**Summary:**

**This tutorial will explain how to create a new Plugin for JDemetra+ by taking the example of creating a Plugin to compare two time series (absolute difference between them in their common time domain).**

**Requirements:**

**NetBeans IDE 7.4 or higher**

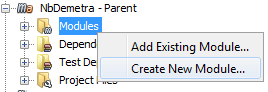
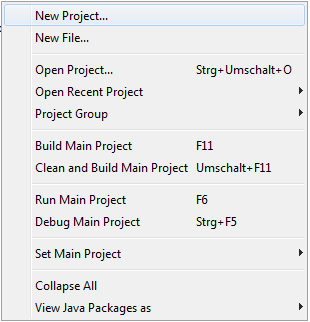
**Connection to maven repositories**

**Basic Java, Maven, Git knowledge**

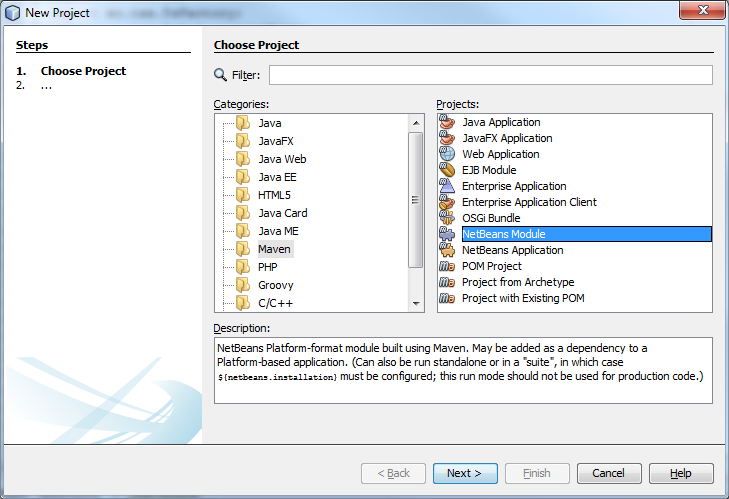
**Sources from the maven projects jdemetra-core and jdemetra-app (https://github.com/jdemetra)**

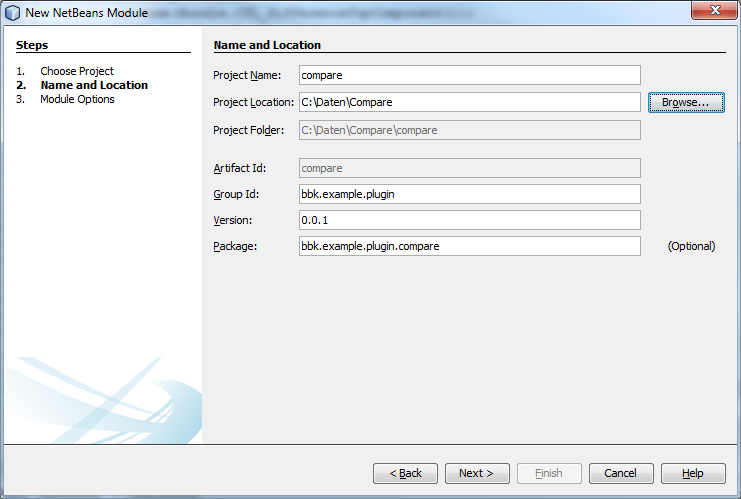
**New Plugin**

To create a new Plugin, you can either create a new Project or a new Module inside an existing Project.

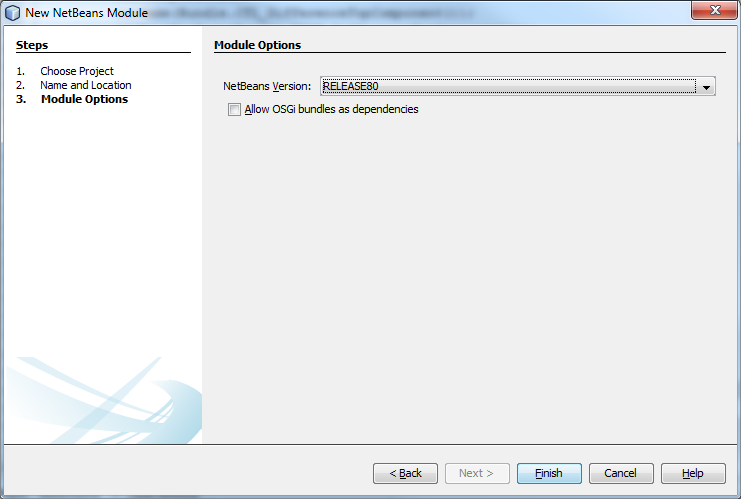


Then you need to choose a Maven / NetBeans Module.



