Developing a basic EPF plugin from scratch

By Roger Champagne, Eng., Ph. D.
Department of Software and IT Engineering
École de technologie supérieure, Montréal, Canada
Last update: 2010-Aug-11

Introduction

This tutorial aims at sharing my experience with the development of an EPF plugin. While there are many tutorials on the web that describe how to extend an existing EPF plugin (for example OpenUP), I have found that there is very little information on how to create an EPF plugin from scratch. This is what is described in this document.

At the end of the tutorial, you will have a published EPF plugin with skeletal content, namely:

- one role, named MyRole;
- one work product, named MyWorkProduct;
- one task, named MyTask, which is performed by MyRole and has MyWorkProduct as output;

The tutorial omits (by design) such details as:

- why things are done as they are (basically, it is tricky, and it works!);
- the definitions of EPF concepts (see the EPF Composer help for that);
- a lot of content (description, etc) of the content elements that are created, because that's the easy part, and well described elsewhere.

The two key issues in developing an EPF plugin from scratch are the following:

- creating the various elements (library, configuration, plugin, content package, ...) **in the right order** and putting things **in the right place**;
- defining **custom categories** to control what is published and how things are displayed and grouped in the tree view.

This tutorial is inspired by this document, which I initially found at http://www-users.cs.york.ac.uk/~malihetb/Publication/Eclipse-process-Framework-Step-by-step-example.pdf.

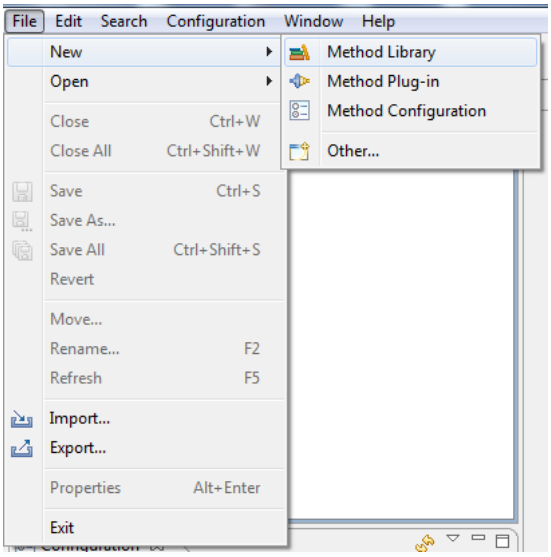
Prerequisite

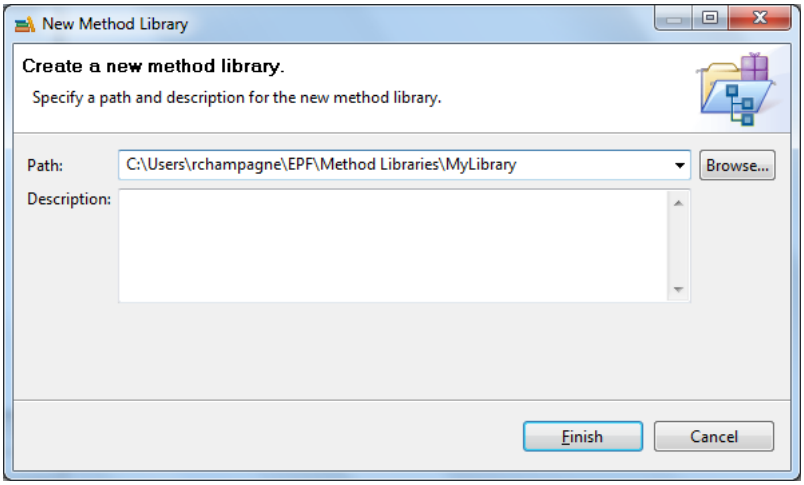
- You have downloaded and installed EPF Composer (this tutorial was produced with version 1.5.0.4)
- You are minimally familiar with working in the Eclipse IDE

Creating the plugin content

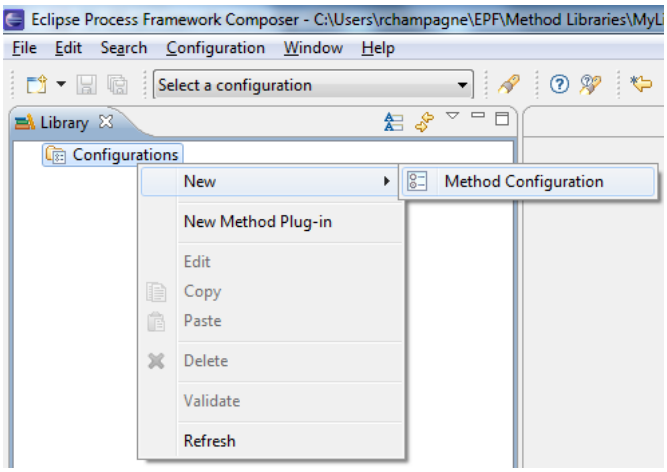
1. Start EPF Composer, and switch to the Authoring perspective, if not already there.

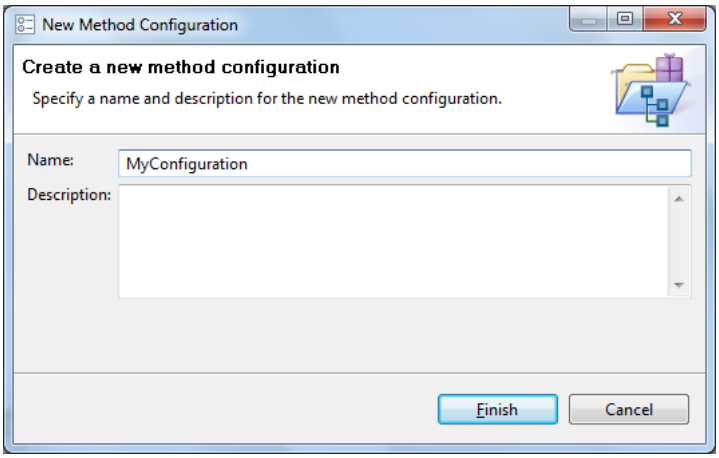
2. Create a new method library called MyLibrary by:
   - selecting File→New→Method Library;
   - replacing the default library name by MyLibrary;
   - pressing the Finish button.



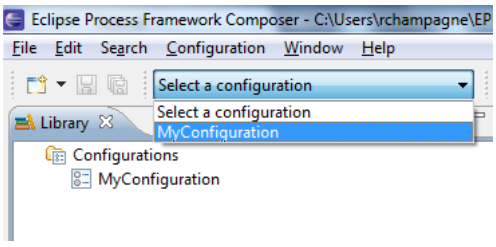


3. Create a new configuration called MyConfiguration by:
   - right-clicking on Configurations and selecting New→Method Configuration;
   - replacing the default configuration name by MyConfiguration;
   - pressing the Finish button.

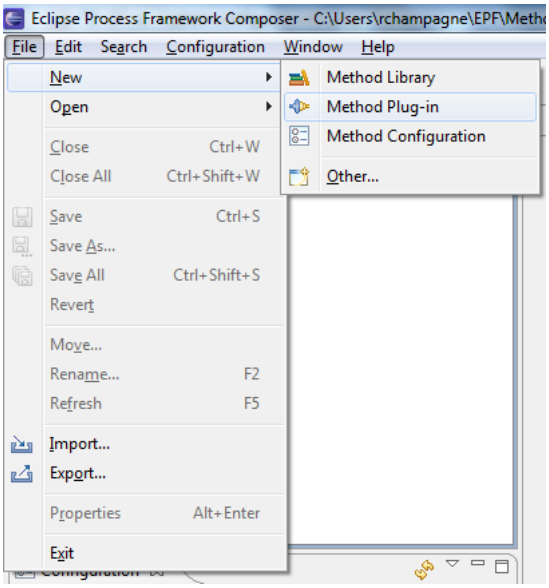


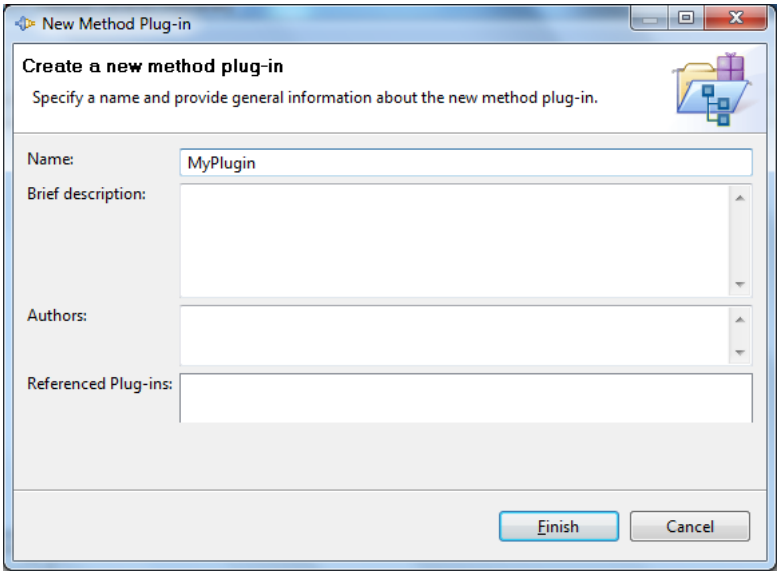


4. Select the new configuration by:
   - accessing the "Select a configuration" pull-down;
   - selecting MyConfiguration.

