



**Sofa 2.0**

**Repository Editor Eclipse plug-in.**

**Installation guide**

***Maksym Nesen***

Chapter 1. Install using the update manager.....	3
Chapter 2. Install from the source code.....	8
Chapter 3. Build plug-in from the ant script.....	10

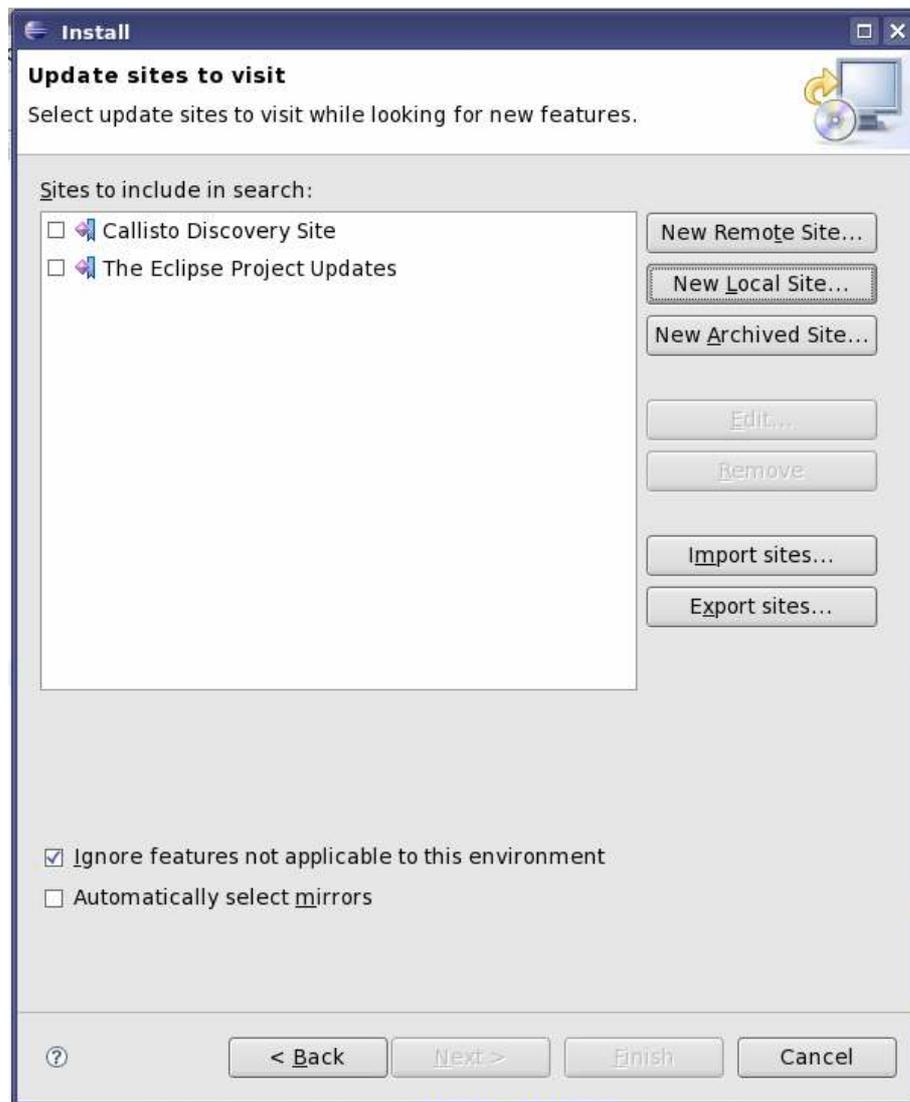
# Chapter 1. Install using the update manager

To install the plugin you should:

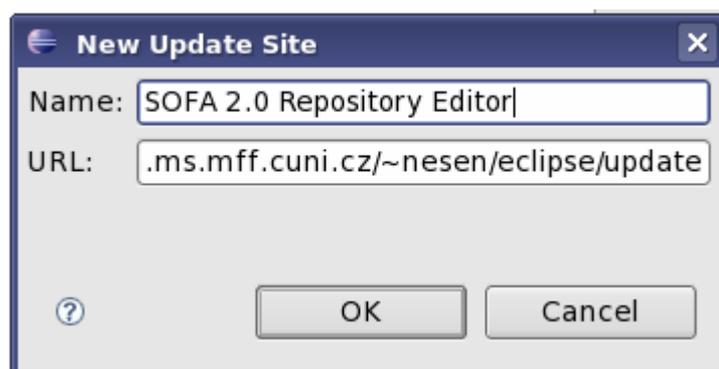
1. Download SOFA Repository with cushion extension. <http://sofa.objectweb.org/>
2. Build and deploy Repository Server
3. Run Eclipse and install plug-in from the remote site:
  - \* open Help->Software Updates -> find and install;
  - \* Choose Search for new features to install;



