

# Flow of Work

User Documentation

# Table of Contents

1	Installation.....	3
1.1	Preconditions.....	3
1.1.1	Java Runtime.....	3
1.1.2	Set up Graphviz.....	3
1.2	Set up Eclipse and the FoW Plugins.....	3
2	Tutorials.....	4
2.1	A very simple process model.....	4
2.1.1	Create an empty project.....	4
2.1.2	Create a folder and a file for the model.....	6
2.1.3	Creating our first Model.....	8
3	Keyboard Shortcuts.....	12
3.1	Shortcuts for textual FoW editor.....	12

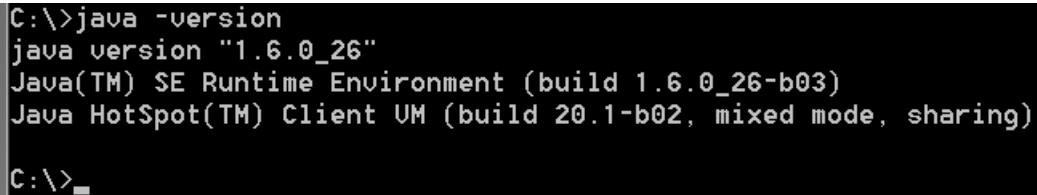
# 1 Installation

## 1.1 Preconditions

### 1.1.1 Java Runtime

Make sure a current Java Runtime (1.6 or higher) is installed to run Eclipse.

To check that, start a command shell (Windows: accessories -> shell) and execute the command "java -version".



```
C:\>java -version
java version "1.6.0_26"
Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java HotSpot(TM) Client VM (build 20.1-b02, mixed mode, sharing)
C:\>_
```

If there is no Java installed or the version is lower than 1.6, install JRE or JDK of Java Standard Edition (SE).

### 1.1.2 Set up Graphviz

Graphviz is open source graph visualization software and is used to create the diagrams for the process models. FoW codegenerators generate graphs as ASCII .dot files which are converted by the layouter dot into .jpg files.

1. Install the current version of graphviz: <http://www.graphviz.org/>
2. In order to be able to run the layouter, add the path to the program dot.exe (e.g. C:\Program Files (x86)\Graphviz 2.28\bin) to your environment variable "Path" (user).

If the path is set, the generated process graphs can be converted automatically from the dot format into jpg.

## 1.2 Set up Eclipse and the FoW Plugins

The Distribution of *Flow of Work* comes as a set of plugins for your Eclipse distribution.

1. Download a **Eclipse Kepler** distribution of the **Eclipse Modeling Tools**:  
<https://www.eclipse.org/downloads/>
2. Unpack the distribution to a folder on your harddisk
3. Start Eclipse: eclipse/eclipse.exe
4. Install Xtext SDK: Help -> Install New Software -> Work with: "Kepler - ..." -> Modeling -> Xtext SDK
5. Exit Eclipse
6. Copy the plugin files (\*.jar) of the folder  
<https://github.com/protossoftware/FlowOfWork/releases/<version>/plugins> to your eclipse dropins folder (e.g. eclipse/dropins)
7. Now you can start Eclipse (eclipse.exe) and select a folder for your workspace

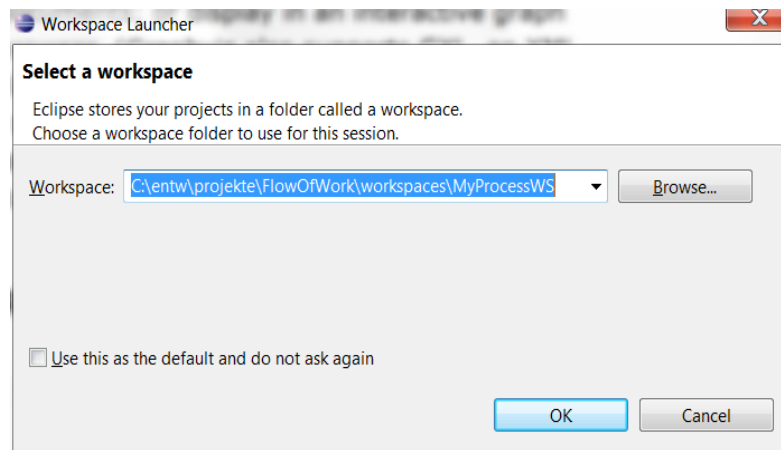
## 2 Tutorials

### 2.1 A very simple process model

We start with a minimal process model to introduce the basic features.