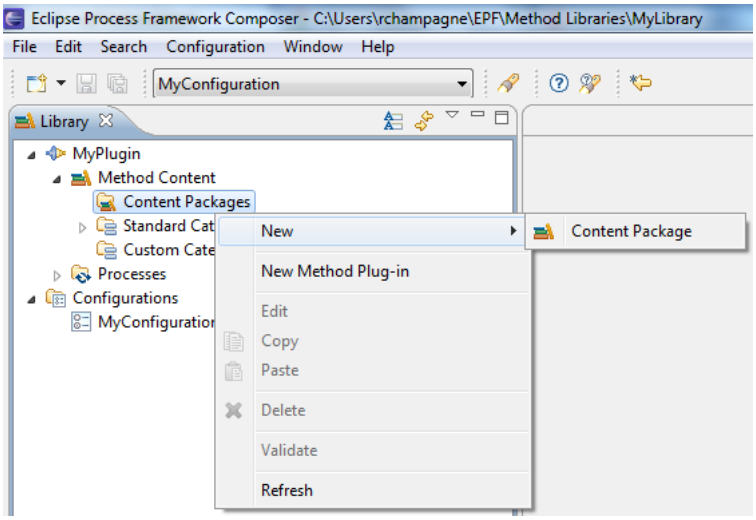


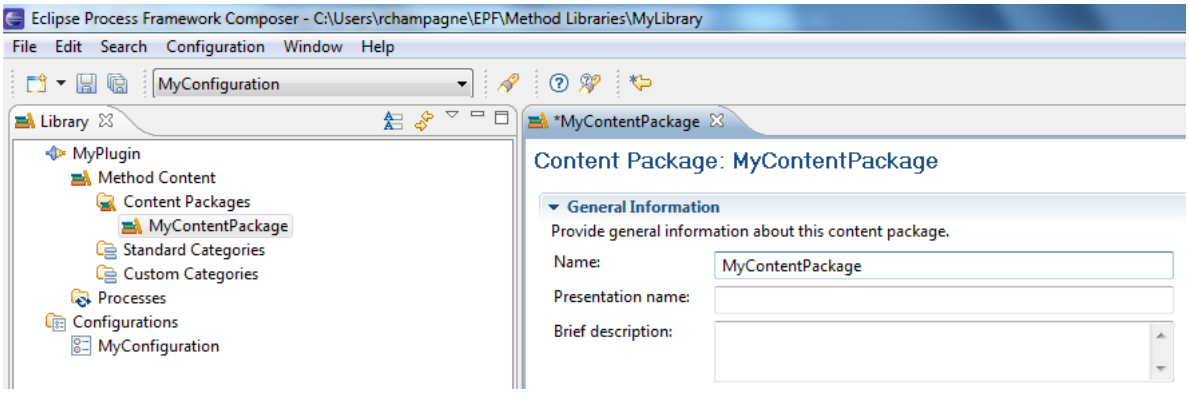
5. Create a new method plugin called MyPlugin by:
   - selecting File→New→Method Plug-in;
   - replacing the default plugin name by MyPlugin;
   - pressing the Finish button.

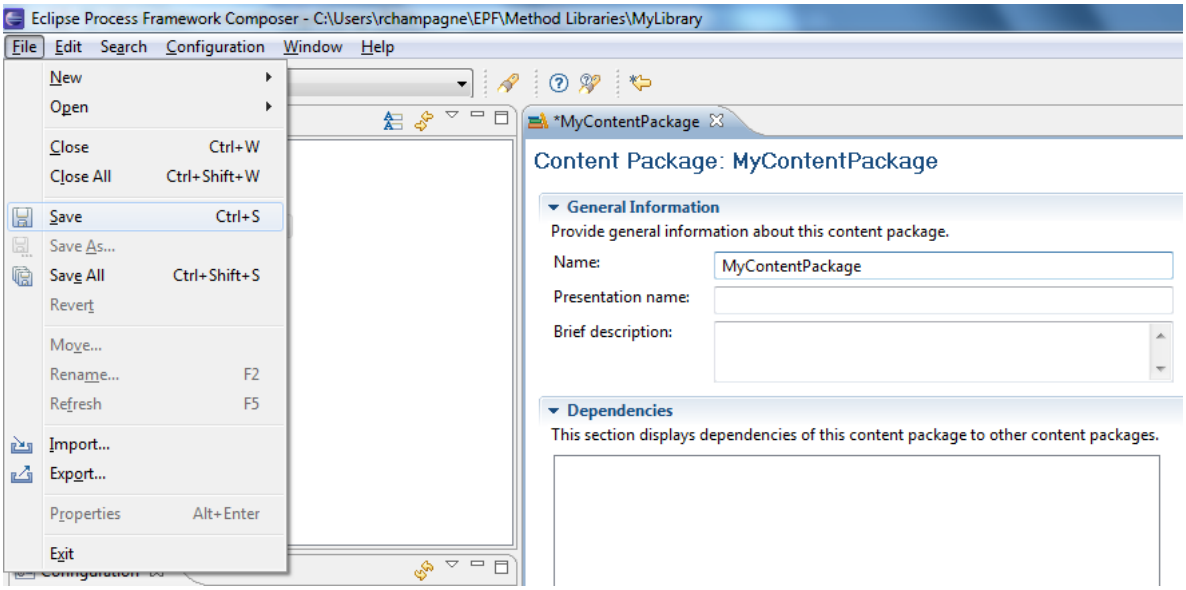




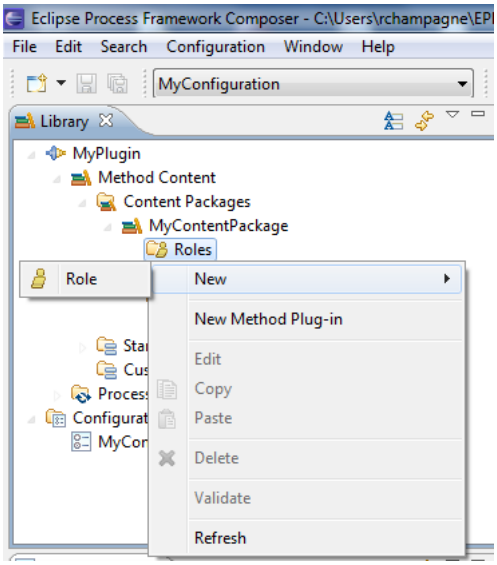
6. Create a new content package called MyContentPackage in MyPlugin by:
   - right-clicking on Content Packages and selecting New→Content Package;
   - replacing the default name by MyContentPackage;
   - selecting File→Save.

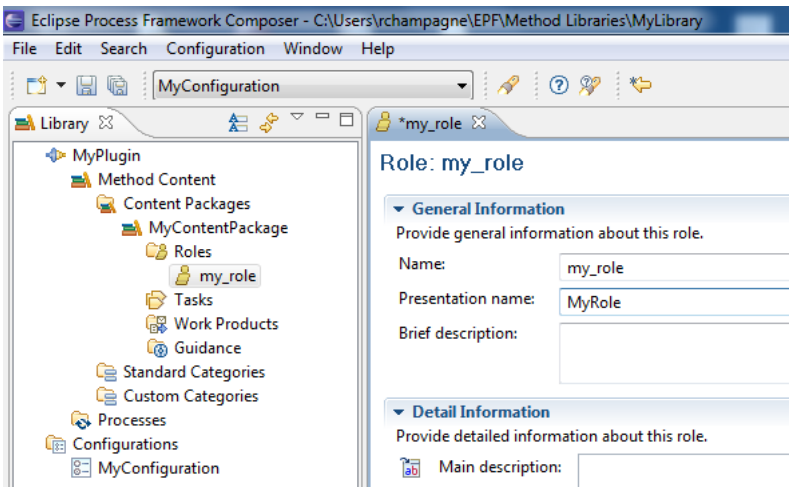


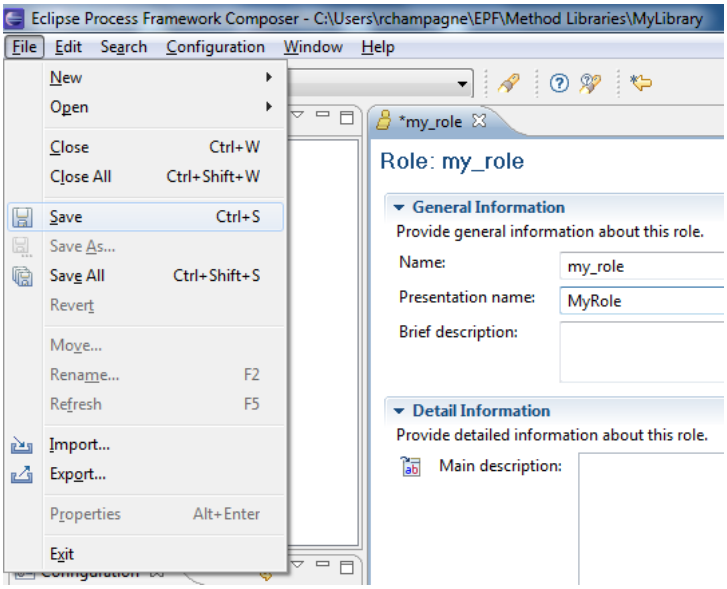




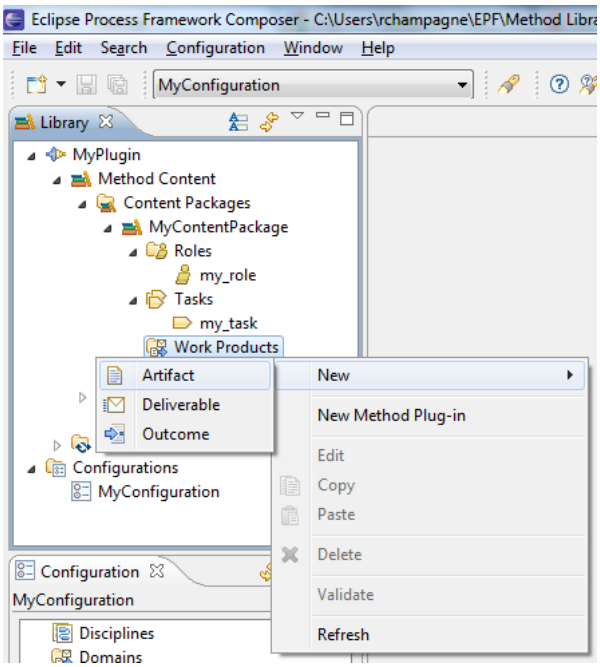
7. Add a role called MyRole in MyContentPackage by:
   - right-clicking on Roles and selecting New→Role;
   - replacing the default Name by my_role and the default Presentation name by MyRole;
     - NOTE: EPF uses the Name field as a pointer to the elements, and the Presentation name as a (typically better formatted for humans) display name. You can use pretty much anything in the Presentation name (spaces, accented characters, ...) but keep the Name field free of any non-standard characters to avoir problems.
   - selecting File→Save.

