Eclipse Plugin FAQ

Requirements and Installation

Q. Where can I install Groovy-Eclipse?

A. The source code for the new version is all in codehaus.

ℹ️ **Groovy-Eclipse update site**
To install 2.0.0 in Eclipse 3.5.x, add the following URL to your Eclipse update manager:
`http://dist.springsource.org/release/GRECLIPSE/e3.5/`

To install in Eclipse 3.4.2, use the following URL:
`http://ci.repository.codehaus.org/greclipse/snapshot/e34/`

Note that 3.4 support is available, but we are no longer actively working on it.

Q. What version of Eclipse do I need?

A. You will need Eclipse 3.6.x for the 3.6 version of the plugin, Eclipse 3.5.x the 3.5 version of the plugin and Eclipse 3.4.2 for the 3.4 version of the plugin. Since Groovy-Eclipse relies on Eclipse's Java tooling (the JDT), almost any flavor of Eclipse that includes the JDT will be compatible. This includes the Java IDE, Classic, RCP, JavaEE, etc. Flavors such as Eclipse for PHP developers are not compatible because it does not include the JDT. If you are interested in Grails tooling, then you must install the SpringSource Tool Suite.

Q. How do I install from an update site?

A. Navigate to "Help > Software Updates". On the "Available Software" tab, click "Add Site" and enter the update site URL appropriate for your version of Eclipse: http://dist.codehaus.org/groovy/distributions/greclipse/snapshot/e3.6 or http://dist.codehaus.org/groovy/distributions/greclipse/snapshot/e3.5 then click OK. Now the update site will be on the list, open it and mark the entry 'Groovy-Eclipse plugin'. Finally, in the top right,

click "Install" and follow the dialogs to complete installation.

Q. How do I get started?

A. Once installed you can create new Groovy projects in a similar way to how you can create Java projects. Or if you have an existing Java project which you want to add groovy code too, you can select the project in the package explorer, right click and navigate to "Configure" > "Convert to Groovy Project". This action adds the Groovy nature to your project. Once this happens, the icon decorator for your project will change to reflect the new Groovy nature. And any .java files, the builder for the project will also build .groovy files. Finally, if you have some older groovy projects created with the previous version of the plugin, they must be migrated to exploit the new plugin. See the next question:

Q. Existing groovy projects

A. Existing projects that are based on the old groovy plugin need to be migrated.   There is a migration tool included with the plugin. To use the tool, select the projects that you want to migrate, right click, select the Groovy sub-menu, and then choose "Convert legacy groovy projects". Note that this option will only appear if you are selecting projects that have the old groovy nature.

Q. So you modify the JDT Compiler - will that affect my pure Java projects?

A. The changes to the JDT compiler are minimal, and all the extra extensions added to it are only executed if a project has the groovy nature. Any adverse affects on building pure Java projects should be considered a bug.

Q. What version of groovy does Groovy-Eclipse use?

A. Groovy-Eclipse ships with Groovy 1.7.x. It is possible to install a Groovy 1.8 compiler, but you must install from the development update site. See Compiler Switching within Groovy-Eclipse for more information on this.

Q. How do I change the compiler version in Groovy-Eclipse?

You can change the version by going to Preferences -> Groovy -> Compiler. If this does not work for you, follow the instructions in Compiler Switching within Groovy-Eclipse. Currently, the 1.7.x and 1.8.x streams are supported.

Q. Before we upgrade to the new groovy eclipse-plugin (V1.x), should we remove the prior Groovy plugin?

A. Yes, there may be some unexpected interactions between the two versions of the plugin. However, once you are using a 2.x version of the plugin, you do not need to uninstall before upgrading (see also previous question).

Q. I am using Aptana Studio and cannot install Groovy-Eclipse.

A. Aptana Studio does not come with the JDT (Java Development Tools) pre-installed. This is something that Groovy-Eclipse is built on top of and is therefore a requirement. You must therefore install JDT first. Here's how:

1. Go to your update manager
2. Select the Galileo - http://download.eclipse.org/releases/galileo update site (or the Helios or Indigo update site, whichever is relevant)
3. Choose the Category Programming Languages -> Java Development tools
4. Install
5. (No need to restart)
6. Now install Groovy
7. Restart and things should be working

Q. I am running Linux and cannot install Groovy-Eclipse.

You may be running into GRECLIPSE-498. Make sure that you are using an Eclipse version downloaded from Eclipse.org, and not one that has come from your Linux distribution's package manager. Using a repackaged Eclipse may not work because when Linux distributions repackage Eclipse, they typically change the feature structure so that it is not possible to install the JDT feature patch.

Feature set

Q. Why is groovy-aware syntax highlighting not working?

A. The most likely cause for this is that you are editing your groovy file in a Java editor. Look at the editor icon. Is it a groovy icon or a java icon? If it is a Java icon, you can change to a groovy editor like this:

1. close the editor.
2. right click on the file
3. Select Open With -> Groovy Editor

Also, please note that lines with multiple slashes are sometimes interpreted as regular expressions when they shouldn't be (see GRECLIPSE-124).

Q. What is the theme of the first release?

A. See the New and Noteworthy for a list of features.

Q. Can I use this compiler on the command line?

A. We have not yet released the batch compiler for Groovy, but we expect to do so for the 2.0.1 release (see the Groovy-Eclipse Roadmap).

Q. Does this release support joint Groovy-Java projects?

A. Yes. You may have references between Groovy and Java files within or across projects.

Q. Does it support incremental compilation?

A. Yes. If you make small changes to either java or groovy files in a mixed project and save them, they will be compiled together with any directly affected by the change.