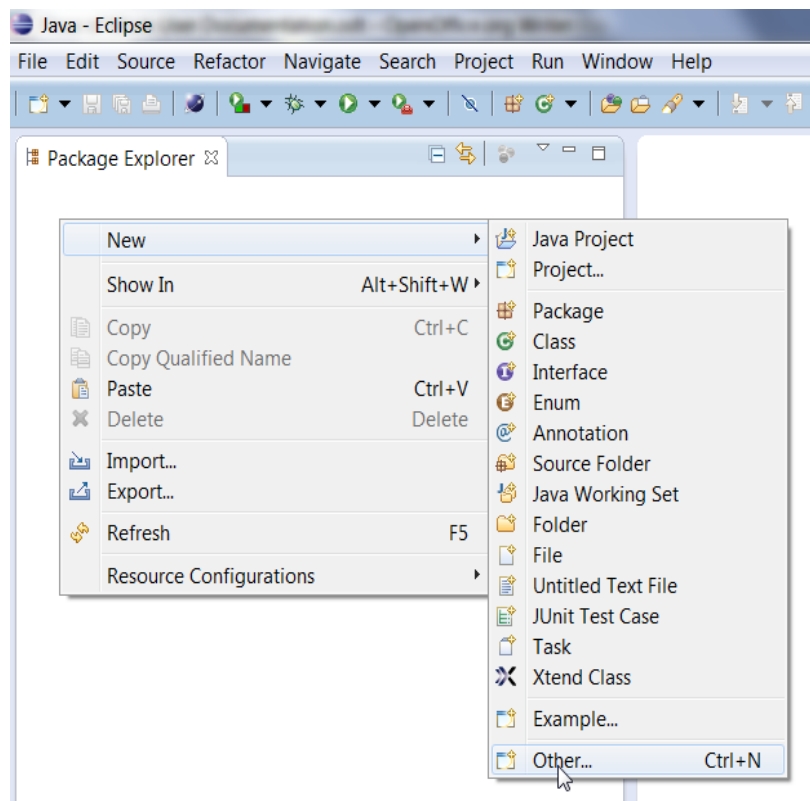
#### 2.1.1 Create an empty project

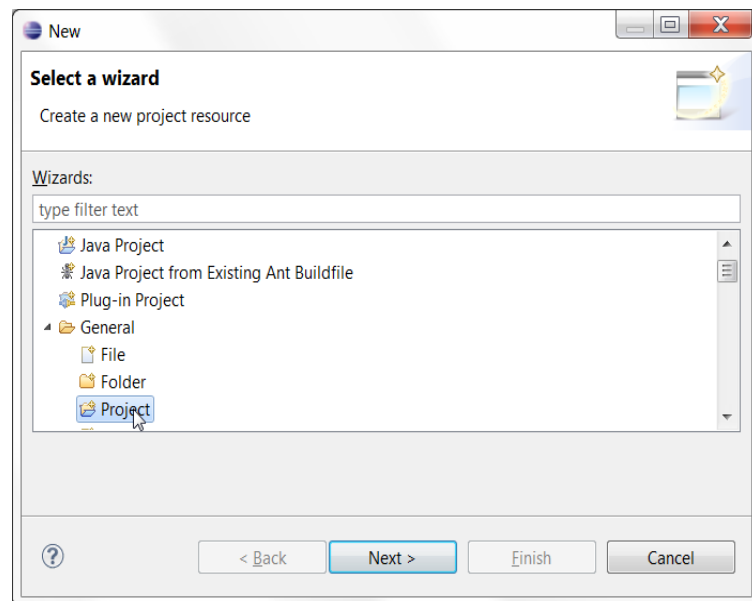
Start Eclipse and select an **empty** folder for your workspace, e.g. MyProcessWS.



