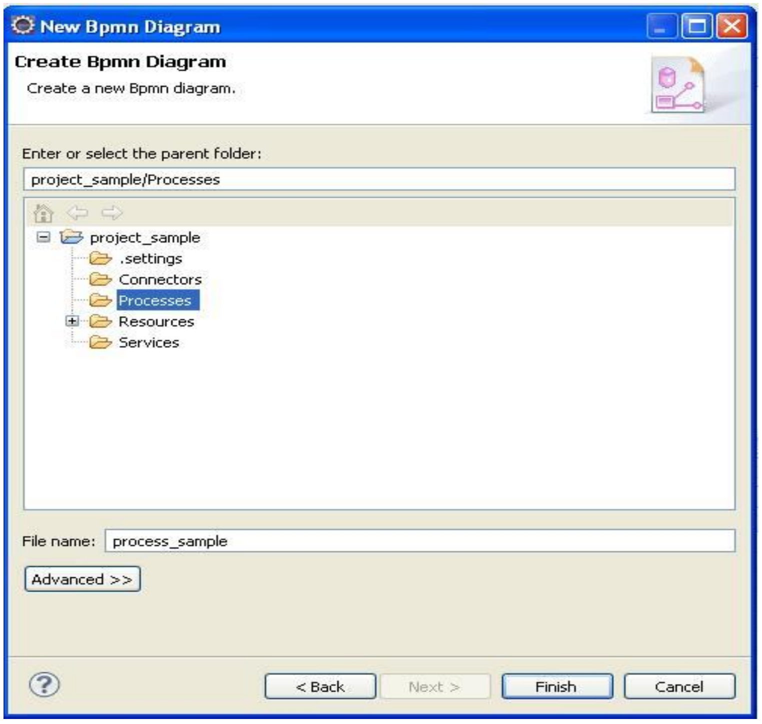
- Java EE
- Java Emitter Templates
- JavaScript
- JET Transformations
- JPA
- Plug-in Development
- Remote System Explorer
- Server
- Spagic
  - New Metadata
  - New Service
  - Spagic3 BPMN Diagram Wizard
  - Spagic3 Project
- SQL Development
- Tasks
- User Assistance

< Back | Next > | Finish | Cancel

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

creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/


New Bpmn Diagram

Create Bpmn Diagram

Create a new Bpmn diagram.

Enter or select the parent folder:

project_sample/Processes

project_sample
.settings
Connectors
Processes
Resources
Services

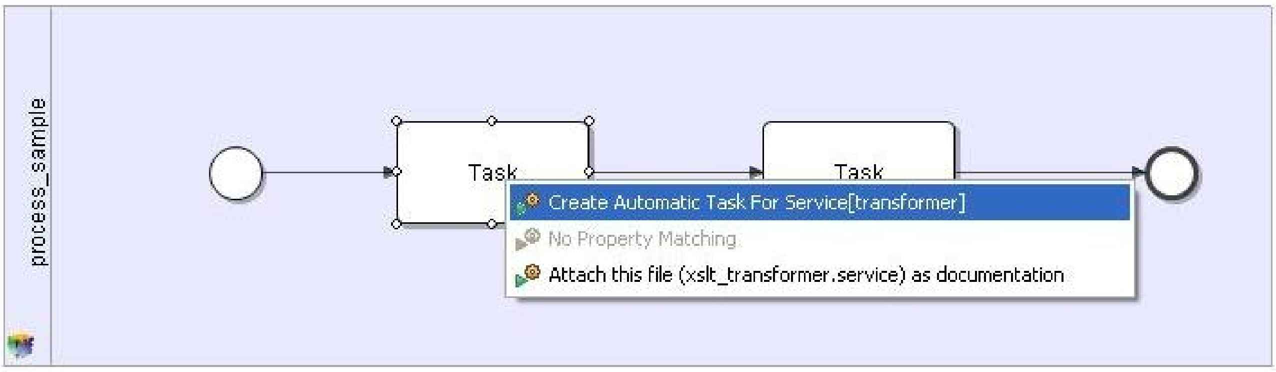
File name: process_sample

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*.