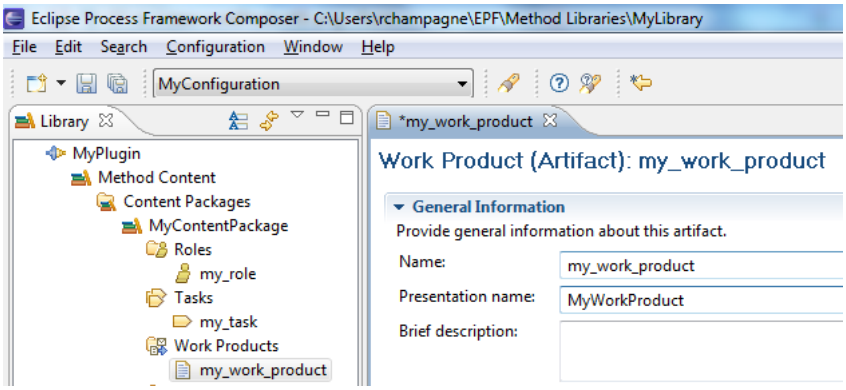


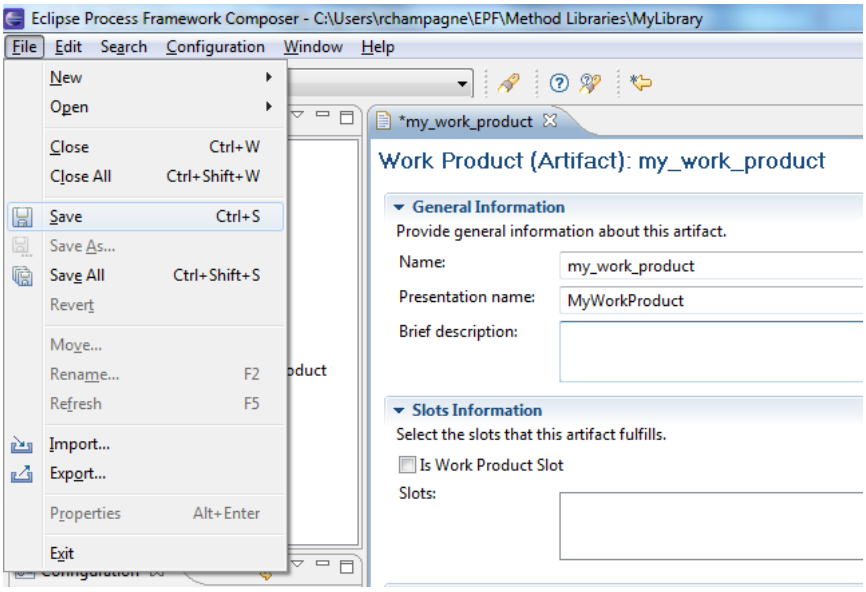




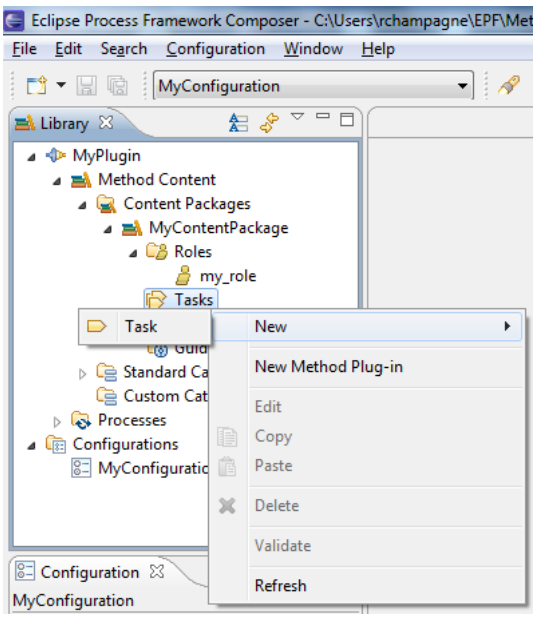
8. Add a work product called MyWorkProduct in MyContentPackage by:
   - right-clicking on Work Products and selecting New→Artifact;
   - replacing the default Name by my_work_product and the default Presentation name by MyWorkProduct;
   - selecting File→Save.