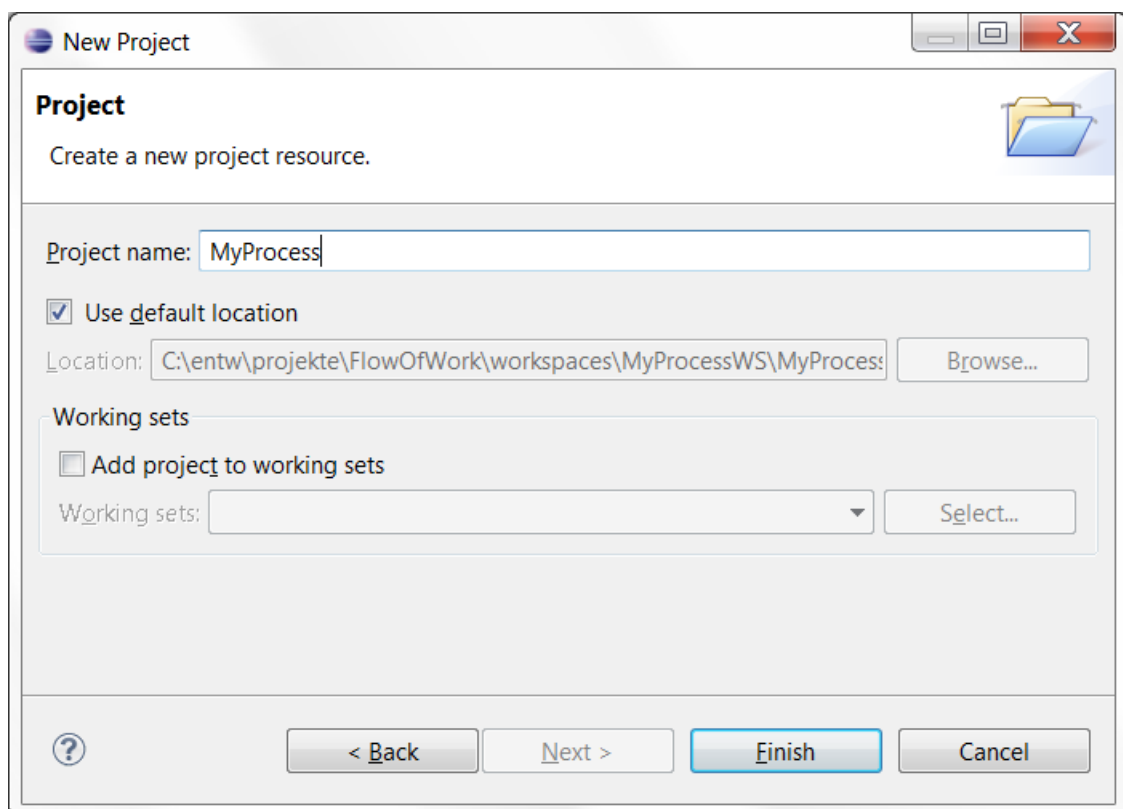
Create an empty project in the Package Explorer.

<right mouse click> -> New-> Other -> General -> Project





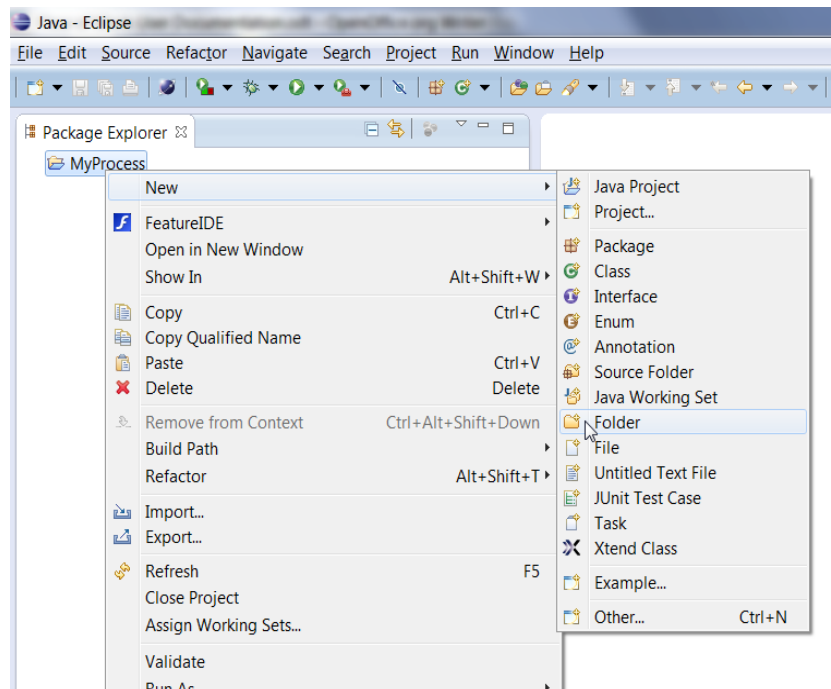
Define a Project Name, e.g. "MyProcess" and click Finish



