

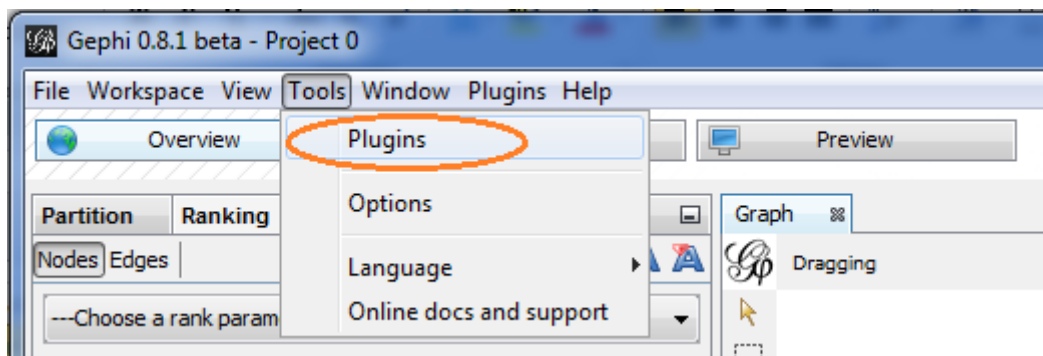
## User manual

This manual is dedicated to the Gephi “Java Class Dependency” plugin. It allows users to create graphs, showing java class dependencies. It’s compatible with Gephi 0.8.1 and higher.

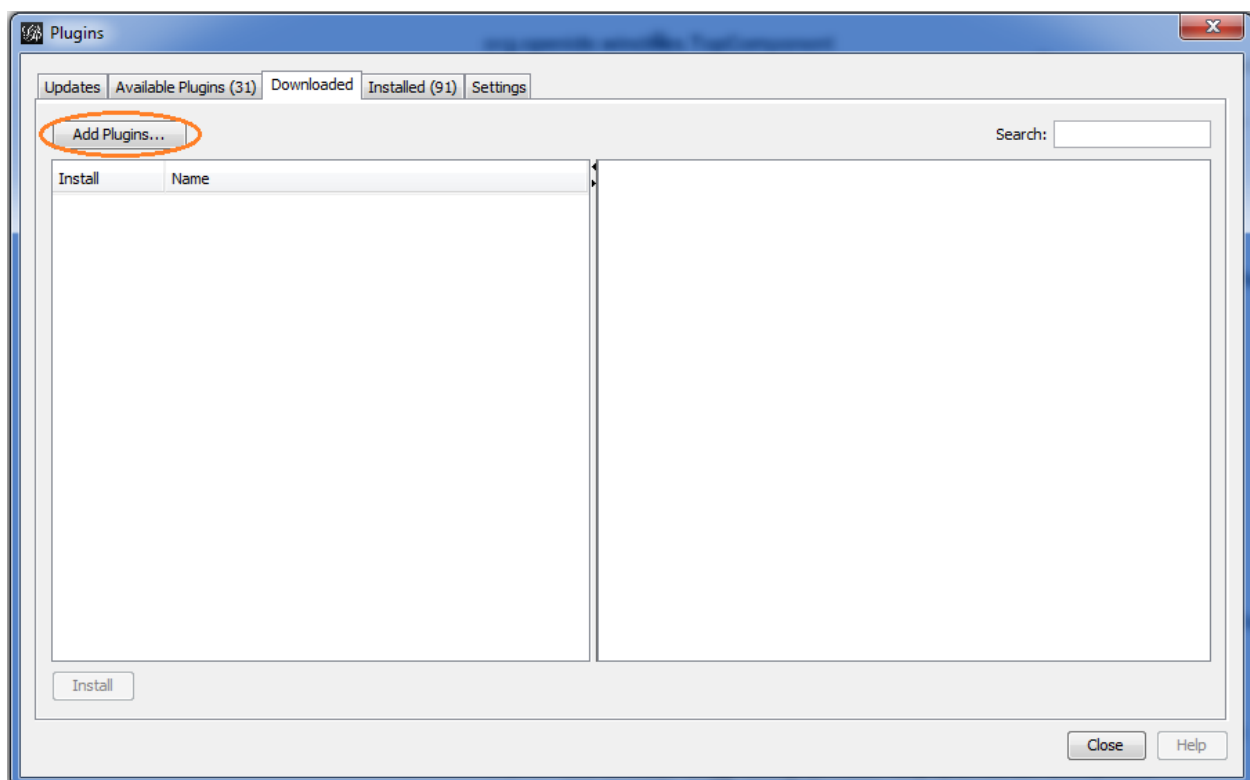
### Plugin installation

First of all run the Gephi application and check it’s version, using “Help->About” menu. If it is older than 0.8.1 please download the latest version.

To begin the plugin installation you need to open menu “Tools->Plugins” in Gephi application.