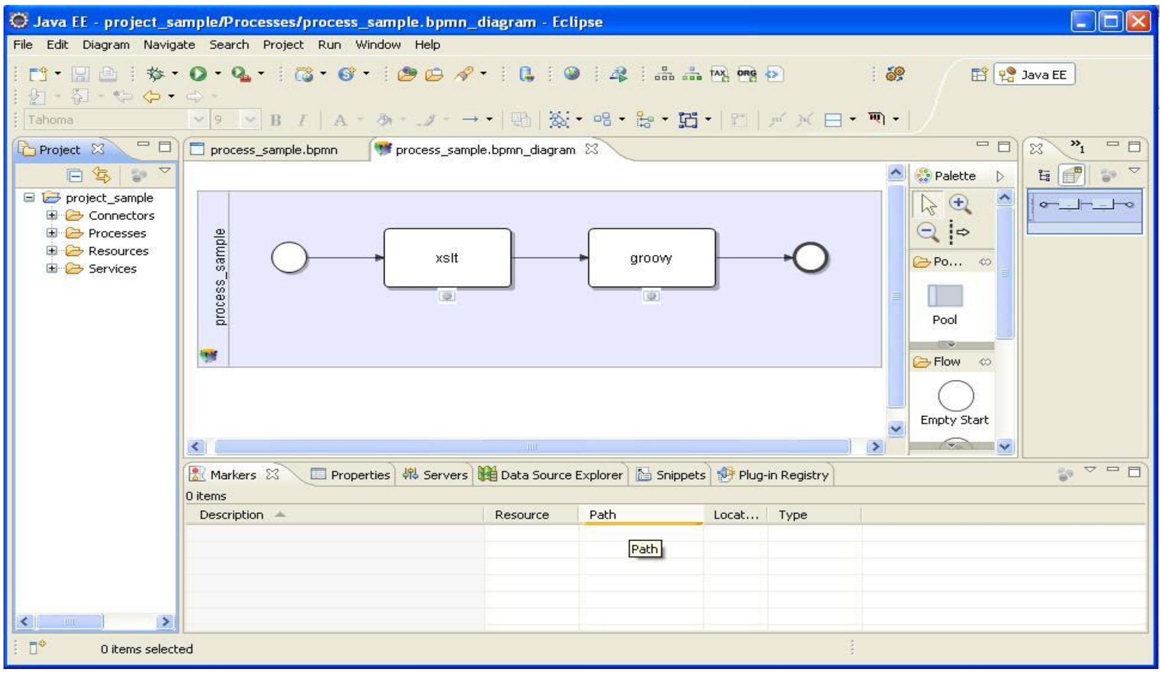
To attach the services to the steps of the process you just have to drag and drop the services to the steps of the process, and choose "*Create Automatic Task for Service*". The service will be attached to the task.

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


process_sample
Task
Task
Create Automatic Task For Service[transformer]
No Property Matching
Attach this file (xslt_transformer.service) as documentation

creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

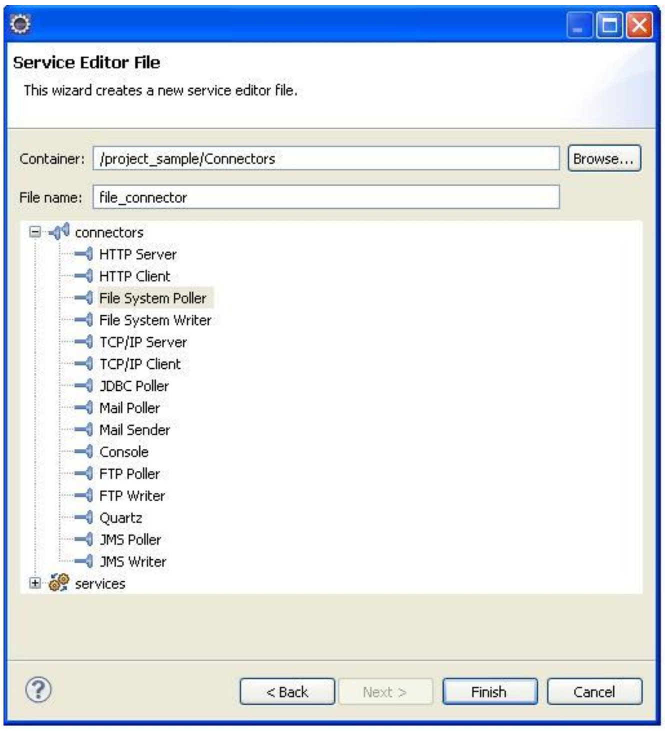


The BPMN Diagram Wizard created automatically also the service needed in order to execute our process, it is under Services folder and it's called process_sampleOrchestration.service.