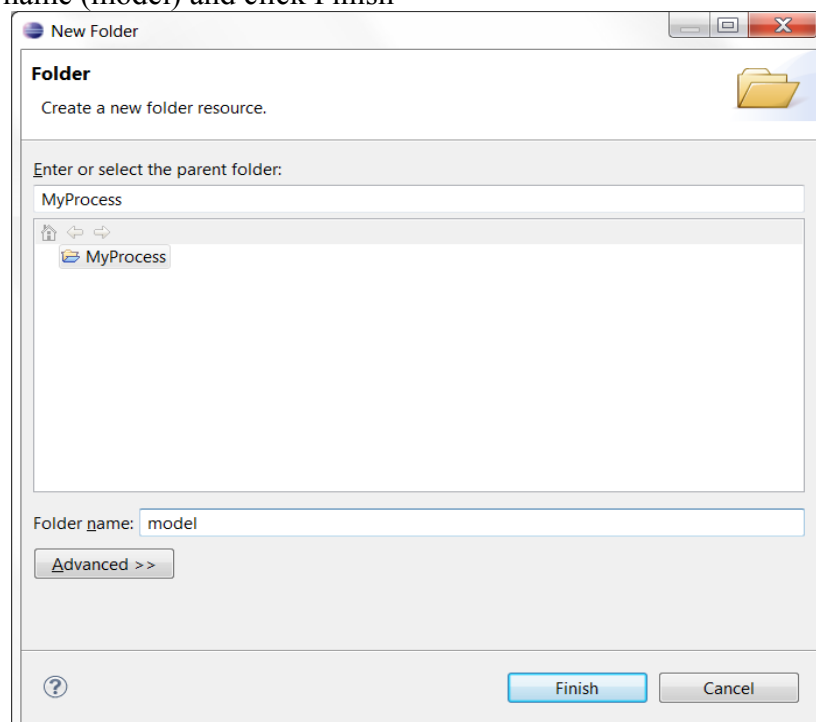
## 2.1.2 Create a folder and a file for the model

Create a new folder in the new project, named "model":