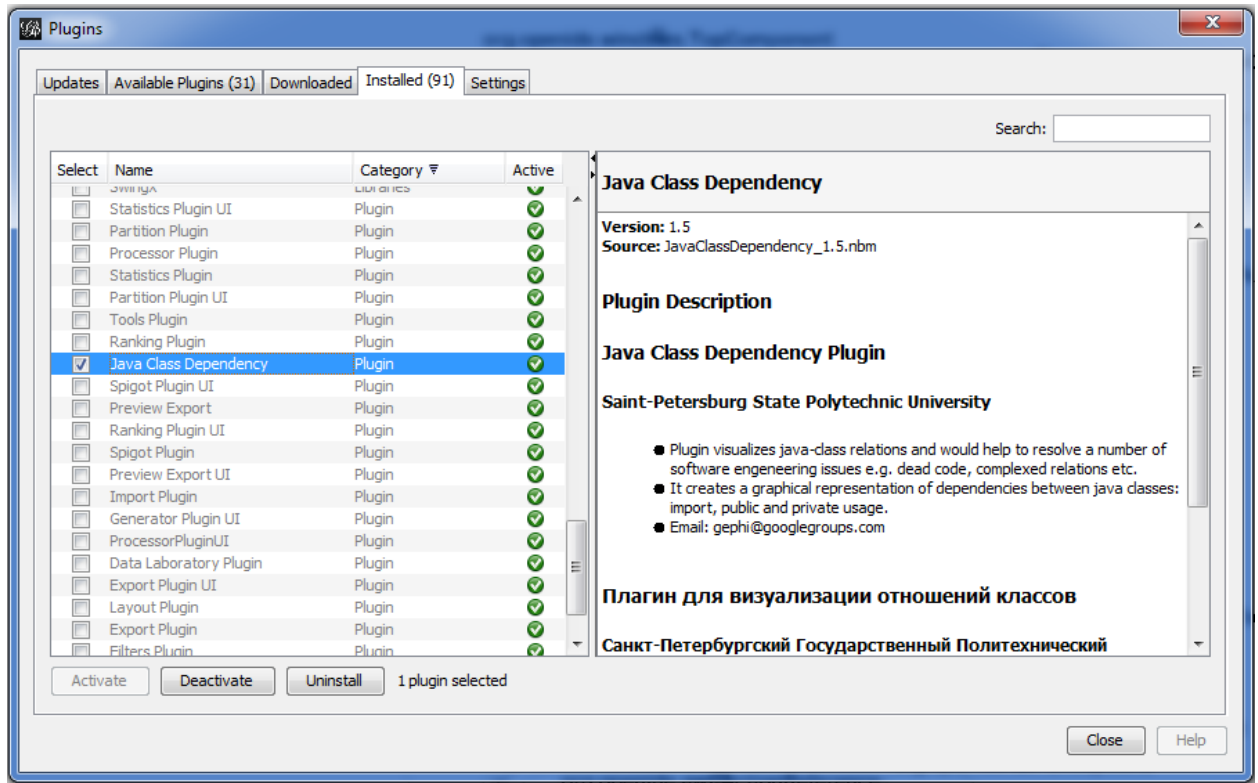


Then you should go to the “Downloaded” tab and press “Add plugins” button. There will be shown a window, where you should specify a path to the plugin .ndm file.



After this step, a special installation wizard will start up. It will help you to choose installation properties and control the installation process. After the end of installation it will restart the Gephi application.

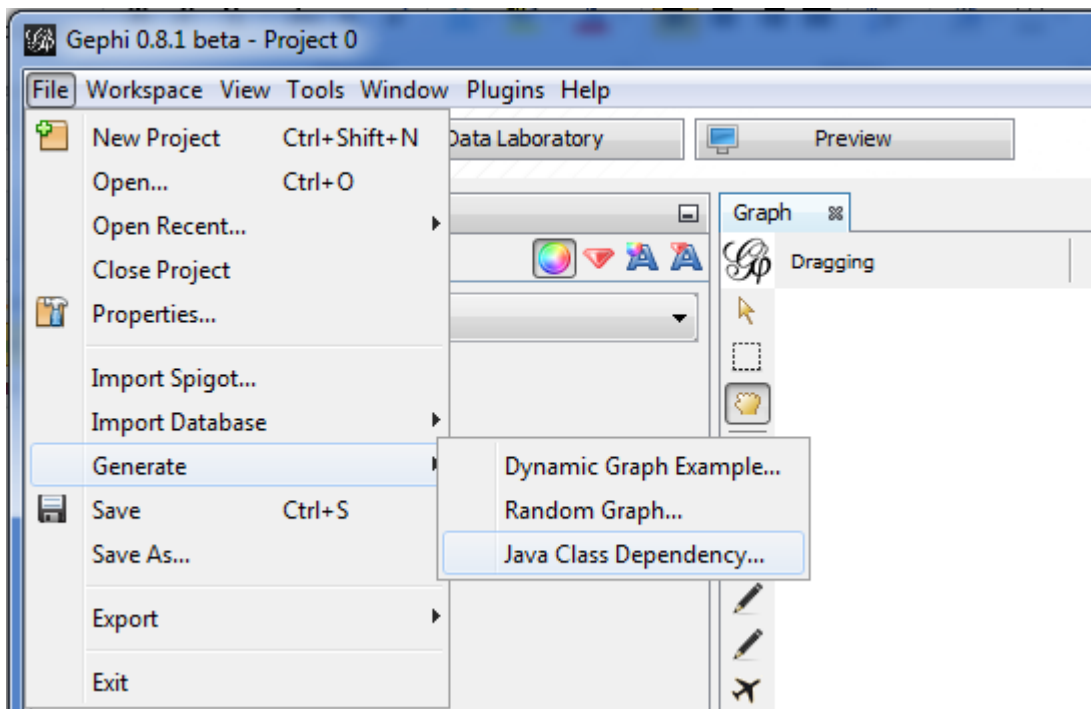
Now you need to activate the plugin using “Installed” tab (Tools/Plugins/Installed).