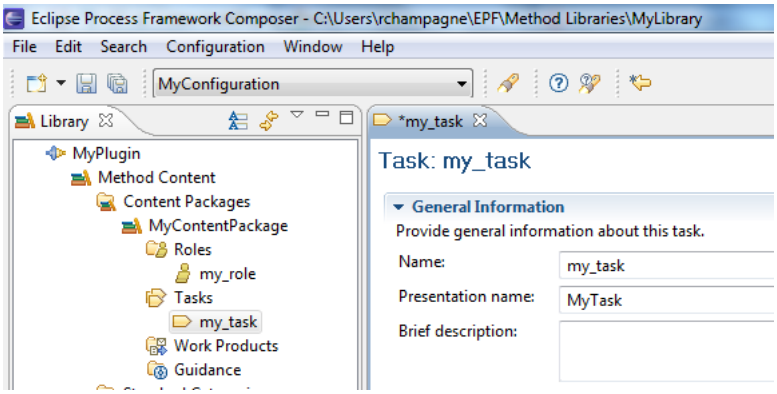


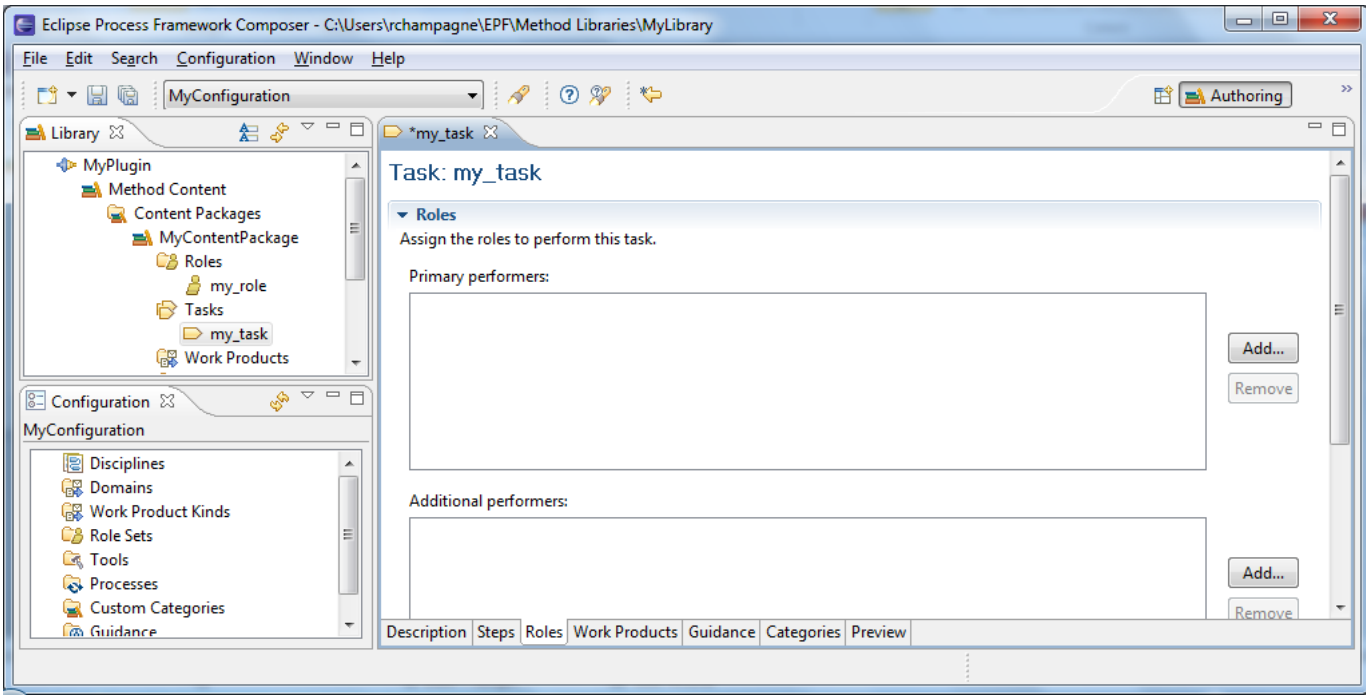


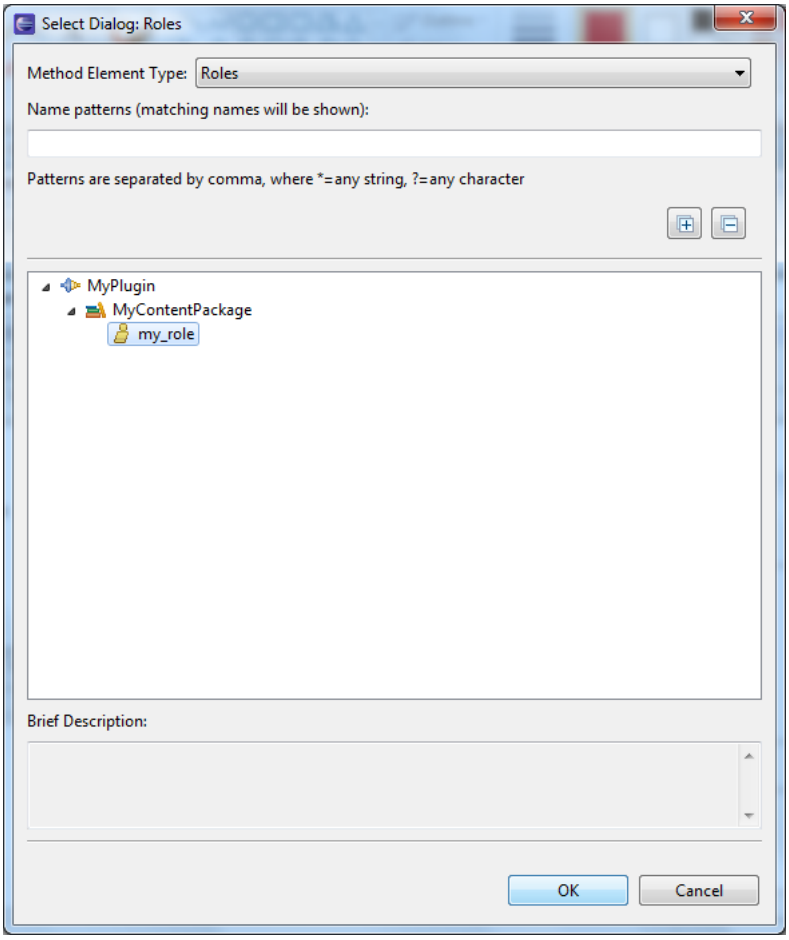


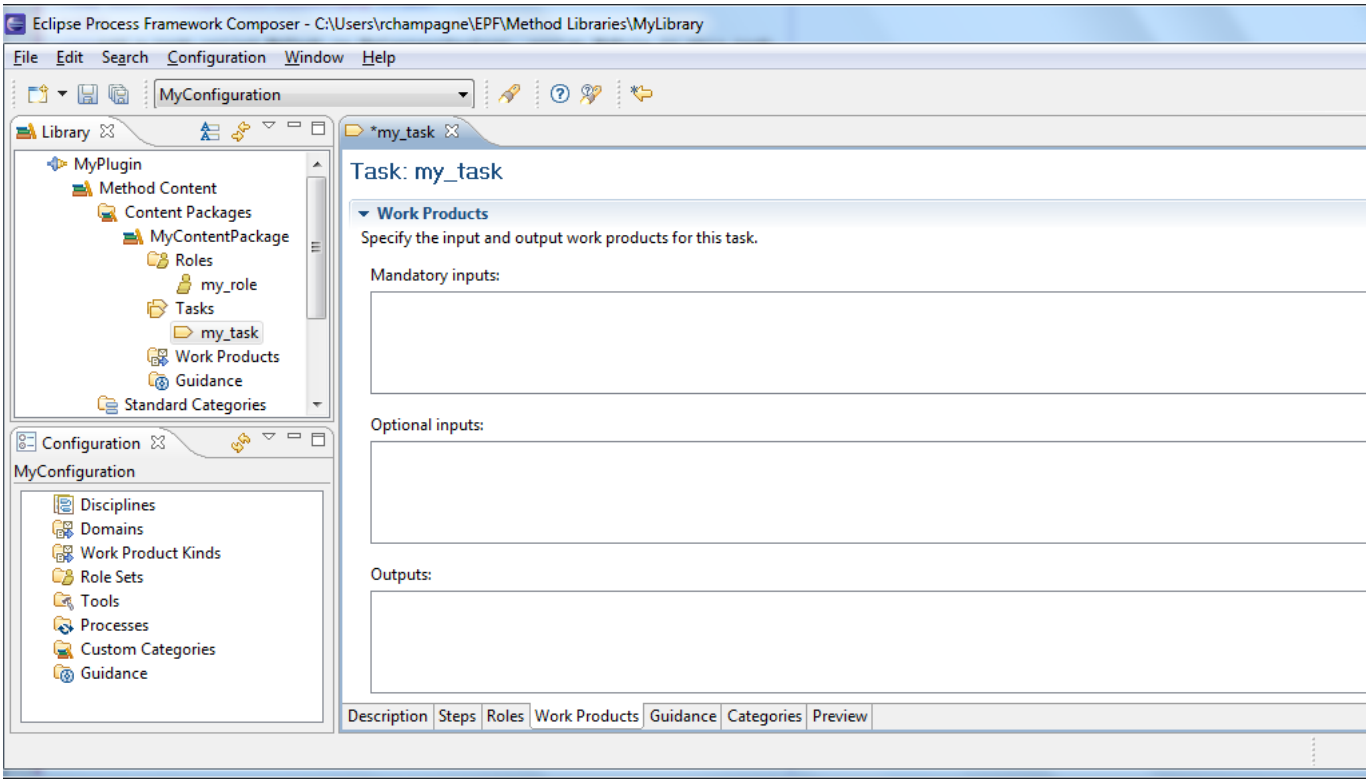
9. Add a task called MyTask in MyContentPackage, assign MyRole to this task, and define MyWorkProduct as the output of this task by:
   - right-clicking on Tasks and selecting New→Task;
   - replacing the default Name by my_task and the default Presentation name by MyTask;
   - selecting the Roles tab, and clicking the Add button to the right of the Primary performers section;
   - selecting Roles at the top of the new window, selecting my_role in the bottom portion, and pressing the OK button;
   - selecting the Work Products tab, and clicking the Add button to the right of the Outputs section;
   - selecting Work Products at the top of the new window, selecting my_work_product in the bottom portion, and pressing the OK button;
   - selecting File→Save.

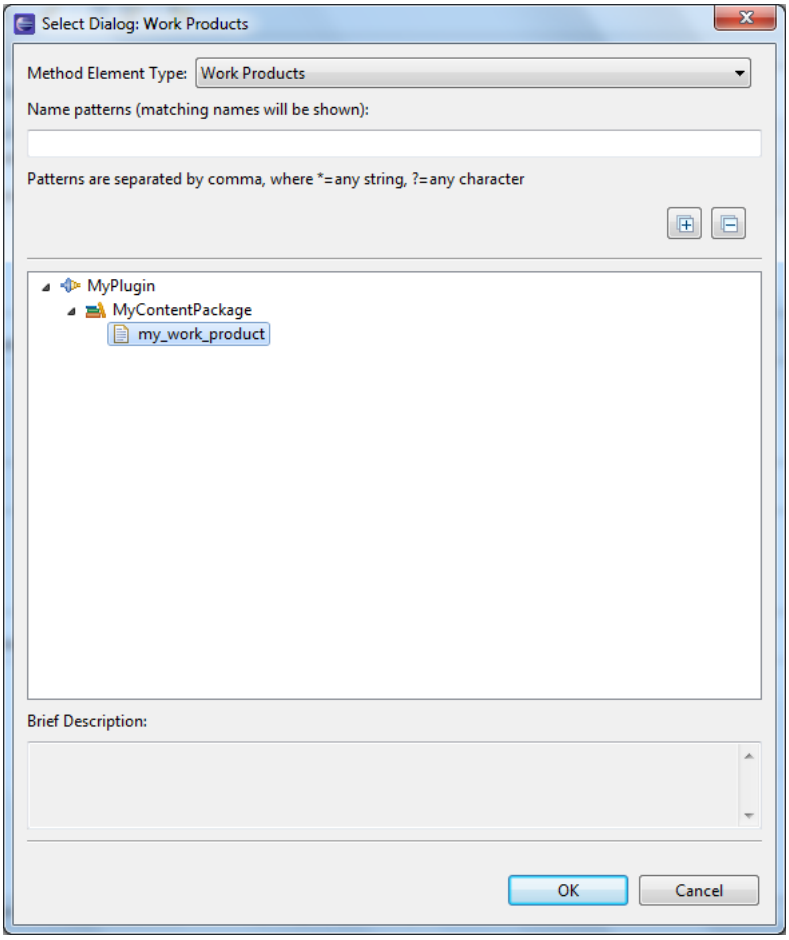


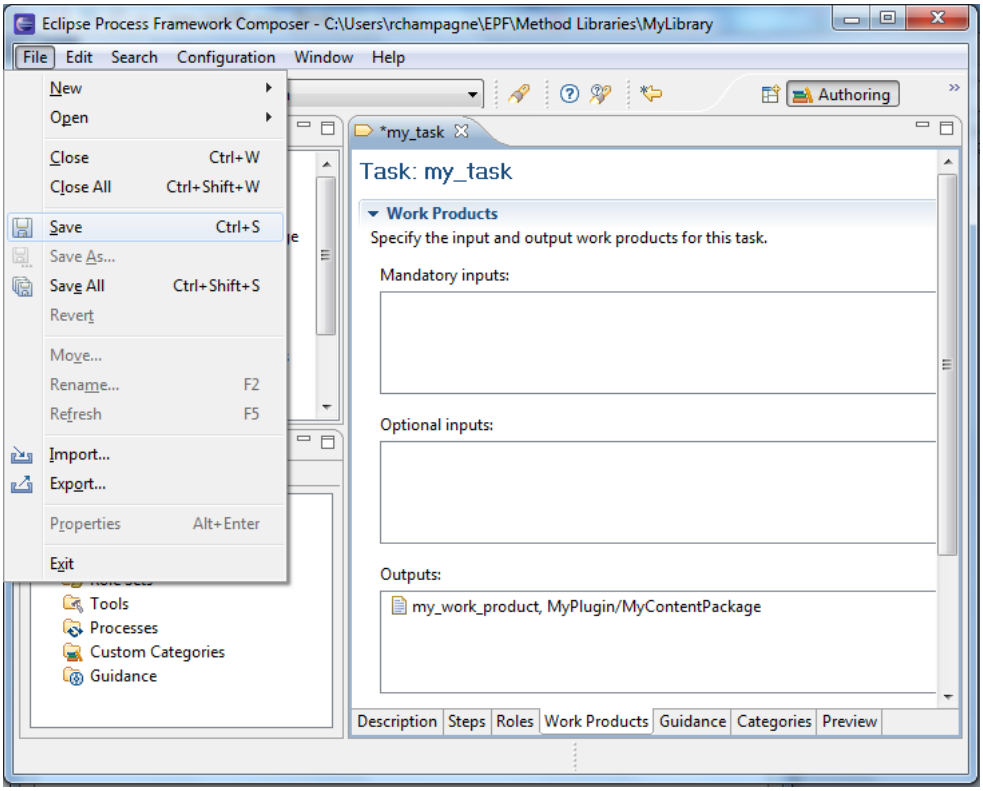

Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Libraries\MyLibrary
File Edit Search Configuration Window Help
MyConfiguration
Library
MyPlugin
Method Content
Content Packages
MyContentPackage
Roles
my_role
Tasks
my_task
Work Products
Guidance
*my_task
Task: my_task
General Information
Provide general information about this task.
Name:
my_task
Presentation name:
MyTask
Brief description:


Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Libraries\MyLibrary
File Edit Search Configuration Window Help
MyConfiguration
Authoring
Library
MyPlugin
Method Content
Content Packages
MyContentPackage
Roles
my_role
Tasks
my_task
Work Products
Configuration
MyConfiguration
Disciplines
Domains
Work Product Kinds
Role Sets
Tools
Processes
Custom Categories
Guidance
*my_task
Task: my_task
Roles
Assign the roles to perform this task.
Primary performers:
Add...
Remove
Additional performers:
Add...
Remove
Description
Steps
Roles
Work Products
Guidance
Categories
Preview


Select Dialog: Roles
Method Element Type: Roles
Name patterns (matching names will be shown):
Patterns are separated by comma, where *=any string, ?=any character
MyPlugin
MyContentPackage
my_role
Brief Description:
OK
Cancel

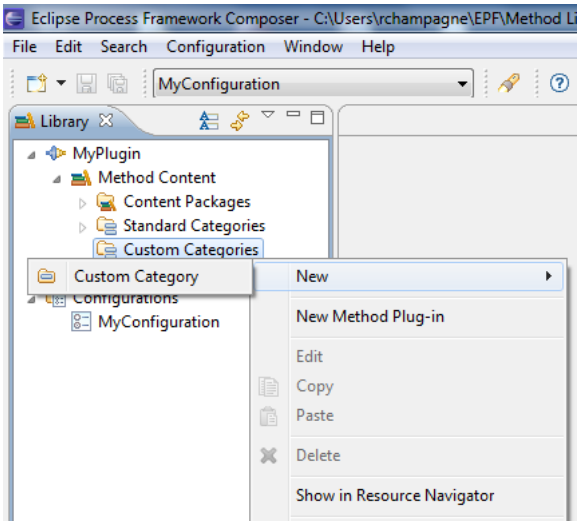

Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Libraries\MyLibrary
File Edit Search Configuration Window Help
MyConfiguration
Library
MyPlugin
Method Content
Content Packages
MyContentPackage
Roles
my_role
Tasks
my_task
Work Products
Guidance
Standard Categories
Configuration
MyConfiguration
Disciplines
Domains
Work Product Kinds
Role Sets
Tools
Processes
Custom Categories
Guidance
*my_task
Task: my_task
Work Products
Specify the input and output work products for this task.
Mandatory inputs:
Optional inputs:
Outputs:
Description Steps Roles Work Products Guidance Categories Preview