Q. Does it support JUnit?

A. Yes. All of your Groovy unit tests can be run using the standard JUnit launcher.

Q. Is there any integration with Maven?

A. Yes. There are two options: GMaven and the groovy-eclipse-compiler. Each has different advantages. See the full discussion at Groovy-Eclipse compiler plugin for Maven.

Q. Does it support the AST transformations included with Groovy?

A. Yes. Although any changes introduced by the transformations (eg. new methods) are not currently visible to code in .java files, so you will not get code completion on them or be able to directly write code in .java files to call them. However, these methods will be available via content assist in Groovy files.

Q. Does it support custom AST transformations?

A. Custom Ast transforms are supported and will be picked up from the project classpath during a build. If you wish to define and consume transforms within the same project, there is a restriction: the transform must be written in Java (not groovy) and it cannot have dependencies on groovy source in that same project. This restriction exists because Eclipse does not have a notion of separate source folders that can be separately compiled - in eclipse the source within all source folders is compiled together.

Q. What can I expect in terms of code assist, navigation, debug and other eclipse facilities?

A. Content assist and navigation are available and take advantage of Groovy-Eclipse's inferencing engine and so can provide fairly sophisticated results. Many other Eclipse facilities, such as setting breakpoints, refactoring and code formatting are available. Debug support is also available. Please see the Groovy-Eclipse Roadmap for a discussion on what to expect in the future (and please leave your own comments).

Q. How does debugging work?

A. Basic debugging works as it does in Java. You can set a line breakpoint by double clicking on the gutter marker next to where you want the breakpoint to go. You can right click to enable/disable the breakpoint. You can edit the breakpoint properties to make it a conditional breakpoint (conditions must be specified in Java and may require explicit casting). The instruction pointer will stop at breakpoints, and you can use the standard step into, step over, and step outcommands from Java. The variables window will show all variables that are in scope, just like when in Java. However, you will find that some of the runtime types of the variables are not what you may expect. For example, variables declared inside closures will be of type groovy.lang.Reference. Method entry and class loading breakpoints are not yet supported.

If you want a more complete debugging experience, you should install the Grails Tooling, which also includes some enhanced debugging capabilities for Groovy code. Grails Tooling provides support for evaluating code snippets in the context of the currently executing program. This allows the display view, the expressions view, and inline evaluations to work as expected. See New Groovy Debug Support in STS 2.5.1.

Q. Where is the Groovy Console?

A. The Groovy Console feature of the old plugin has been removed. So far, no one has asked for this functionality. If you would like to see it, raise a jira issue (see below) for it or mention it on the mailing list (see below).

Q. When I load a Groovy file, why is the outline view blank?

A. The outline receives structural information from the Groovy compiler. If there is a syntax problem that is preventing the compiler from parsing the Groovy file, then the outline view will remain empty until because the compiler cannot provide any information about the file's contents.

Q. Where can I get Grails support?

A. Grails support is not included in this plugin. However, it is available in the SpringSource Tools Suite (STS). You can download STS here. For a basic description of how to install Grails into STS, create a new project or import an existing project into STS, here is a tutorial This tutorial also covers how to use the pop-up grails console to invoke grails commands.

Q. Whoa, those editor colors are psychedelic - can I have Java colours?

A. To switch to Java colors, open the eclipse preferences, go to Groovy > Editor and click 'Copy Java Color Preferences'

Troubleshooting

Q. Help! The Eclipse installer (p2) won't let me Groovy-Eclipse!

A.Groovy-Eclipse cannot be installed into a read-only directory. The Eclipse installation must be writable by the current user. *Shared installs* are not supported. This is because Groovy-Eclipse ships with a feature patch for JDT and feature patches cannot be installed into shared installs. This includes protected directories on Windows, like `C:\Program Files`. For more information on this problem see Eclipse Bug 395516.

As a workaround, you can temporarily provide write access to the current user in the Eclipse directory, restart Eclipse, and install Groovy-Eclipse. Once Groovy-Eclipse is installed, you no longer need write access to the install location.

Q. Help! The Eclipse installer (p2) won't let me upgrade my plugin!

It is likely that you are hitting this Eclipse bug, which should be fixed in Eclipse 3.6 and beyond.

Try the following:

* Use the *Install new Software* menu item instead of *Check for updates*.
* Uninstall your current version of the plugin and then reinstall the new version.

Q. I am getting a MissingPropertyException when running inside of Eclipse, but not when I run outside. What gives?

A. You could be hitting this issue: http://jira.codehaus.org/browse/GRECLIPSE-664.

It could be that you have the Groovy Libraries on the classpath twice. This may happen, for example, if you originally imported a Maven project and then added the Groovy Nature separately. The solution is to remove one version of the Groovy Libraries from your classpath.

Raising issues and asking questions

Q. Where should I report issues?

A. http://jira.codehaus.org/browse/GRECLIPSE
Be sure to specify version 2.0 of the plugin.

Q. Where can I raise questions?

A. Subscribe to the groovy eclipse plugin user's mailing list:
http://xircles.codehaus.org/lists/eclipse-plugin-user@groovy.codehaus.org/subscribe

There is also a developer's mailing list:
http://xircles.codehaus.org/lists/eclipse-plugin-dev@groovy.codehaus.org/subscribe

Q. What are the open bugs?

A. See below:

com.atlassian.confluence.macro.MacroExecutionException: The URL filter is not available to you, perhaps it has been deleted or had its permissions changed