On the next Window you choose a name and a location for your Plugin. For naming conventions of the Group Id and Version please read the mini-guide at <https://maven.apache.org/guides/mini/guide-naming-conventions.html>.

On the last page you can choose which NetBeans Version should be used, in general the default is fine.

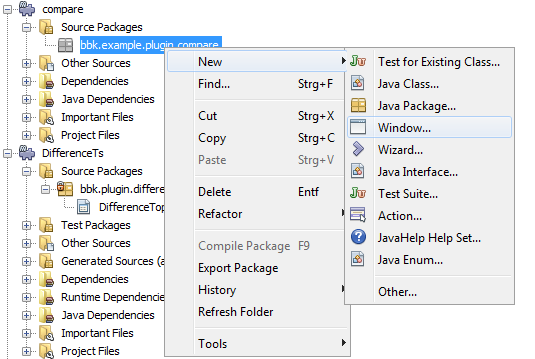


Now, you created your own Plugin. It is ready to install, but it has no functionality. This is explained in the next chapter.

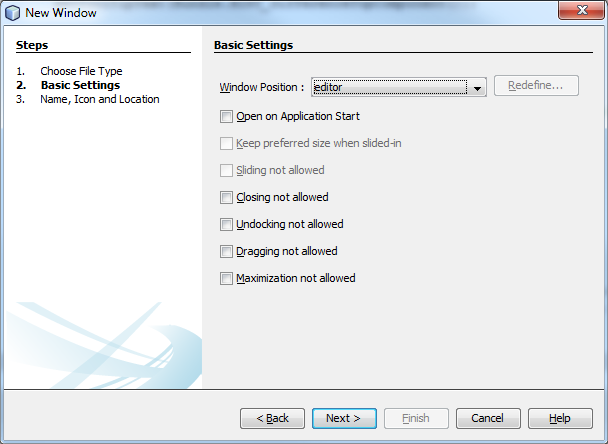
**New Window**

Your new Plugin can’t do anything at the moment, just the structure is created. So let’s add a Window to it. To add a new Window Right-click the package in which you want to store the Class. Then choose

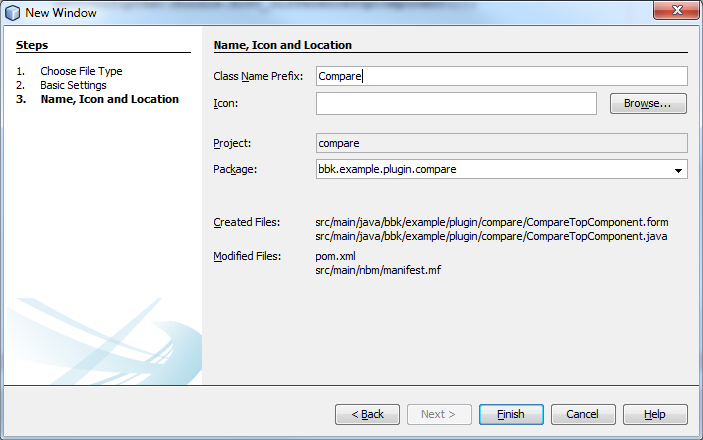
New -> Window.



On the first page you choose some basic settings, but at the moment we’re fine with them.



On the finish page you can choose a name prefix, an icon and the location. Furthermore you can see that NetBeans will create two files and modify the pom and the manifest for you.



After finishing the wizard NetBeans opens your new TopComponent in the Design-View and you can start building a GUI or change the Source in the Source-View.

**The User Interface (Example)**

For the user interface we just add the necessary components to the CompareTopComponent: a toolbar with a “run”-button, an input field and an output field.

* **The Button**

As toolbar we take NbComponents.newInnerToolbar() and add a JButton to it. Furthermore we add an ActionListener to the button, which takes the time series of the input field, uses the algorithm and put the result in the output field.

* **Input Field**

JDemetra+ provides you with several possible components to display and contain time series. One of them is JTsList. To get time series out of JTsList you just need getTsCollection and work with it like any other Collection.

* **Output Field**

For the output a chart would be nice, so we could use JTsChart. But if we want to look at the two original time series and their absolute difference JTsDualChart is better.

**The Logic (Example)**

In the Source-View you can see that NetBeans has already generated all the necessary code for your TopComponent.

JDemetra+ already has methods to do calculations with time series. You can find them in the TsData Class. For further information please visit <https://github.com/jdemetra/jdemetra-core/wiki/Time-Series>.