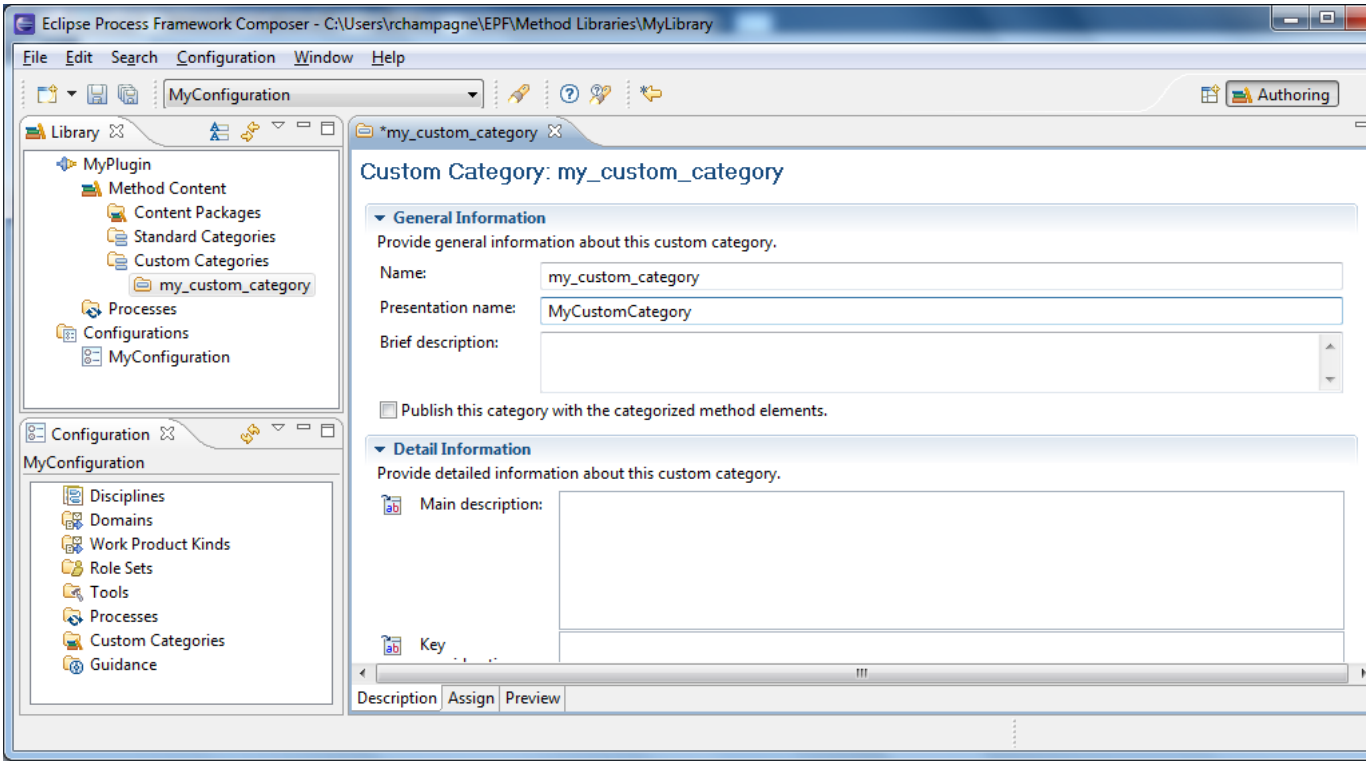


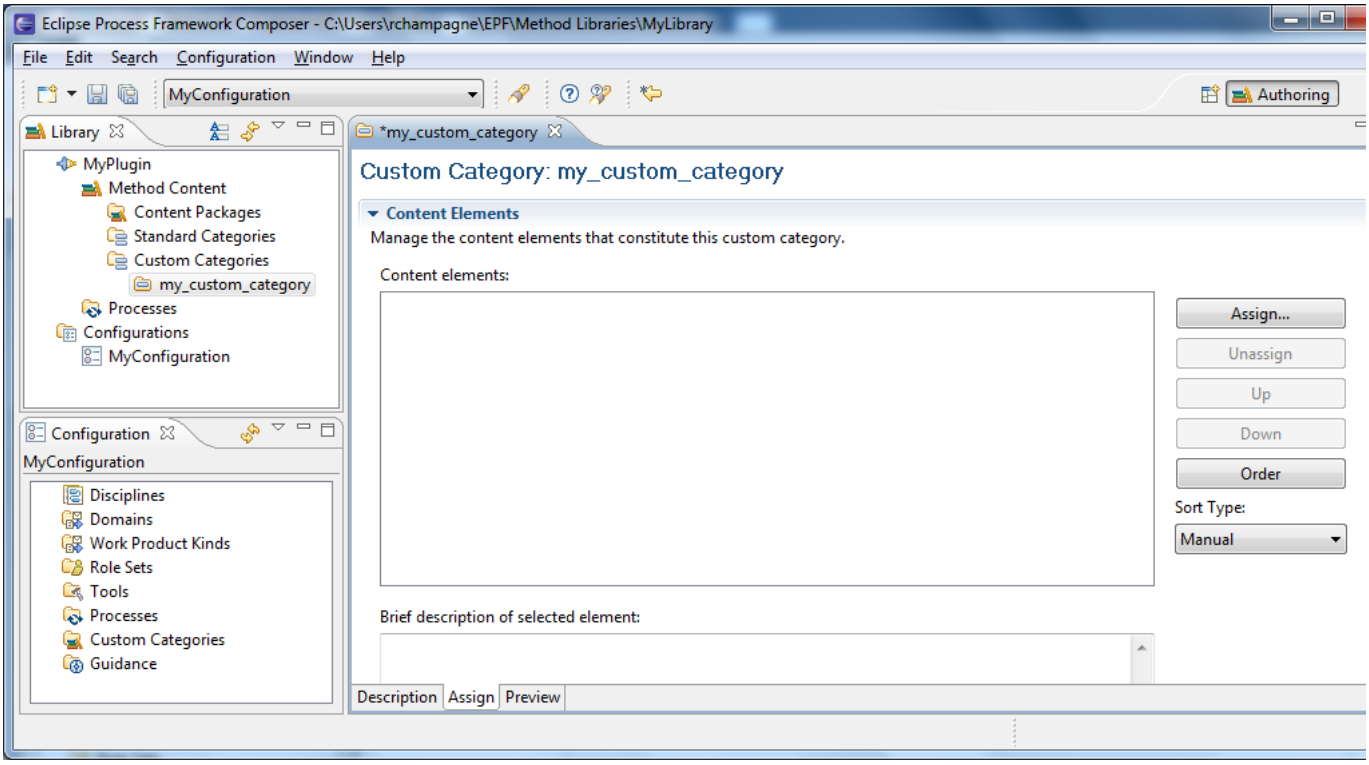


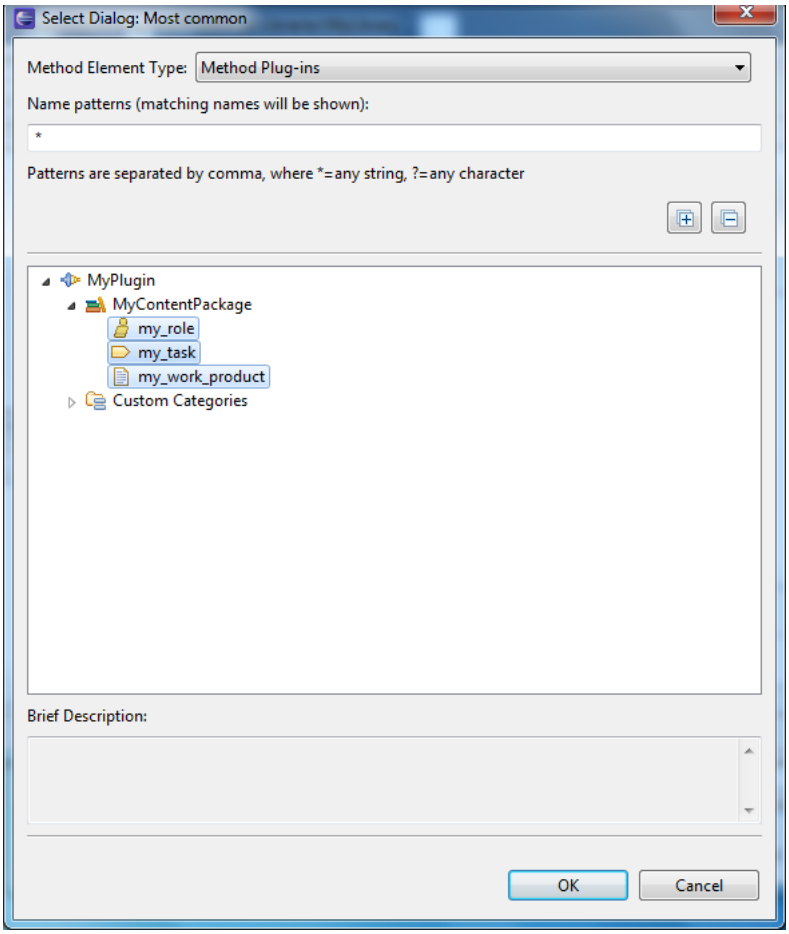
Preparing your content for publication

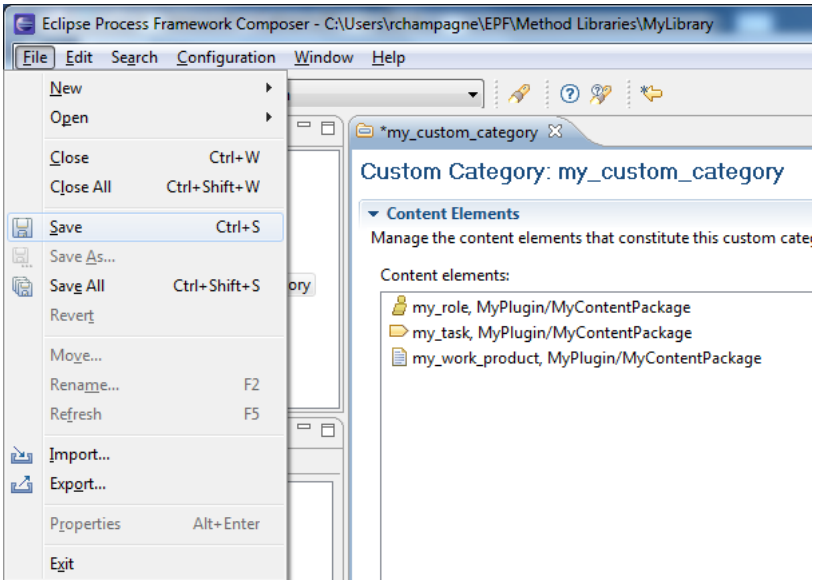
1. Create a custom category called MyCustomCategory and assign MyRole, MyTask and MyWorkProduct to MyCustomCategory by:
   - right-clicking on Custom Categories and selecting New→Custom Category;
   - replacing the default Name by my_custom_category and the default Presentation name by MyCustomCategory;
   - selecting the Assign tab and pressing the Assign button to the right of the Content elements section;
   - selecting my_role, my_task, and my_work_product, and pressing the OK button;
   - selecting File→Save.


Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Li
File Edit Search Configuration Window Help
MyConfiguration
Library
MyPlugin
Method Content
Content Packages
Standard Categories
Custom Categories
Custom Category
Configurations
MyConfiguration
New
New Method Plug-in
Edit
Copy
Paste
Delete
Show in Resource Navigator

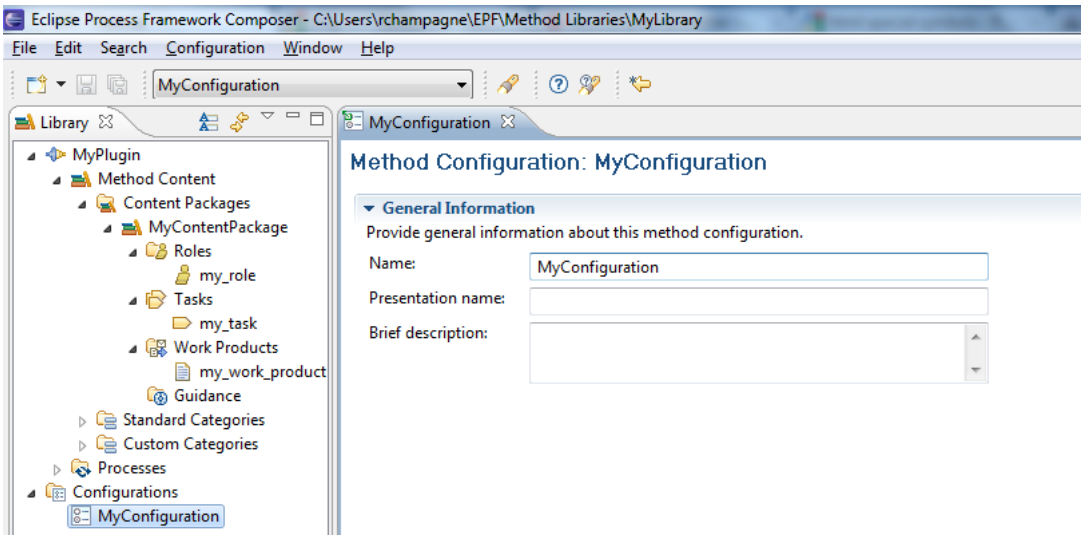

Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Libraries\MyLibrary
File Edit Search Configuration Window Help
MyConfiguration
Authoring
Library
MyPlugin
Method Content
Content Packages
Standard Categories
Custom Categories
my_custom_category
Processes
Configurations
MyConfiguration
Configuration
MyConfiguration
Disciplines
Domains
Work Product Kinds
Role Sets
Tools
Processes
Custom Categories
Guidance
*my_custom_category
Custom Category: my_custom_category
General Information
Provide general information about this custom category.
Name:
my_custom_category
Presentation name:
MyCustomCategory
Brief description:
Publish this category with the categorized method elements.
Detail Information
Provide detailed information about this custom category.
Main description:
Key
Description
Assign
Preview


Eclipse Process Framework Composer - C:\Users\rchampagne\EPF\Method Libraries\MyLibrary
File Edit Search Configuration Window Help
MyConfiguration
Authoring
Library
MyPlugin
Method Content
Content Packages
Standard Categories
Custom Categories
my_custom_category
Processes
Configurations
MyConfiguration
Configuration
MyConfiguration
Disciplines
Domains
Work Product Kinds
Role Sets
Tools
Processes
Custom Categories
Guidance
*my_custom_category
Custom Category: my_custom_category
Content Elements
Manage the content elements that constitute this custom category.
Content elements:
Assign...
Unassign
Up
Down
Order
Sort Type:
Manual
Brief description of selected element:
Description
Assign
Preview

