

# NetbeansModulesUsingMevenide

## Using Maven and Mevenide to create Netbeans modules



### Please note

These instructions apply to Maven1. If you are using Maven2 (recommended), please check [this excellent article](#)

This is a step by step guide on developing Netbeans modules (plugins) using Maven and Mevenide. It doesn't matter if the module is for Netbeans IDE or your custom application based on Netbeans Platform.

The tutorial assumes basic knowledge module development as described on netbeans.org site ([Api List page](#) and [openide site](#) in general). In this text below we discuss the specific issues of building Netbeans modules using Maven, not how to use the APIs to create a module.

## Install Maven

Installing Maven is rather straightforward, just follow [instructions](#).

It's important that you setup the MAVEN\_HOME environment property (Windows installer does it for you).

If you are behind firewall, you should [setup Maven](#) to use your proxy settings. It's best done by creating a file named **build.properties** in your user home directory and populate it with the required proxy related properties.

## Install Mevenide

First download the latest [Mevenide release](#).

This tutorial assumes you have installed Mevenide for Netbeans, however the same can be achieved through the Mevenide for Eclipse with minor variations, in case you want to develop Netbeans modules with that IDE.

## Prepare environment

On the command-line, run this command which downloads the appropriate maven plugin for creating Netbeans modules:

```
maven plugin:download -DgroupId=mevenide -Dversion=1.3 -DartifactId=maven-nbm-plugin
```

Assuming you already have the Netbeans IDE or Netbeans Platform that you want to develop against, please recall it's installation directory. If you don't have it, please install it first.

Run goal

```
maven nbm:fill-repository
```

. It will prompt you for the installation directory. Then it will find all Netbeans modules that are in that directory structure and copy them to your local Maven repository. (That's where Maven is looking for artifacts/dependencies, unfortunately there's currently no remote repository that would host Netbeans artifacts).

You can later check [maven-nbm-plugin](#) homepage for updates and details on other available goal and customization properties.

## Create project

Start up Netbeans IDE (assuming Mevenide 0.6 installed) and create a new project, under Maven category there is "Sample Netbeans module" project template. A skeleton project is created for you.

Now we need to specify the correct Netbeans API module jar dependencies.



### Specify dependencies visually

You can do it **visually** in the project's Properties dialog. Invoke the popup menu and in the shown dialog go to the "Dependencies" tab. There you can see that openide (the main API module) is already added as dependency. If there's a small red line next to it's icon, it's not present locally in your repository. (You have different IDE/platform version). Edit it and select the artifact that you have locally. You can also add additional dependencies if you require some. For other dependencies however you need to declare the

netbeans.module.dep property. It denotes that you want to have a dependency on another netbeans module. The nbm plugin will use this information when generating your module's manifest. The most frequent value for the property is [spec](#). For more details, check [maven-nbm-plugin docs](#)

### Manual editing

Or to edit manually, go to Files view and open the project.xml file in the project's root directory. What's important now is the <dependencies> element.

The **openide** dependency is required, check if the generated version matches the version that you have in your local repository. (USER\_HOME/.maven/repository/netbeans/jars is the directory to check for the correct version number)

```
<dependency>
  <groupId>netbeans</groupId>
  <artifactId>openide</artifactId>
  <version>4.41</version>
  <type>jar</type>
</dependency>
```

The project template also generated a manifest file (manifest.mf) under conf/ directory. Edit it to include the correct version as well.

If your module depends on API that are not part of openide, you have to declare it as dependency here as well. Again check for correct version and artifactId in your local repository.

```
<dependency>
  <groupId>netbeans</groupId>
  <artifactId>org-openide-io</artifactId>
  <version>1.2</version>
  <type>jar</type>
  <properties>
    <netbeans.module.dep>spec</netbeans.module.dep>
  </properties>
</dependency>
```

What is important here is the **netbeans.module.dep** property. It tells the maven-nbm-plugin to generate a module dependency when generating the jar and nbm file. (You should not declare those dependencies in the manifest file but let the plugin generate it dynamically from the project definition)

**spec** is the most common value, meaning you depend on the specification version. For more details, check [maven-nbm-plugin docs](#)

If you have done everything correctly, you should have the Netbeans APIs now available in the editor when you start coding your module.

## Build project

Building is simple, just run "Build" or "Rebuild" from the project's popup menu. The resulting module jar and nbm files will appear under target/nbm directory in your project.

That's it.

wiki