<right mouse click> -> New-> Folder

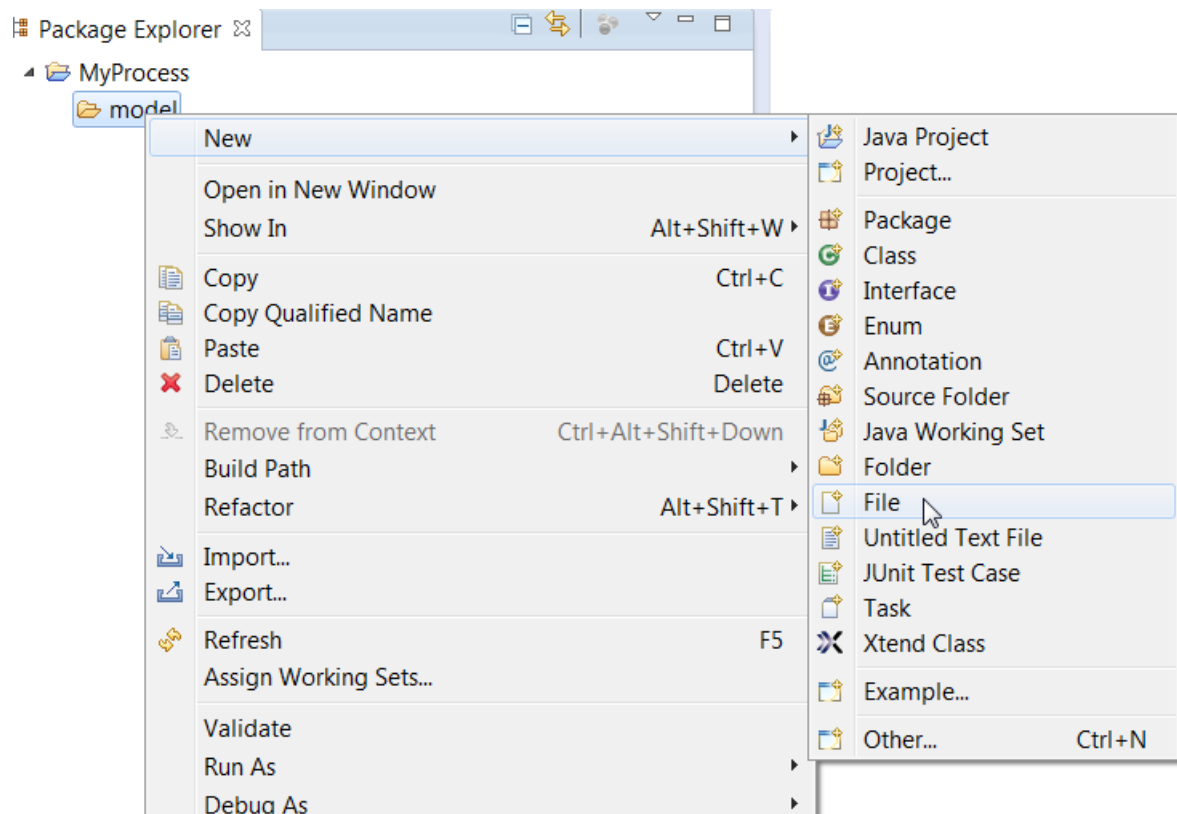


Define the folder name (model) and click Finish



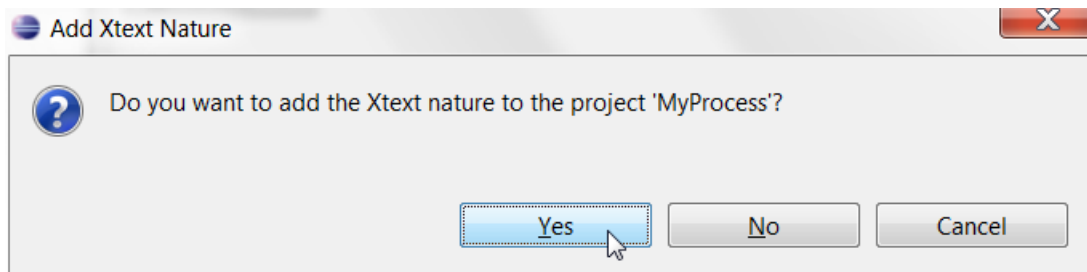
Create a new file in the model folder, named "MyProcess.fow":

<right mouse click> -> New-> File



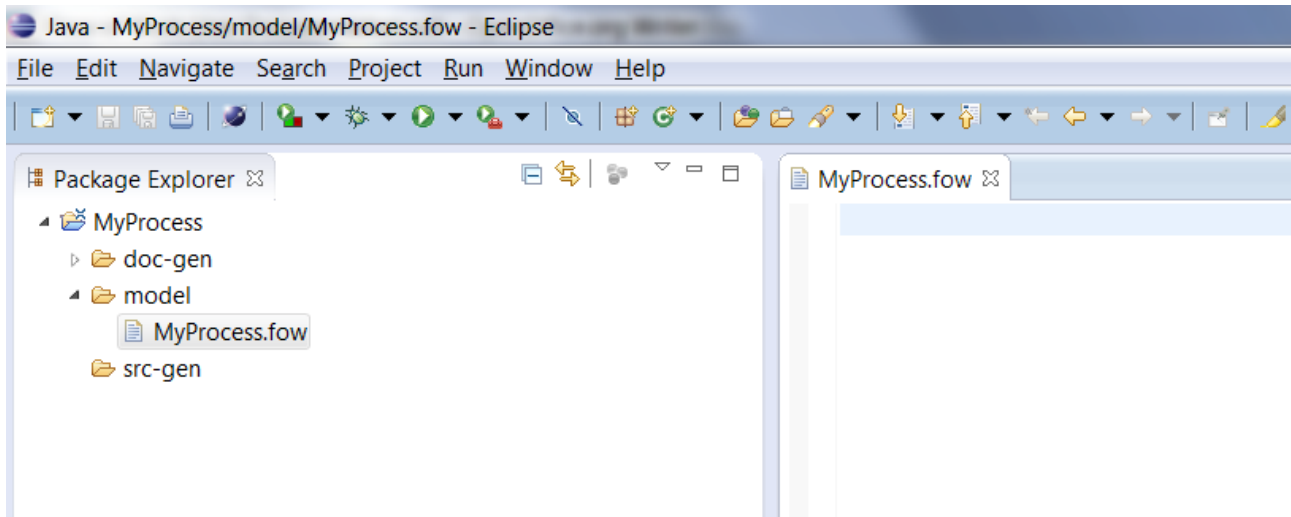
...