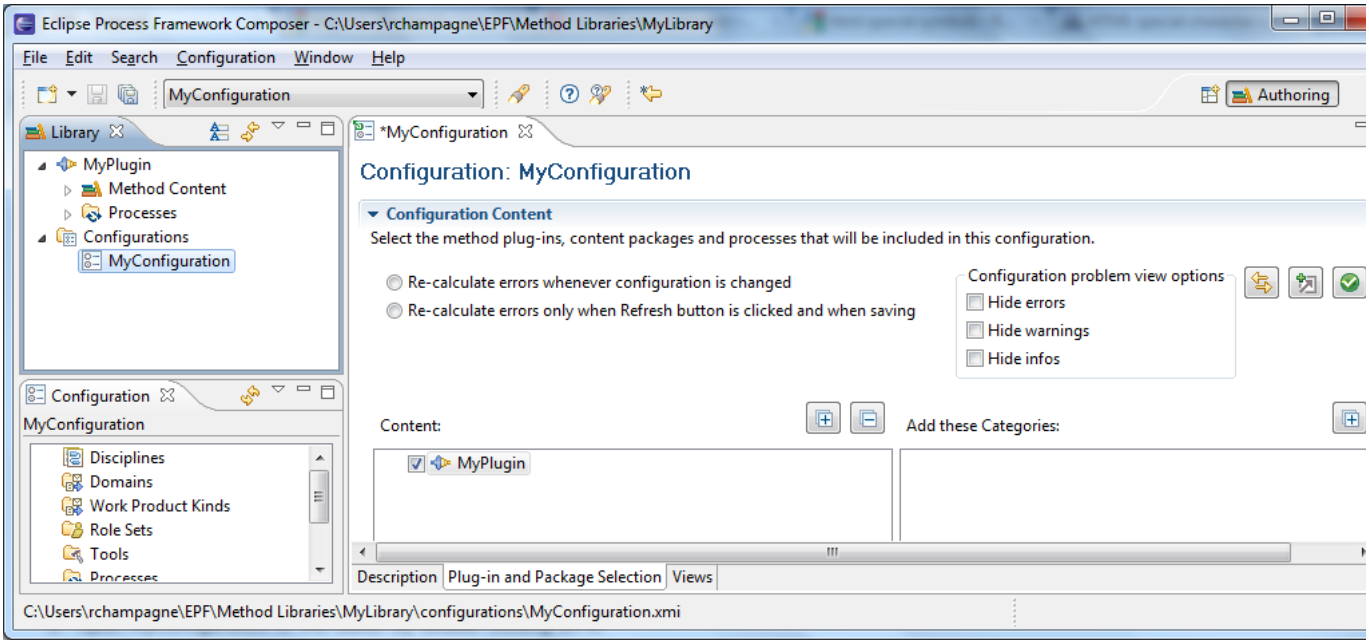

Select Dialog: Most common
Method Element Type: Method Plug-ins
Name patterns (matching names will be shown):
*
Patterns are separated by comma, where *=any string, ?=any character
MyPlugin
MyContentPackage
my_role
my_task
my_work_product
Custom Categories
Brief Description:
OK
Cancel



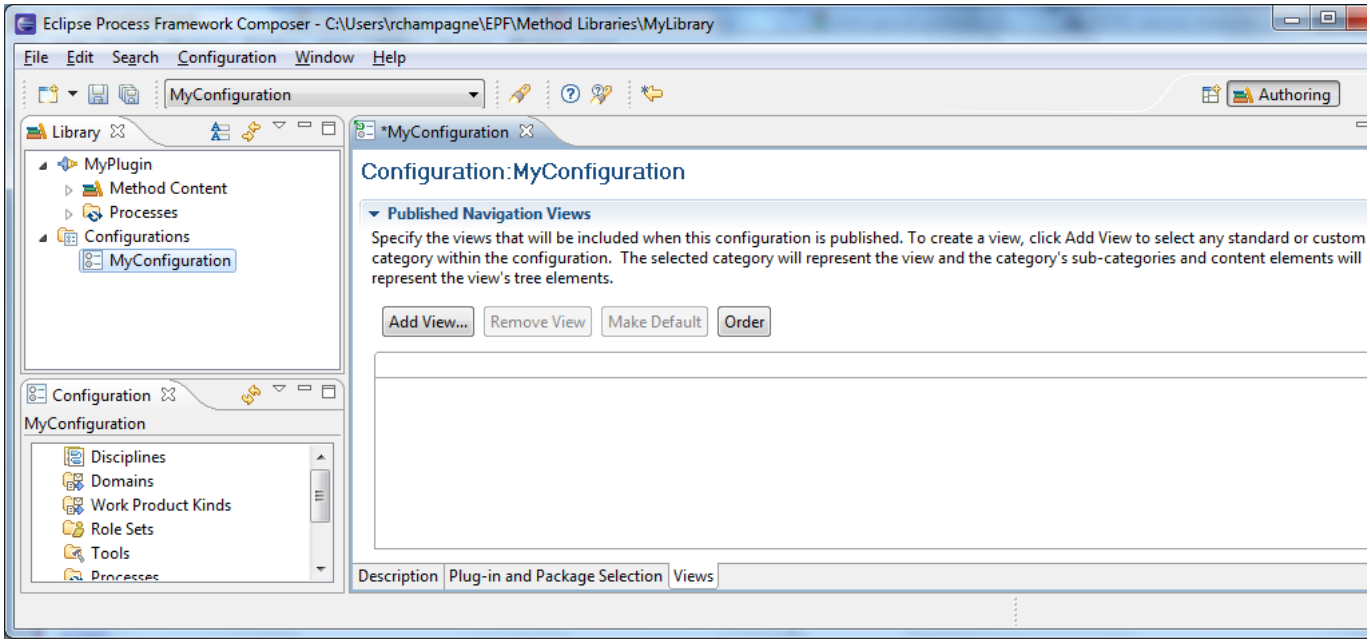
2. Open MyConfiguration in the editor by double-clicking on it.



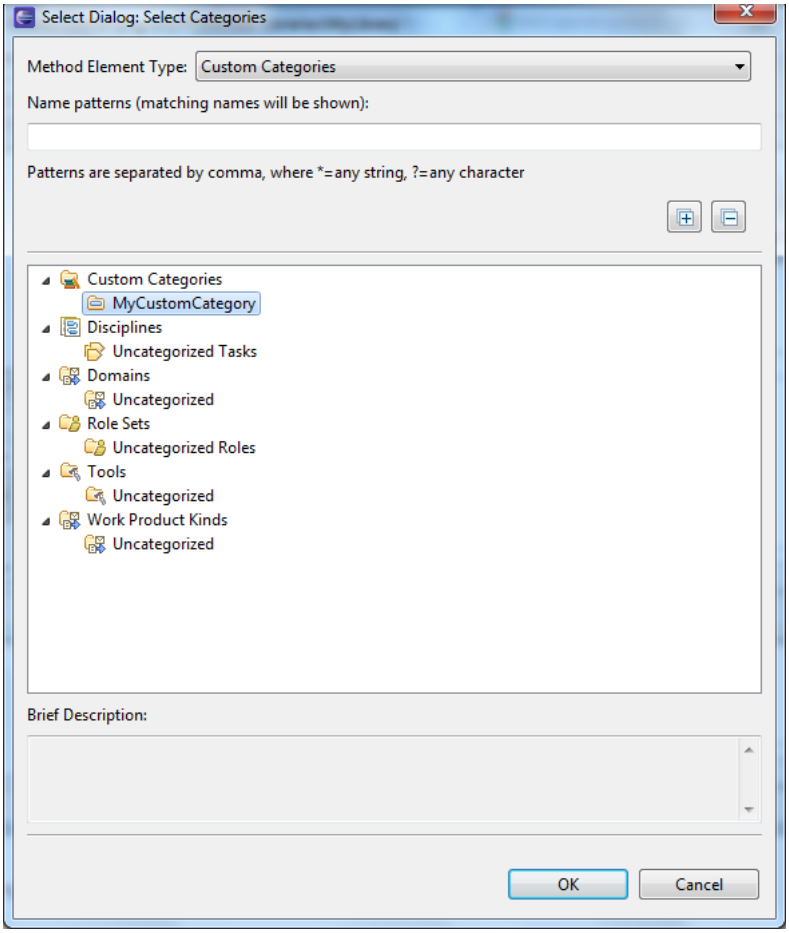
3. In the "Plugin and package selection" tab, select MyPlugin in the Content section.



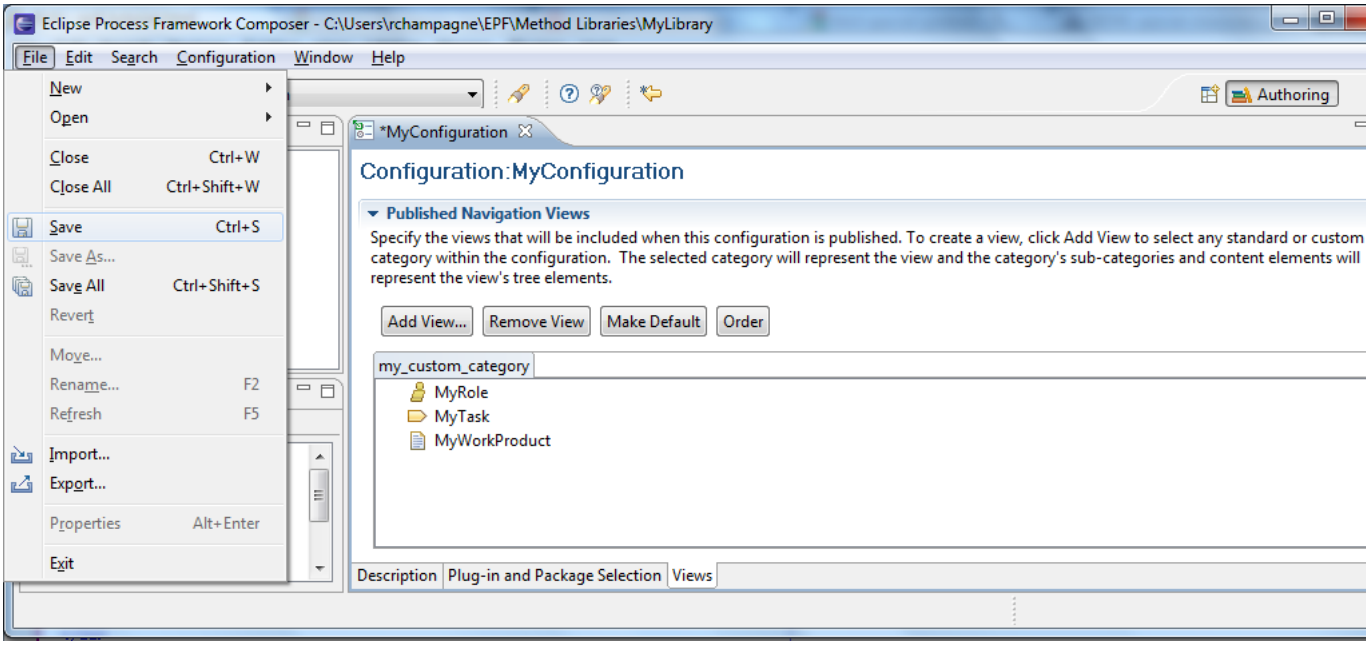
4. In the Views tab, click on the "Add view" button.