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.


Service Editor File

This wizard creates a new service editor file.

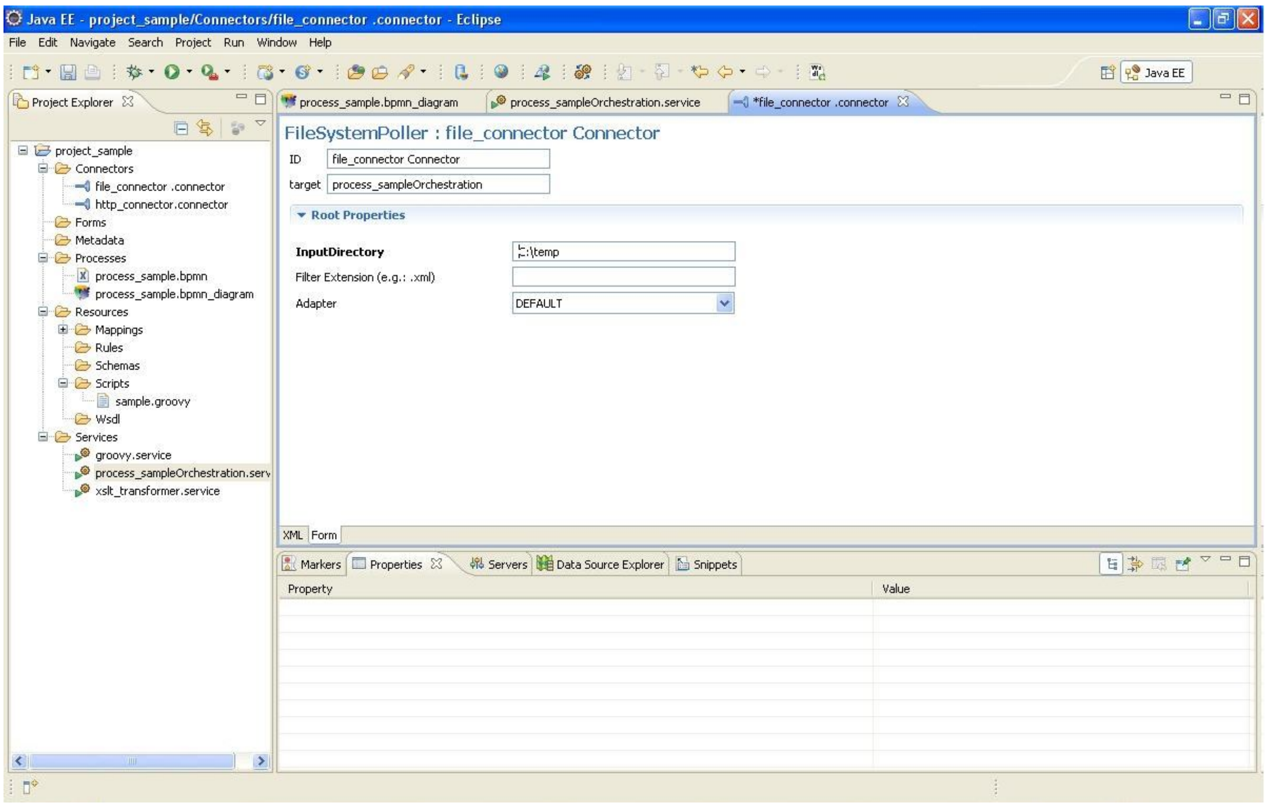
Container: /project_sample/Connectors

File name: file_connector

creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

In the Service editor of the *file_connector* we have to define the ID, the directory to check in, and the target. Leave the default ADAPTER. The target field will be defined later.