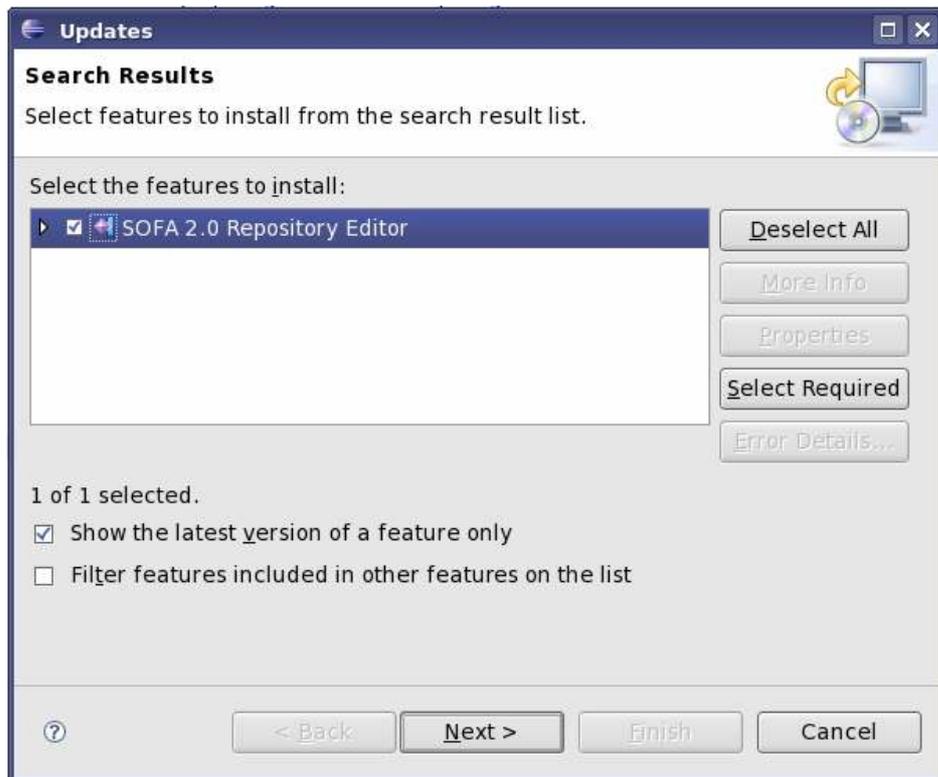
- \* Click 'New Remote Site';



- Set URL to: <http://nenya.ms.mff.cuni.cz/~nesen/eclipse/update>;