Make sure you add the Xtext nature with "Yes"



The empty file should open automatically in an editor, if not, doubleclick the file in the Package Explorer.

Result:

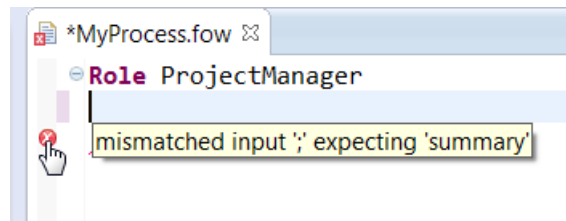


### 2.1.3 Creating our first Model

Now we create our first minimal process model. We start with the definition of a **Role**.

In the editor we enter the definition for a role. The syntax highlighting helps you doing it right. In the following example, you can see the error text by hovering with the mouse over the error marker.

The Role needs at least a summary text.

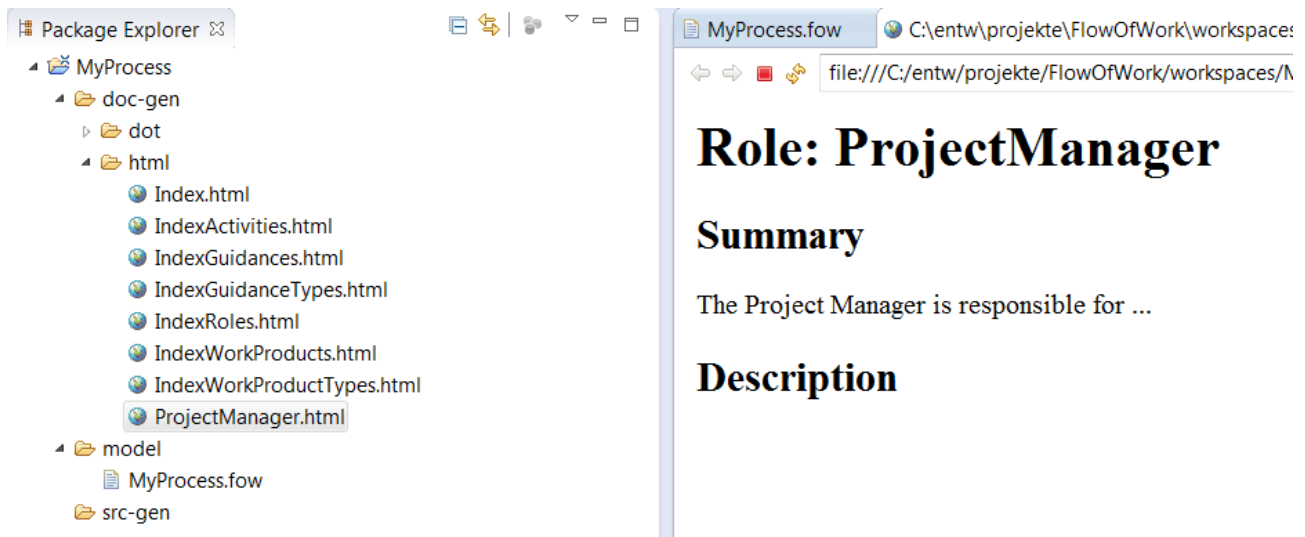


Here the correct model:

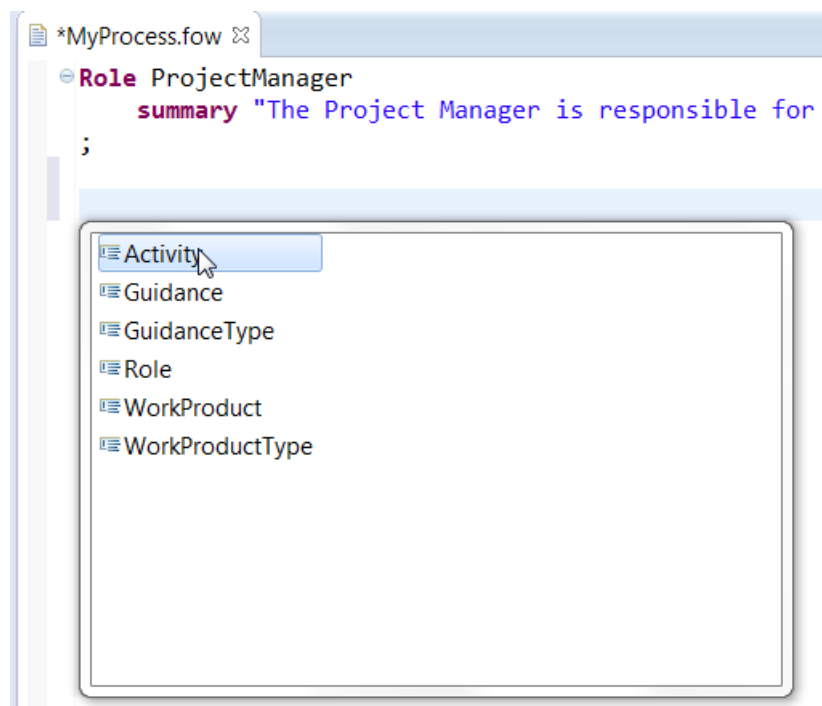
```
Role ProjectManager
    summary "The Project Manager is responsible for ..."
;
```

Save the model (shortcut: <ctrl>+s). When ever a model is saved the codegenerator updates the generated documentation, including all folders and index files. Look at the generated html documentation by double clicking doc-gen/html/ProjectManager.html. The file Index.html is the entry point to browse the complete documentation. The other Index... files contain a list of all Elements of a certain Type (e.g. IndexRoles.html contains a list for all Roles with the links to the generated html pages).

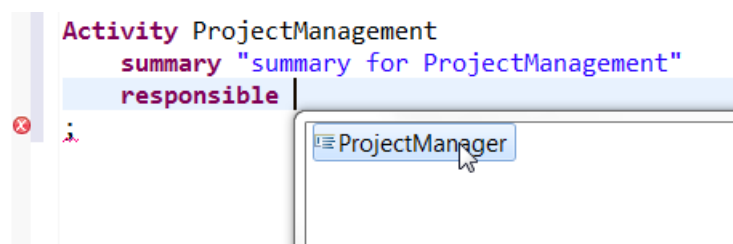




Now we want to create an Activity. You can use <ctrl>+<space> to see which model elements can be used at this position of the model. Select the Activity.



Complete the minimal definition of the Activity, using the code completion. After the keyword "responsible" press <ctrl>+<space> and select a role (only one available in our small model).



And here the complete model:

```
Role ProjectManager
    summary "The Project Manager is responsible for ..."
;

Activity ProjectManagement
    summary "summary for ProjectManagement"
    responsible ProjectManager
;
```

Browse the doc-gen/html folder and have a look at the generated Activity file (ProjectManagement.html):

**Activity: ProjectManagement**

<b>Purpose:</b>	summary for ProjectManagement		
<b>Scope:</b>	Where does the scope come from?		
<b>Inputs:</b>	no inputs	<b>Outputs:</b>	no outputs
<b>Entry Criteria:</b>	really needed???	<b>Exit Criteria:</b>	really needed???
<b>Responsible:</b>	ProjectManager		
<b>Description:</b>			

