

Application and WPS design environment

- <https://github.com/jpbabau/Noumea>
- Download and unzip *EclipseWPS*
- *EclipseWPS* includes a Win32 Neon version of *Eclipse* (in the *EclipseNeon* folder) and a workspace *wpsWorkspace* containing the necessary folders
- To start *EclipseWPS*, double-click on *Eclipse*

Geoserver

Geoserver is recommended for the project. It is used to locally test the generated WPS
<http://docs.geoserver.org/latest/en/user/installation/index.html#installation>
http://docs.geoserver.org/latest/en/user/installation/win_installer.html

- *GeoServer* installation manual: <http://geoserver.org/download/>
- during install, you have to provide
 - o an Admin id and a Password, by default *admin* and *geoserver*
 - o a port for request invocation, by default 8080
- WPS extension installation :
<http://docs.geoserver.org/stable/en/user/services/wps/install.html>
- download the WPS extension : <http://geoserver.org/release/stable/>
- unzip the extension in the *GeoServerinstall/webapps/geoserver/WEB-INF/lib* repository:
 - o unzip the WPS extension
 - o add *gt-wps-17.2.jar* in the same repository, the version number (here 17.2) has to be the same as the version number of installed *geotools* jar in *GeoServer*
- Start the server, using the generated *Start GeoServer* in the *GeoServer Menu*
- Check that *WPS* is in the *Services* menu of *GeoServer*
- To test a deployed WPS, use your browser with the local address
<http://localhost:8080/geoserver>
- Connection to the server using *admin / geoserver*
- To stop the server, use the generated *Stop GeoServer* in the *GeoServer menu*

Create a WPS library

- In the *wfwpsLibrary* project create a wps library
 - o *File/New/Other /Example EMF Model Creation Wizards/Wfwps Model*
 - o *Next*
 - o Select the *wfwpsLibrary* folder
 - o Give a name to your lib *myLib.wfwps* (the extension *wfwps* is mandatory)
 - o *Next*
 - o *Model Object* : select *Workflow Wps*
 - o *Finish*
- Select the *wfwpsLibrary* folder
 - o Right click and *Viewpoints selection*
 - o If not selected, click on *wfwps view point*

Create a WPS Java project by reusing the TemplateProject

- Copy/paste the *TemplateProject* modifying the new *projectName*
 - o The project uses the jdk 1.8 (*Build Path/Configure Build Path*)
- Modify the following information in the *pom.xml* file


```
<groupId>yourGroupID</groupId>
<artifactId>NewProjectName</artifactId>
```

```
<version>VersionNumber</version>
<name>WPSjarName</name>
```

- Modify if necessary the version numbers in the *pom.xml* file



```
<geotools.version>17.2</geotools.version>
<geoserver.version>2.11.2</geoserver.version>
```

 - o the version numbers have to be the same as the geoserver et geotools version numbers installed in your *Geoserver* installation
- add the domain-specific java code in a specific package in the *src/main/java* folder
 - o you need to implement a non static *public* function *myFunction* member in a class *myClass.java*
 - o the class has a public by-default constructor, with no parameter
 - o the inputs and the output of the function must have the following types : *boolean, int, double, String, Geometry, FeatureCollection<SimpleFeatureType, SimpleFeature>*
 - o the execution of the function is self-consistent, no extra code has to be executed before it

Edit models of WPS

- launch the Noumea User Interface
 - o *My FX View* tab: click on the *Noumea Button*
 - o *Configuration* tab
 - select your *Library* file *myLib.wfwps*
 - select your *Java Project*
 - o *Modelling* tab
 - select your *class* file *myClass.java*
 - select your *function* *myFunction*
 - Click on *Modelling*
 - Gives a name for your WPS
- WPS java code generation
 - o *Generation* tab
 - select your function in the *Local WPS List*
 - click on *Generate WPS*
 - Java code has been generated in *src/main/java/*

Edit models of workflow

- in *myLib.wfwps*, select the *Workflow Wps*, right click *New Representation / new Library Diagram*
- the model of java WPS are yet represented
- use the *Properties* tab to have access to the properties of modeled elements
- add workflow (*new Workflow* in the *PaletteWPS*)
 - o add a *name* and an *abstract* to the workflow (*Properties* tab)
- add inputs and output to the workflow (*new Workflow Input* in the *PaletteWPS*)
 - o add a *name* and an *abstract* for each (*Properties* tab)
 - o select the correct type for each (*Properties* tab)
- double click on the workflow to open the corresponding graphical workflow editor
 - o the inputs and outputs of the workflow are represented and cannot be deleted from this view
- import local WPS (*import local WPS* in the *PaletteWPS*)
 - o select the corresponding WPS by clicking on  (*Properties* tab)
 - o double click on the WPS to add inputs and output
- add links to the workflow (*link* in the *PaletteWPS*)

- Use the palette to add WPS, remote WPS, WMS, WFS, constant data and links
- To delete an element (WPS, link, constant, í)
 - o *Right click on the element - Edit / Delete from Model*
- Validate the diagram before generation
 - o *Right click on the diagram - Validate Diagram*
- WPS java code generation
 - o *Configuration tab*
 - Re-select the *Library* file *myLib.wfwps*
 - o *Generation tab*
 - select the workflow in the *Workflow List*
 - click on Generate WF
 - Java code has been generated in src/main/java/

WPS deployment

- Select the Java Project
 - o *Right click -> Run As / Maven Build*
 - *Goals: package* (required for the first *Maven Build*)
 - o *Run*
 - generation of the *WPSjarName-VersionNumber.jar* in the *projectName/target* folder
- Nousemea User Interface
 - o checks that the *Geoserver* project repository is set
 - o *Configuration tab*
 - Select the *GeoServer Path*
 - The folder containing the *Geoserver bin* folder
 - o *Deployment tab*
 - Just click on *Deploy*
 - stop *GeoServer*, copy the generated jar in the corresponding *GeoServer* folder *GeoServerinstall/webapps/geoserver/WEB-INF/lib* re-start *GeoServer*

WPS test

- you can test the deployed WPS with *GeoServer* using the local address <http://localhost:8080/geoserver>