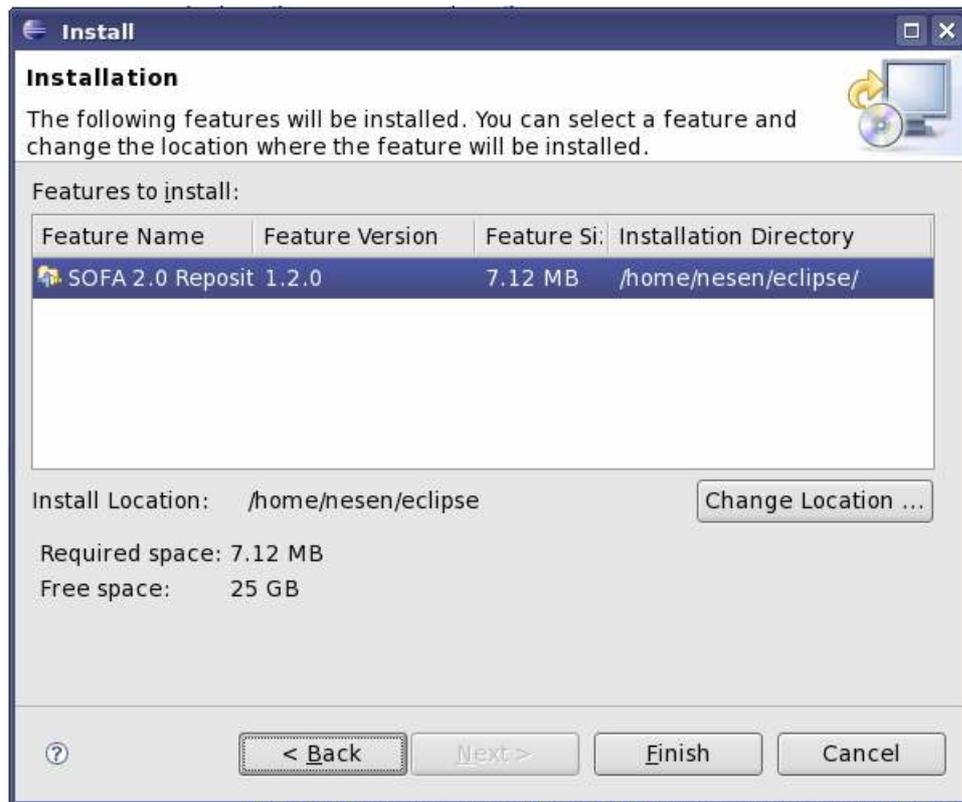
- \* Set Name to: SOFA 2.0 Repository Editor;
- \* Click ok and then Finish;



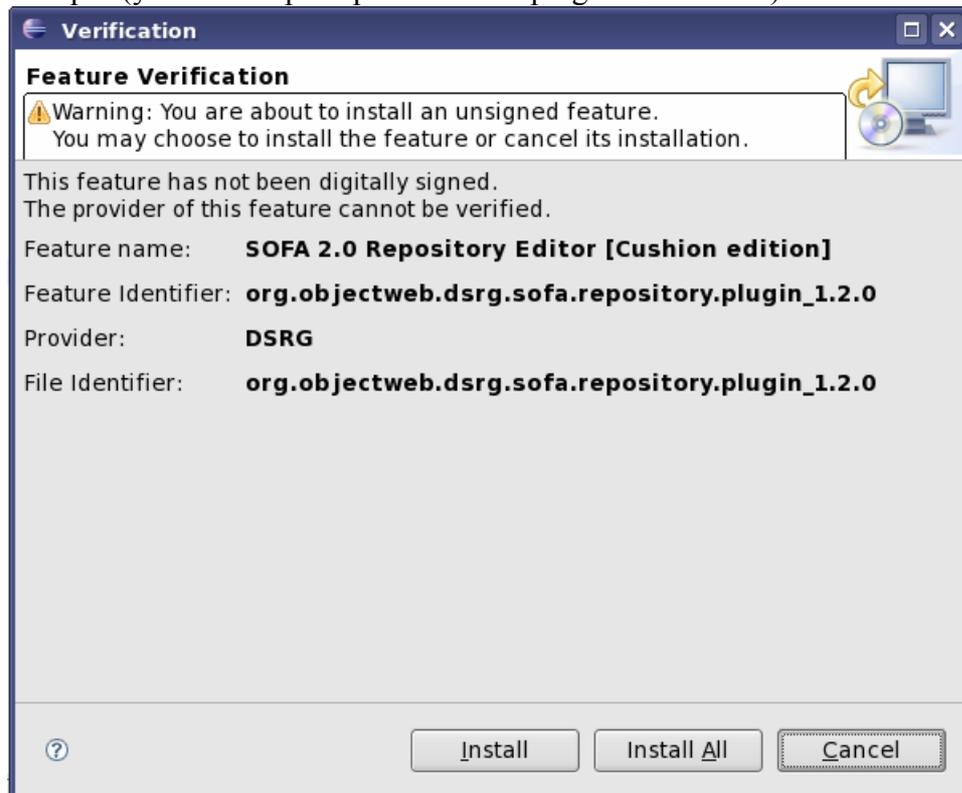
\* Accept all terms and agreements and install all listed features from the plug-in



\* You will be prompted for permission to install not signed Eclipse plug-in. Just click 'Install All'



\* Restart Eclipse (you will be prompted after the plug-in is installed)



