

SpringSource Tool Suite 2.7.0.M2

- New and Noteworthy -

Martin Lippert

2.7.0.M2

June 13, 2011

Updated for 2.7.0.M2

ENHANCEMENTS – 2.7.0.M2

General Updates

Memory Settings

We raised the default memory settings for running STS slightly from a max. heap size of 768m to 1024m and from a max. size for the PermGen of 256m to 384m. If you would like to use different values, you can easily tweak them in your sts.ini file of your STS installation.

Spring Development Tools

Cloud Foundry Support

Since the last release we have built on top of the Cloud Foundry integration by adding the functionality to change your Cloud Foundry account password from within STS.

▼ Account Information

Email:

Password:

URL:

We have also added a server status section in the Cloud Foundry server editor to display whether it is connected to your Cloud Foundry account and you can connect or disconnect from the editor.

▼ Server Status

VMware Cloud Foundry: Connected

It is now also possible to change memory allocation to an application once it is deployed. Note that it does not take effect till the application is restarted.

General

Name:

Mapped URLs:

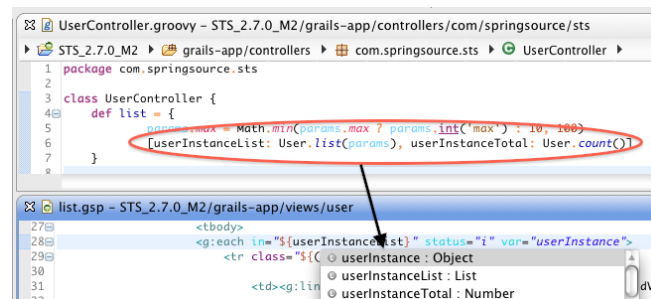
Memory limit: Change is not updated until application restarts

Instances:

Groovy & Grails Development Tools

GSP content assist now recognizes arguments coming from controller actions

In the screenshot below, you can see that the controller action (list) returns a map of two elements. The map values are available as variables in the associated GSP. You can see that in the content assist list. Also, STS makes a best effort to infer the types of all of these arguments.



GSP content assist now recognizes empty tags

As described here (<http://grails.org/Doc+Comments+for+GSP+tags>), it is now possible to specify that your custom GSP tags should be 'empty' when being applied in content assist. This means that your tag will be filled in like this:

```
<g:myTag />
```

instead of like this:

```
<g:myTag></g:myTag>
```

To do so, simply create a javadoc for your custom tag and ensure that the @emptyTag attribute is somewhere in the Javadoc on its own line.

Grails 1.4-M2 will be taking advantage of this for all of its standard tags (eg- def, set, etc).

Improved GSP content assist inside of scriptlets

We have made many small enhancements to content assist in GSPs, especially when inside of groovy scriptlets. You will have a much smoother experience when editing GSP files.

Grails 1.4 Support

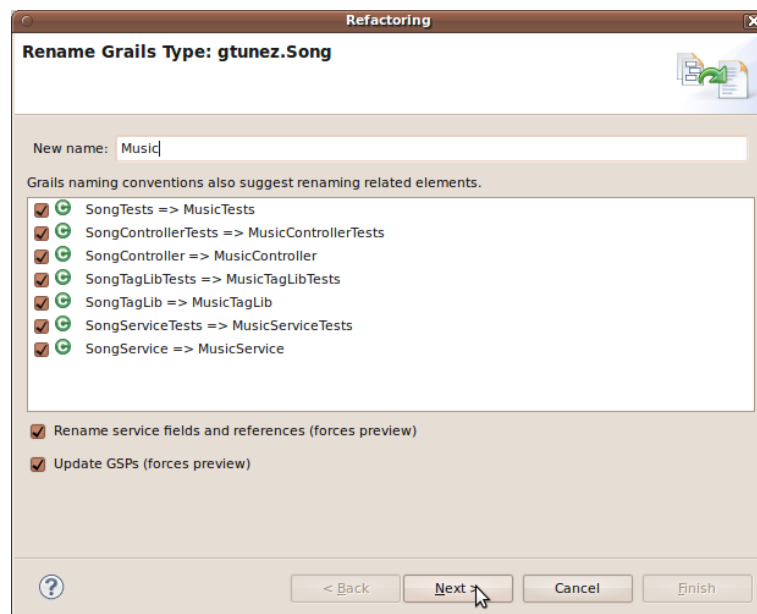
Greclipse and STS Grails tooling can now work with Grails 1.4.M1.