So the easiest algorithm to compare would be:

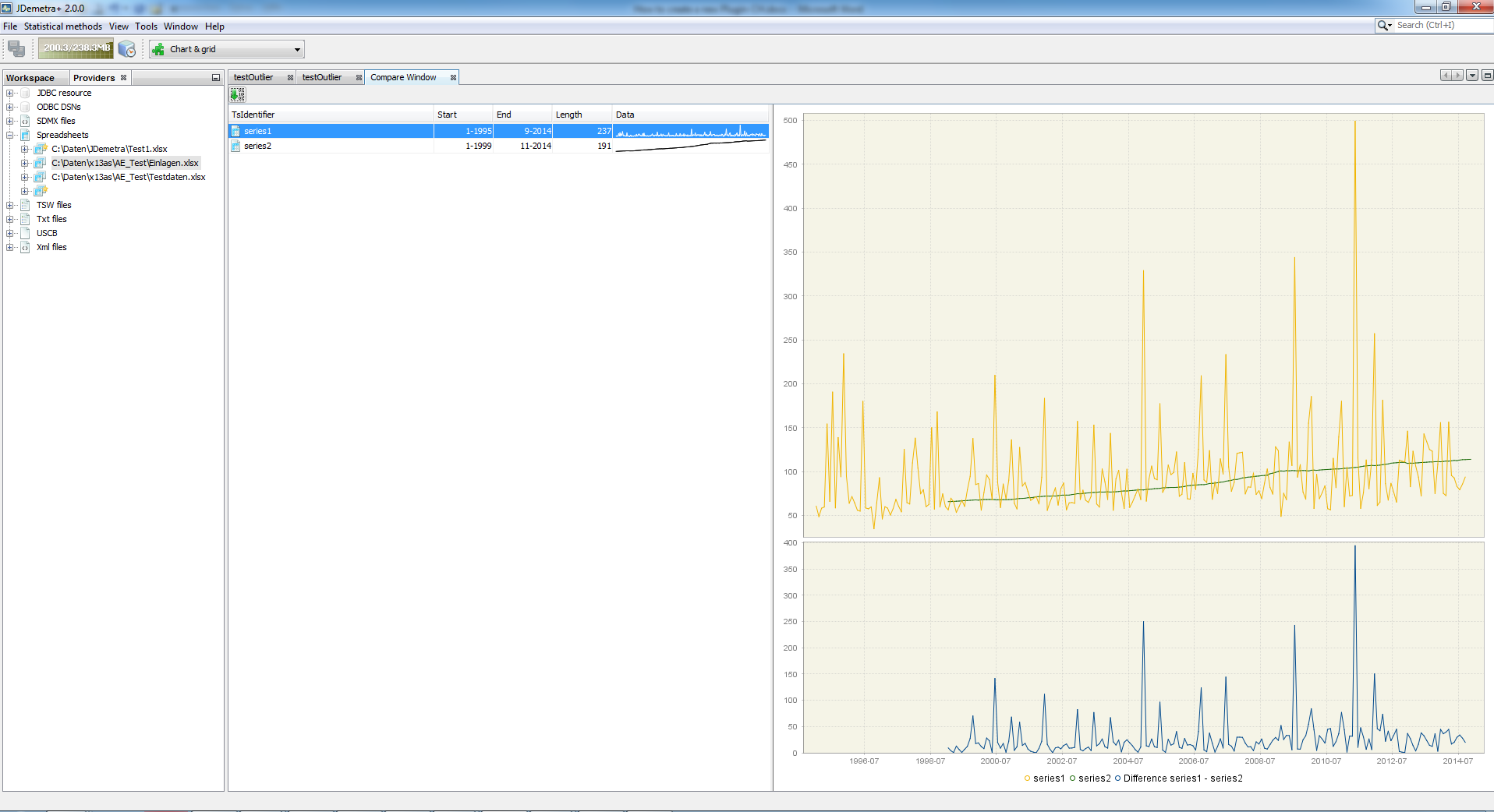
|Time series 1 – Time series 2| or with JDemetra+ methods

tsData1.minus(tsData2).abs()

This algorithm only works with the values on the common time domain and won’t return a result if there is no common domain.

**Additional useful information**

* If you add your Plugin as dependency to NbDemetra (nbdemetra-app) it will always take your last build and install the plugin automatically
* Check your pom to see if any version conflicts exists, the dependency version management should be done by the parent pom if possible



You will find the TopComponent at Window/Compare. For an example code see CompareTopComponent.java