FileSystemPoller : file_connector Connector

ID: file_connector Connector

target: process_sampleOrchestration

Root Properties

InputDirectory: E:\temp

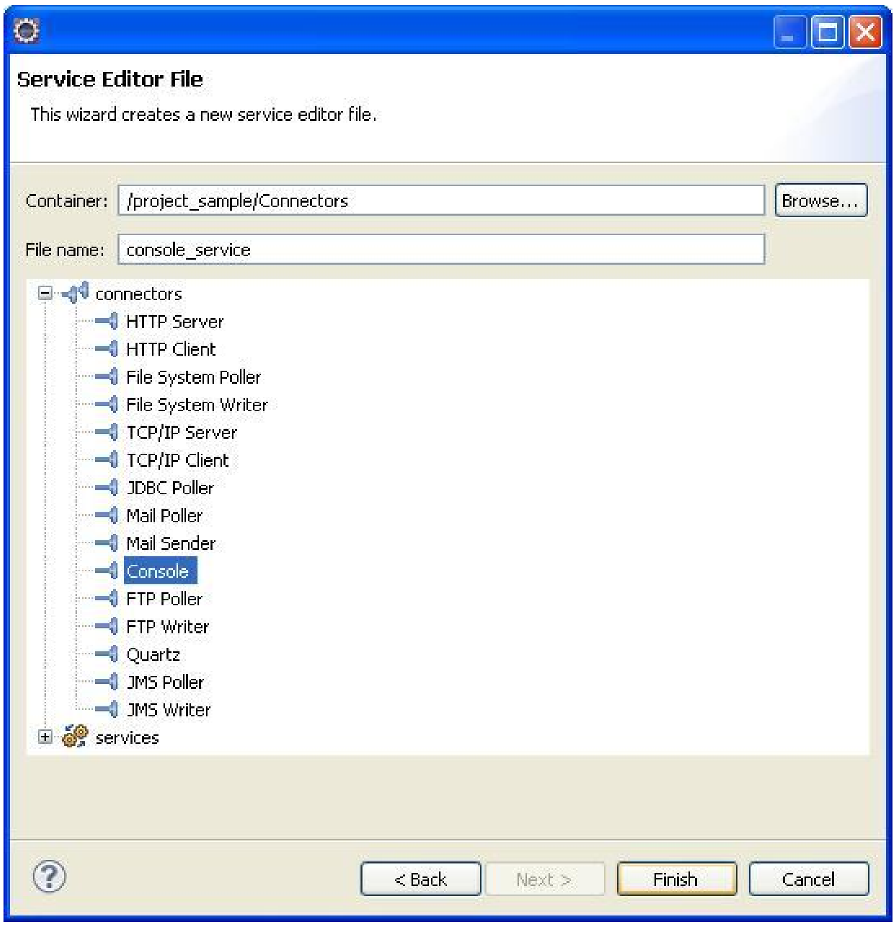
Filter Extension (e.g.: .xml):

Adapter: DEFAULT

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.

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/


Service Editor File

This wizard creates a new service editor file.

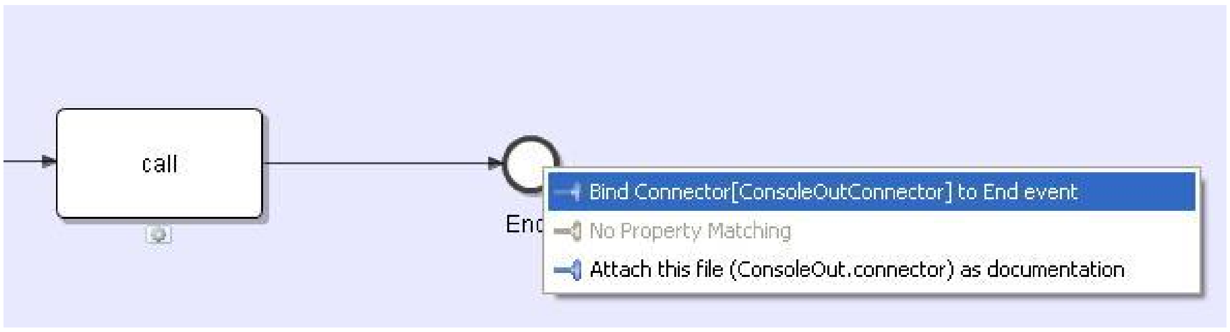
Container: /project_sample/Connectors

File name: console_service

- connectors
  - HTTP Server
  - HTTP Client
  - File System Poller
  - File System Writer
  - TCP/IP Server
  - TCP/IP Client
  - JDBC Poller
  - Mail Poller
  - Mail Sender
  - Console
  - FTP Poller
  - FTP Writer
  - Quartz
  - JMS Poller
  - JMS Writer
- services

This connector has no parameters, except the ID.