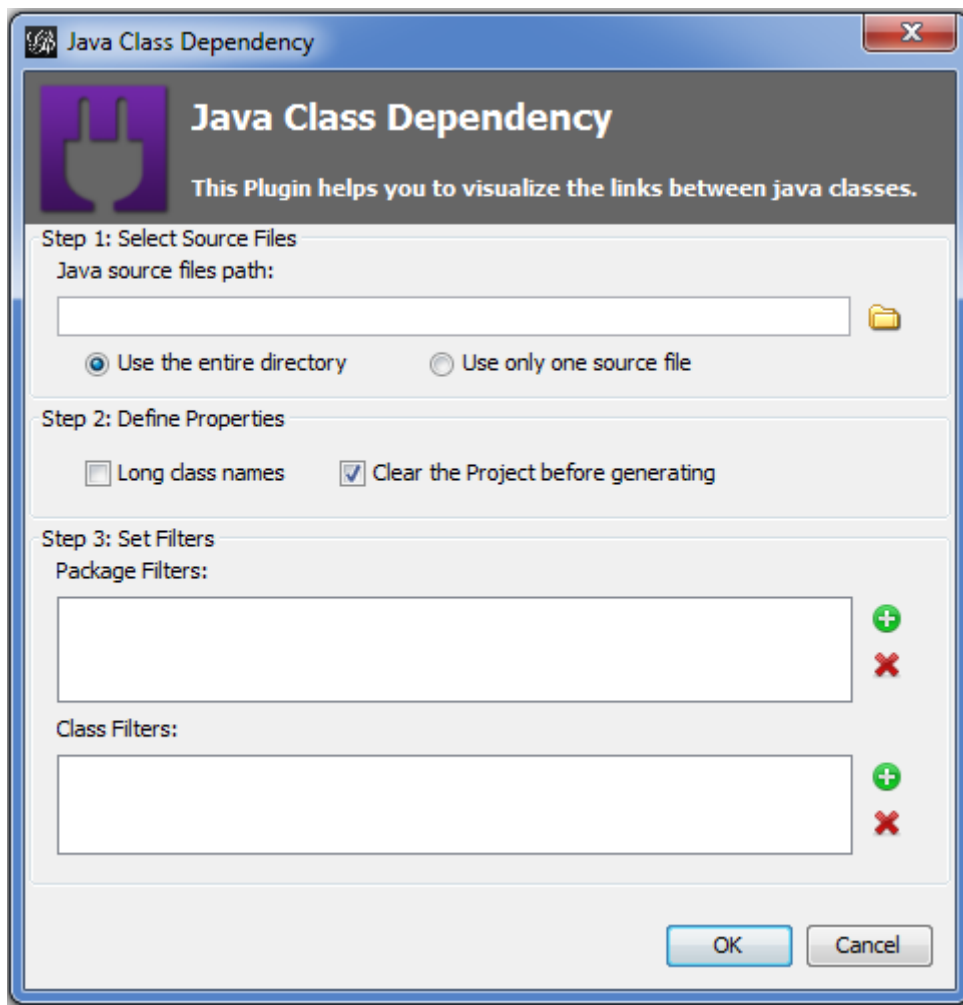
Plugin should be working now!

## Working with plugin

To start working with plugin you should launch it using “*File->Generate->Java Class Dependency*” menu.



Plugin configuration window will be shown:



Here on step 1 you can specify a path to a single java source file or to a folder containing multiple source files.

On the second step you can set “Long class names” and “Clear the project before generating” options. The first one allows you to display the names of the classes on the graph taking into account all packages they belong to. If not checked only the names of the classes themselves will be shown. The next option clears Gephi workspace before generating new java class dependency graph.

On the third step you can set package filters and class filters. They allow you to exclude such classes and packages from graph that are not interesting to you.

After specifying all this options you can press “OK” button to begin graph generation. It will appear on the Gephi workspace.

Have a nice work!