\* Open SOFA 2.0 perspective



## Chapter 2. Install from the source code

Pre-requisites:

- Eclipse 3.2 or higher. Official Eclipse site - <http://eclipse.org/>
- Java 1.6. Official Java site - <http://sun.java.com/>
- Zipped code bundle. Direct download link - [http://dsrg.mff.cuni.cz/~nesen/eclipse/updates/source\\_code\\_bundle.zip](http://dsrg.mff.cuni.cz/~nesen/eclipse/updates/source_code_bundle.zip)

To install the plug-in from the source code distribution you need to download the zipped file which contains source codes for the plug-in itself, the plug-in's feature, and the plug-in's deployment site. The bundle is located here –

[http://dsrg.mff.cuni.cz/~nesen/eclipse/updates/source\\_code\\_bundle.zip](http://dsrg.mff.cuni.cz/~nesen/eclipse/updates/source_code_bundle.zip)

Unzip the file to a folder and open Eclipse IDE. When eclipse is open, you should mouse right-click in the files explorer, choose Import... and use Import Wizard to import existing project into workspace. You should import all 3 projects into the Eclipse workspace. Please make sure you have 3 new folders in the file explorer - **plugin.core**, **plugin.site**, **plugin.features**. After you have all these projects imported, you can compile the plugin.core to the deployment state. This means you should open the project plugin.core, then open the file plugin.xml in the root of the project and at the first tab of the plugin.xml editor you should click on the 'Export' link. This will open the Export wizard. There you should choose the project to be exported to the plugin.site folder, and the exported plugin should be packed as jar archive. When the project is successfully exported, you can close the plugin.core project and open the plugin.feature project. In the root of this project you should open the feature.xml file and in the feature.xml editor click on the 'export' link. This will open the Export wizard, in which you have to set export folder to the plugin.site project location and choose newly exported feature to be packed as a jar archive. Then click export.

After all of this is done, you are nearly to have pre-deployed plug-in. The last step you should do is to close plugin.feature project and open plugin.site project. In the project you should open the site.xml file and in the site.xml editor firstly click on the 'Synchronize' button and choose synchronize all, then perform synchronization. And after the synchronization is done, you should click on the Build button and choose build all option. then perform build. Now the plug-in is ready to installation. Please close all projects and open the Help->software updates->find and install wizard. In this wizard click on the 'new local site' button and set the folder to the plugin.site project location. Set the name: 'SOFA 2.0 repository editor'. Then click the Finish button and perform the plug-in installation as described in the 'binary installation' section of the documentation.

**NOTES:** 1. Plug-in version. The plug-in version can be changed in the plugin.xml editor on the Overview tab. After the version is changed, the according changes should be performed in the plugin.feature project and the plugin.site project. In the plugin.features project the version should be forced to be copied from the plugin.core project version. This can be done at the Plug-ins tab of the feature.xml editor. Please make sure both versions (in the plugin.core project and in the plugin.features project) are equal. Then save the plugin.features project and go to the plugin.site project. Here open the site.xml editor and remove actual feature from the list of properties. This should be done because the version is changed and site.xml can not see newly created feature and plug-in. After the old feature is removed, add the new one to the same place. The only

difference now is version number. Then synchronize and build all features from the site.xml editor.

## Chapter 3. Build plug-in from the ant script

- Ant. Official Ant site - <http://ant.apache.org/>
- Java.1.6. Official Java site - <http://java.sun.com/>

To build the plug-in from the ant build.xml file you should download the file with zipped sources of the plugin.core, plugin.features, plugin.site xml and unzip the file to a folder. Than change dir to the folder and perform 'ant build.update.jar'. This should build a jar file with needed plug-in inside.

But the plugin.features and plugin.site projects you have to deploy using the export wizard from Eclipse. Those project do not contain any java source files and there is no build.xml predefined for those projects. The site.xml file can be edited manually because it has very simple structure. It is needed to edit just name of a deployed feature and attribute whether it is patch for another feature or not.

The feature.xml file should be packed to a jar file with the same name as the plug-in's jar file, but should be stored in the features folder of the plugin.site project. The syntax of the feature.xml file is a bit more complex that the one of the site.xml, but it still can be edited manually, and for now it is not required to edit it much. Just in case the plug-in's version is changed.