Once created the start and 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.


call
End
Bind Connector[ConsoleOutConnector] to End event
No Property Matching
Attach this file (ConsoleOut.connector) as documentation

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

5.2 Deployment

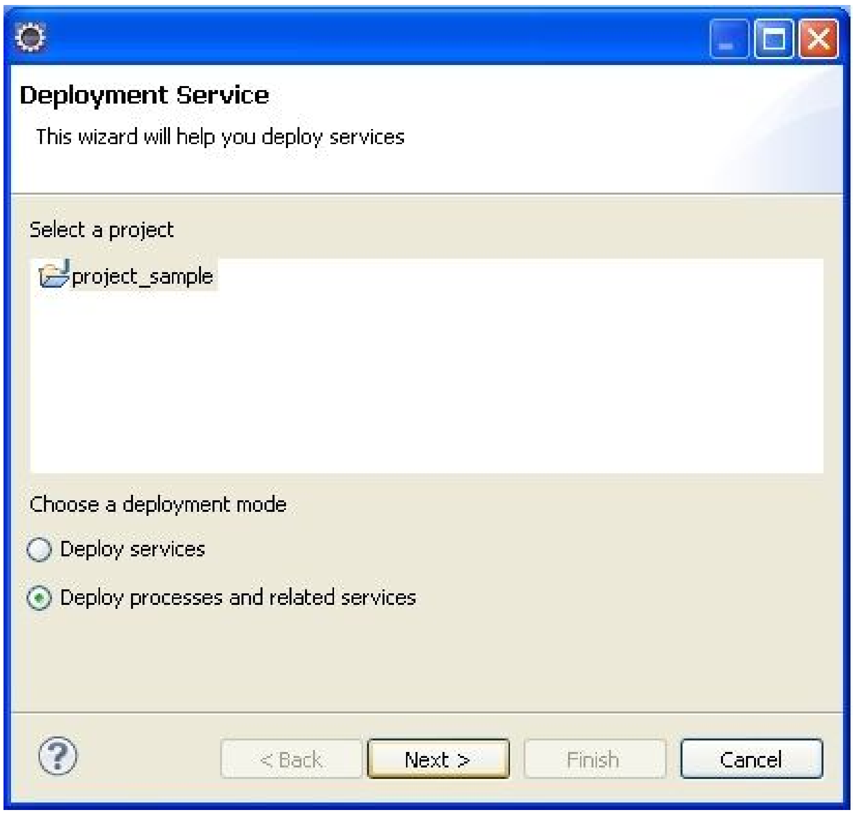
Before deploying your process, you need to start up the service manager (if not already running).

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

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 *project_sample* and choose *"Deploy processes and related services"*. Then click to Next.



