

Developing a Single Plugin

Overview

You can develop a plugin for CShell without having to check out the entire CShell source by using Eclipse, a p2 repository, and a target platform definition.

Video Overview

For quick reference, please view this video overview of creating a basic CShell/Sci2 plugin development environment:

[CShell Environment Setup](#)

Prerequisites

Make sure you have read and followed CShell guide to [Setting Up the Development Environment](#).

Developing a Plugin

If this is your first time developing a CShell plugin, you can get started by following the [Hello World Tutorial](#).

Warning: Plugins will only be updated if their version numbers change. When committing changes, it is **very** important to also update the version number in the *manifest.mf* and the *pom.xml* files.

Debug Tips

Currently, Eclipse debugging is not working for CShell development. We are working on it. To be able to debug, you will need to make friend to OSGi console. See [Accessing the OSGi Console](#)

Adding the Plugin to the Build System

Adding your plugin to CShell's build requires that you both add a *pom.xml* file for Maven and add the plugin to a feature being built into CShell. The directions [given here](#) should provide enough information to include the plugin into the build. Note that you don't need to checkout the entire workspace, just the parent above where you'd like to put your plugin. Once you have that, add your plugin to the pom and desired feature and it can be added without checking out the entire tool.

Possible Problems/Troubleshooting

Very Large *.bundle_core* Folder

- If you notice that the *.bundle_core* used by the p2 repositories is not deleting old, unused p2 directories (possible as of Indigo due to an Eclipse bug), you can force a resolution by creating an empty, dummy Target Platform then deleting it. Eclipse will then refresh the p2 repository plugins.

Forgetting Version Number Updates

- Plugins will only be updated if their version numbers change. When committing changes, it is **very** important to also update the version number in the *manifest.mf* and the *pom.xml* files.

Out of Date Plugins

- If your package imports a plugin that you know to be newer than is shown in your workspace, you might need to reload the target platform. Do this by opening Window -> Preferences then Plug-in Development -> Target Platform. Highlight the target platform on the right side and click Reload. You should now have the most up to date plugins.