**Activities and Responsibilities**

**Guidelines**

**SPEM Diagram**

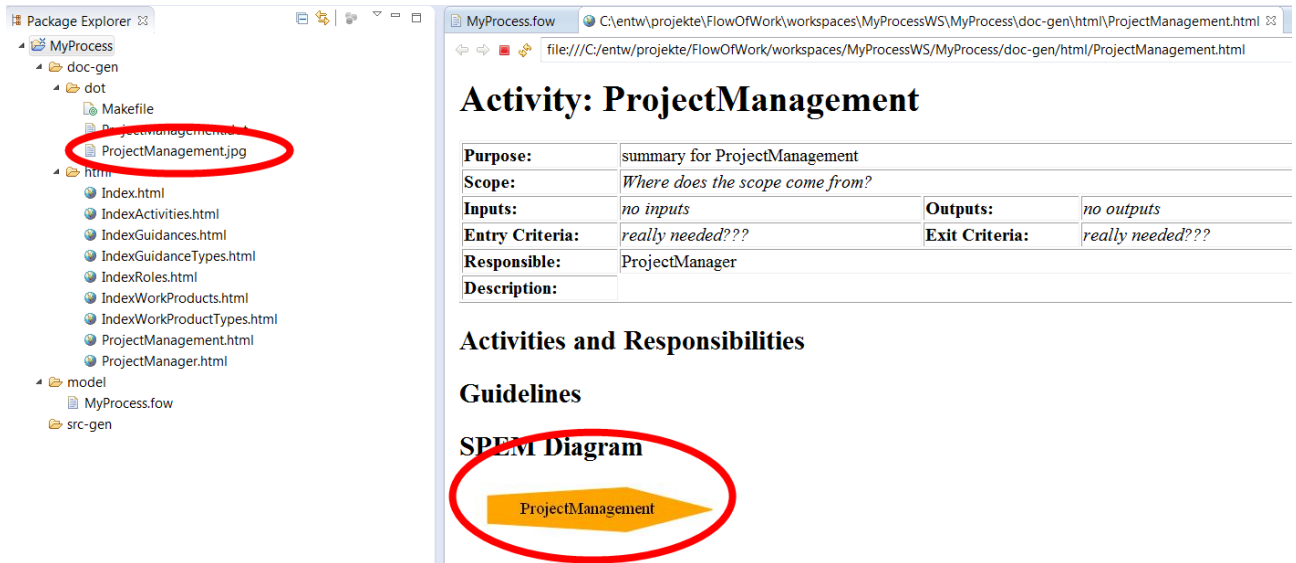
✖ SPEM Diagram: ProjectManagement

As you can see the SPEM Diagram is not yet created. In order to do that, open a **command prompt** (german: **Eingabeaufforderung**) and change the working directory to the dot folder of your project (e.g. C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot). The type make and the dot command creates the jpg files. In further versions, this call will be integrated into the generation process. Result in the command prompt:

```
C:\entw\projekte\FlowOfWork\workspaces>cd C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot
C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot>make
dot -Tjpg ProjectManagement.dot -o ProjectManagement.jpg
C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot>
```

Mark the project name in the Project Explorer and press <F5> to update your project with the new file and click into your html viewer for ProjectManagement.html and also update with <F5>.

Result in the eclipse project:



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure for 'MyProcess'. The 'doc-gen' folder is expanded, showing a 'dot' sub-folder containing 'ProjectManagement.jpg', which is circled in red. Other files in the 'html' folder include 'Index.html', 'IndexActivities.html', 'IndexGuidances.html', 'IndexRoles.html', 'IndexWorkProducts.html', 'IndexWorkProductTypes.html', 'ProjectManagement.html', and 'ProjectManager.html'. The main editor window shows the content of 'ProjectManagement.html'.


## Activity: ProjectManagement

<b>Purpose:</b>	summary for ProjectManagement		
<b>Scope:</b>	Where does the scope come from?		
<b>Inputs:</b>	no inputs	<b>Outputs:</b>	no outputs
<b>Entry Criteria:</b>	really needed???	<b>Exit Criteria:</b>	really needed???
<b>Responsible:</b>	ProjectManager		
<b>Description:</b>			

### Activities and Responsibilities

### Guidelines

### SPEM Diagram



The diagram shows a yellow arrow pointing to the right, labeled 'ProjectManagement'. The arrow and its label are circled in red.

## 3 Keyboard Shortcuts

### 3.1 *Shortcuts for textual FoW editor*

- <Shift> + <Ctrl> + F: Format File
- <Ctrl> + <SPACE>: Code Completion
- <F3>: Open Declaration (jump to Declaration)
- <Shift> + <Alt> + R: Rename Refactoring
- <Strg> + D: Remove Line