Deployment Service

This wizard will help you deploy services

Select a project

project_sample

Choose a deployment mode

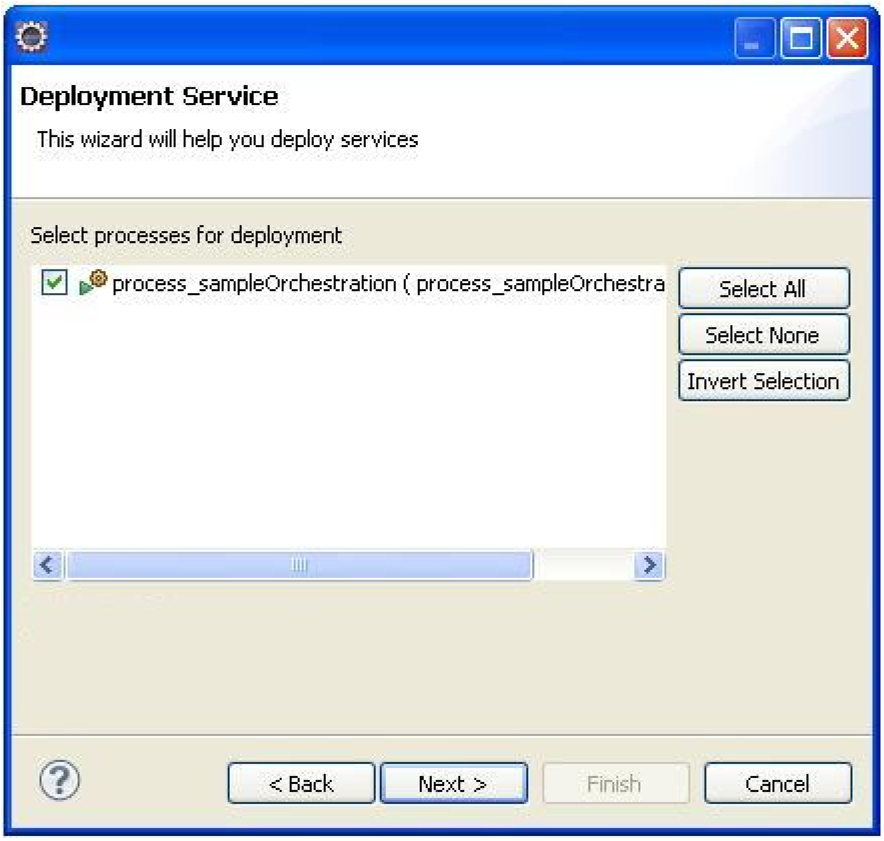
( ) Deploy services

(●) Deploy processes and related services

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.