5. In the new window, select "Custom categories" as "Method element type". In the bottom part of the new window, select MyCustomCategory and press the OK button.

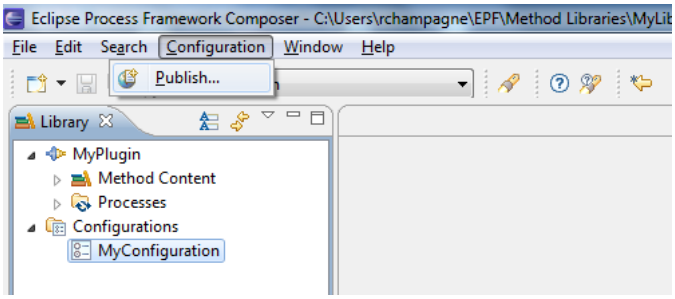


6. Save the configuration by selecting File→Save.

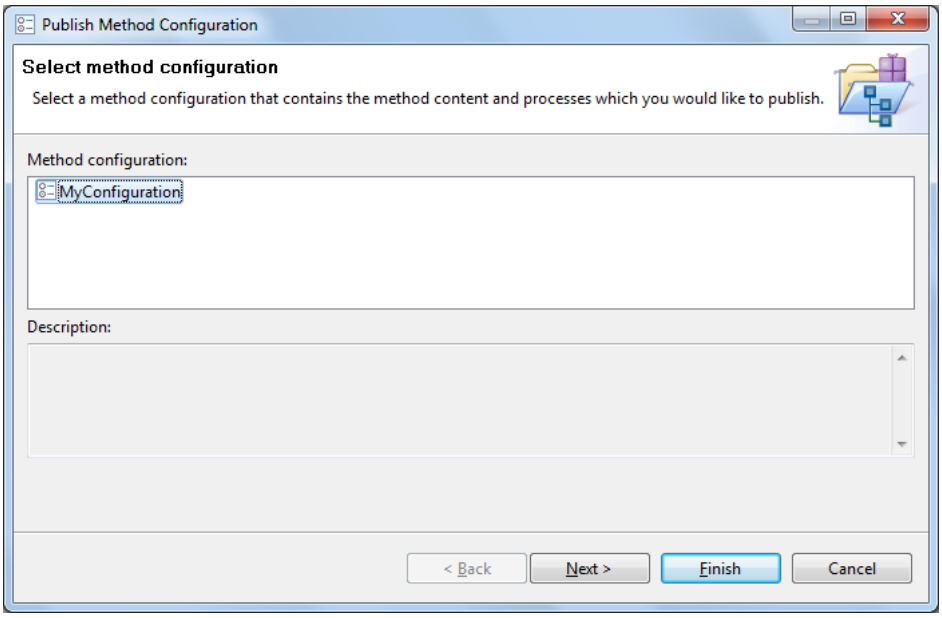


Publishing your plugin

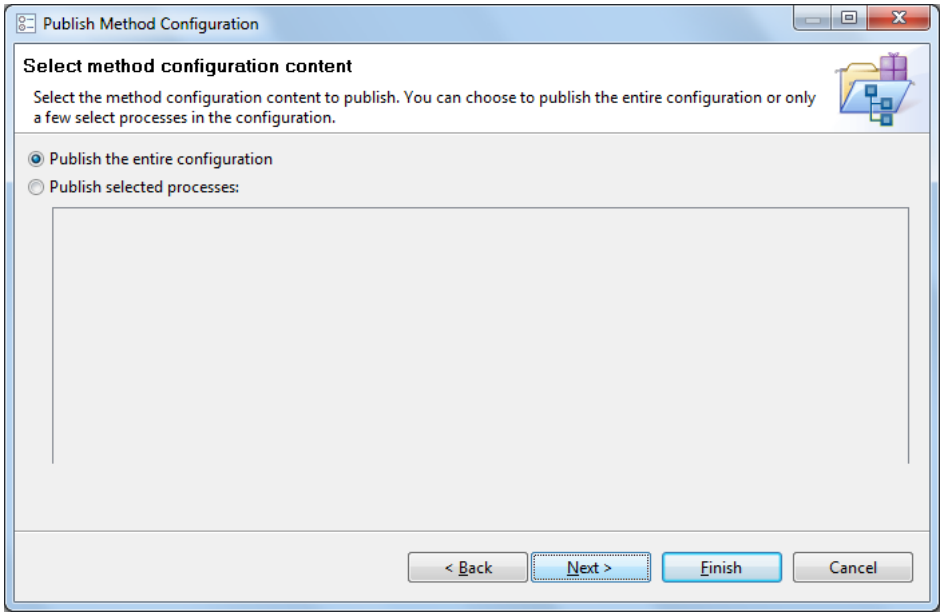
1. Initiate publication by selecting Configuration→Publish...



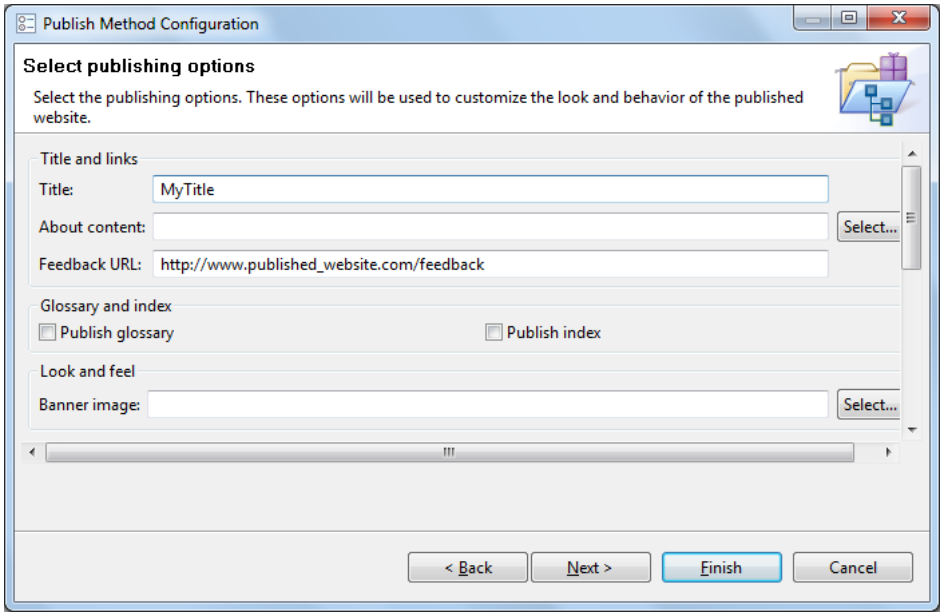
2. Select MyConfiguration and press the Next> button.



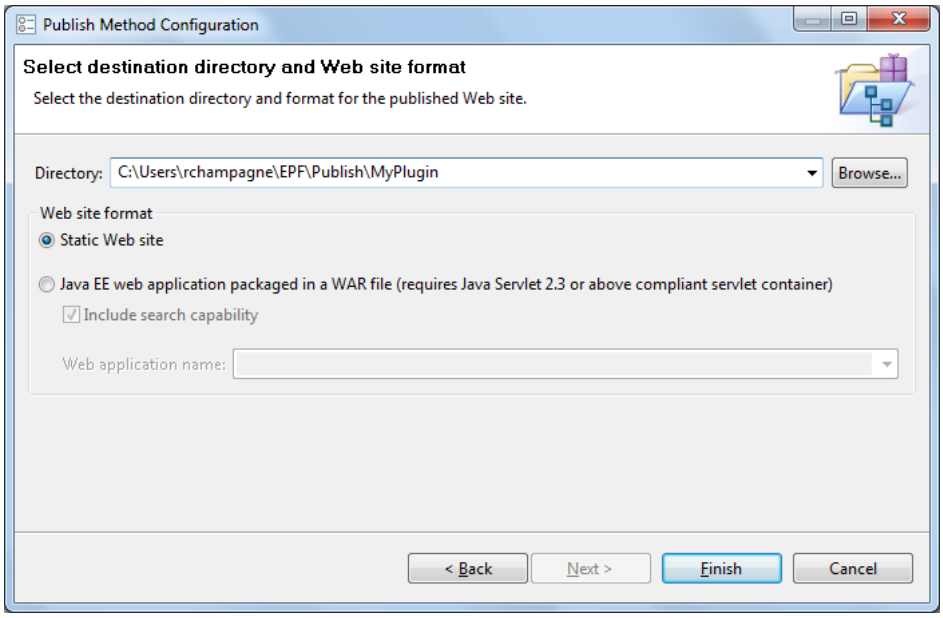
3. Select "Publish the entire configuration" and press the Next> button.



4. Specify MyTitle in the Title field and press the Next> button.



5. The default publication location is the Publish folder. Append "\Myplugin" to the proposed location and press the Next> button.



6. If all goes well, your plugin should appear in your default web browser. Experiment by navigating the three content elements that were created.