creative commons COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/

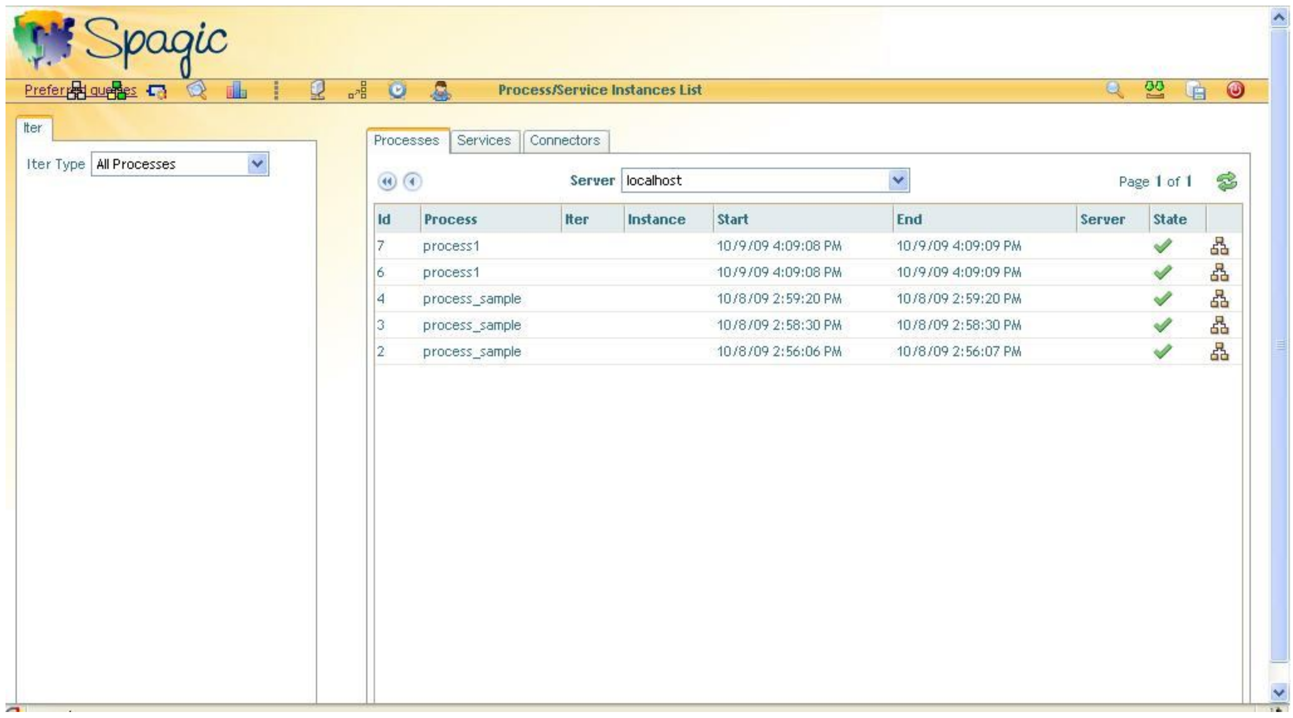

Deployment Service
This wizard will help you deploy services
Select processes for deployment
process_sampleOrchestration (process_sampleOrchestra
Select All
Select None
Invert Selection
< Back
Next >
Finish
Cancel

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.

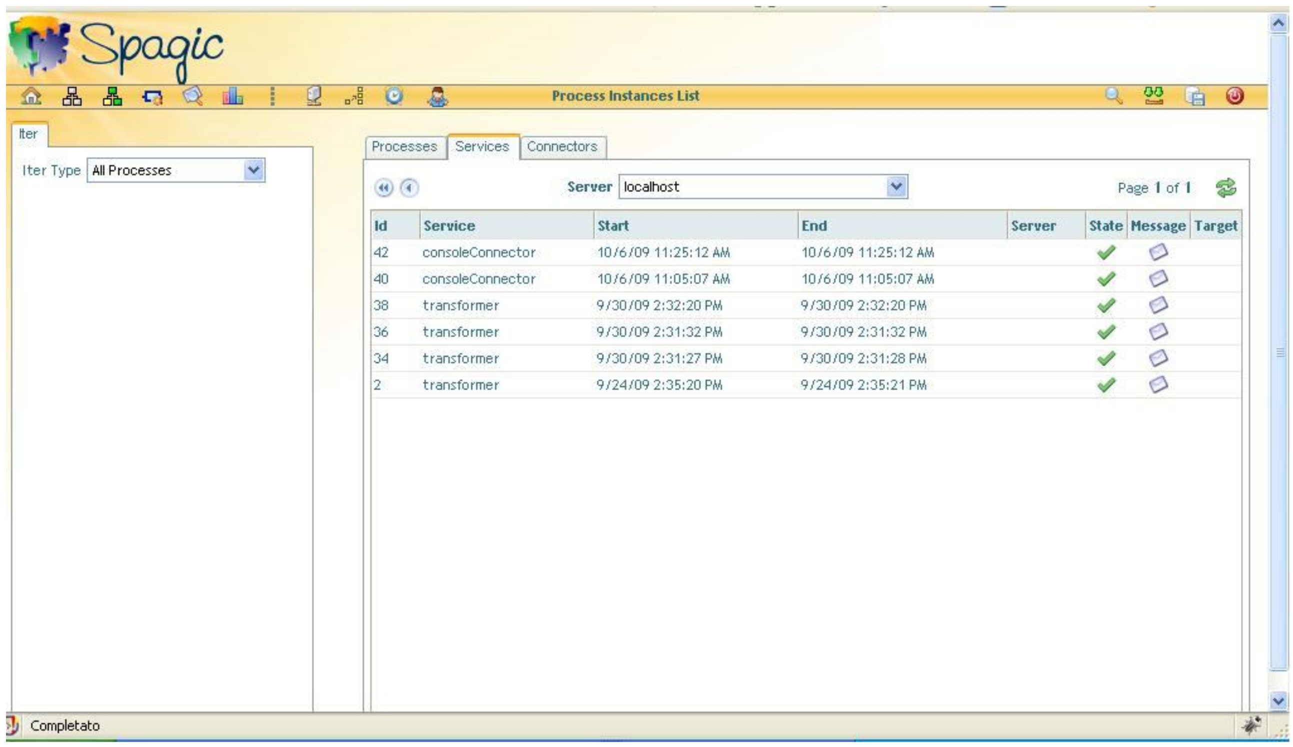
5.3 Test

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*.

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/



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		✓	



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		✓		

creative commons
COMMONS DEED

Engineering Ingegneria Informatica S.p.A., 2007, 2010. This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/