Grails Aware Rename Type Refactoring

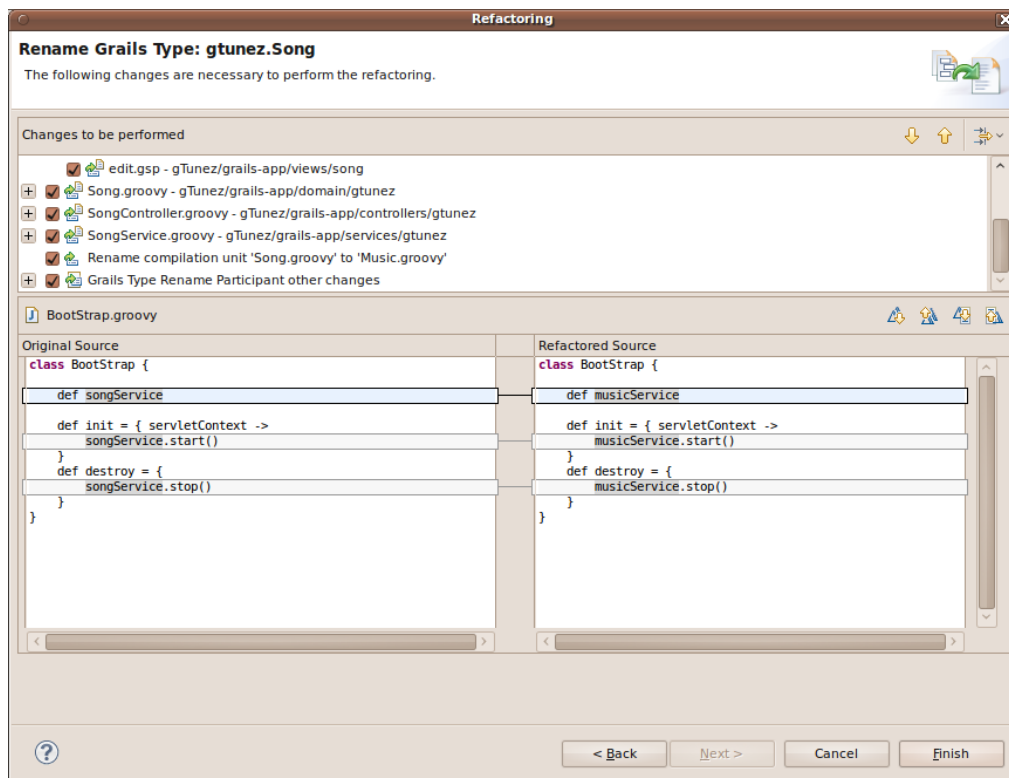
When a rename type refactoring is invoked from the Groovy Editor or the Project Explorer in the Grails perspective the refactoring will be 'Grails Aware'. A Grails specific rename type wizard will popup allowing the user to enable or disable Grails specific support:

- Grails Type Awareness:

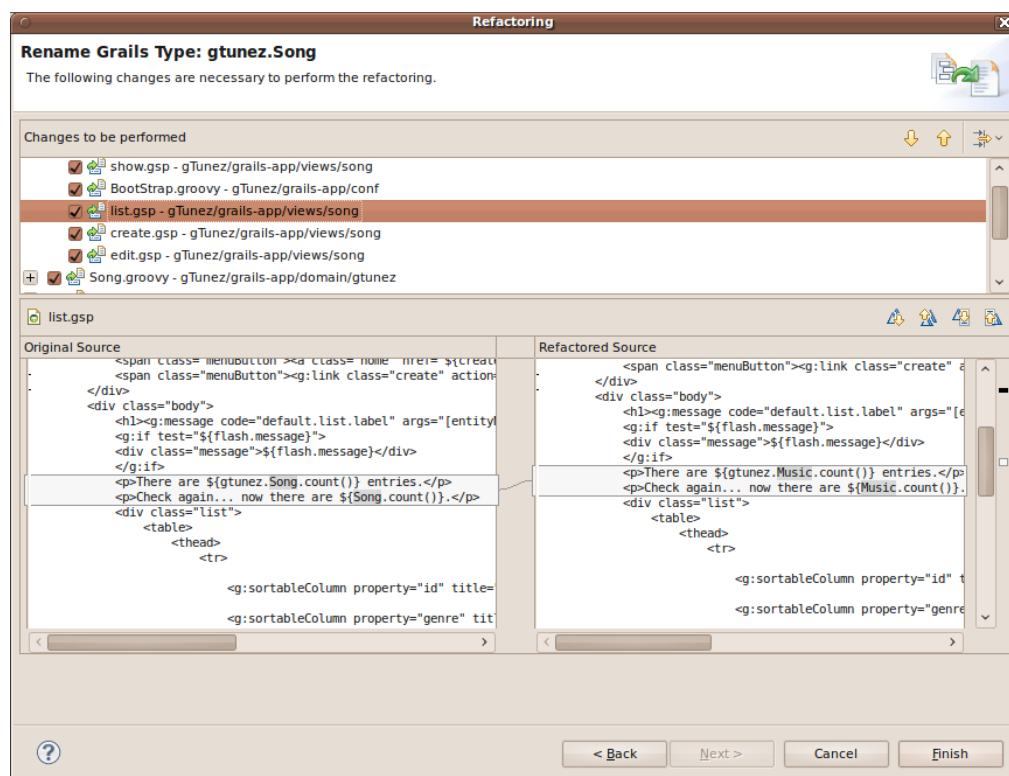
- Rename related types (for example when renaming a Domain class also rename associated Controller, Test and Service classes).
- Service Awareness:
 - Updating autowired service fields when a Service class is being renamed
- GSP Awareness:
 - Updating references to the renamed type(s) inside of GSP files.
 - Renaming the associated views folder when a Controller class is being renamed.



Grails Rename Type Wizard Page One



Grails Rename Type – Preview



Grails Rename Type – GSPs

Groovy-Eclipse Compiler

There is now a milestone version of the groovy-eclipse-compiler plugin for maven available. This allows you to compile your groovy-maven projects using Groovy-Eclipse outside of the IDE. Using this compiler plugin avoids many of the stub generation issues of GMaven and also compiles your projects more quickly.

For documentation on how to use the groovy-eclipse-compiler, see:

<http://groovy.codehaus.org/Groovy-Eclipse+compiler+plugin+for+Maven>

This page also includes a link to an archetype project that can get you started quickly with using the compiler plugin.

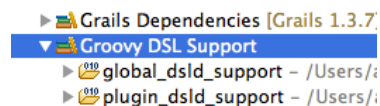
Global location for DSLDs

Groovy-Eclipse now supports a global location to place your DSLDs (see

<http://blog.springsource.com/2011/05/08/better-dsl-support-in-groovy-eclipse/>

for a description of DSLDs).

You will notice that there is a new classpath container on all of your Groovy/Grails projects:



This classpath container contains all of the global and plugin DSLDs for your projects.

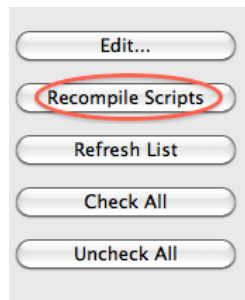
If you want a DSLD file to be applicable to all of your Groovy projects in the workspace, then place your global DSLDs in the `~/.groovy/greclipse/global_dsl_support/dsl` directory. A package statement of `'package dsl'` is required. These DSLDs will appear in the `global_dsl_support` class folder.

DSLDS that are specific to a single version of Groovy-Eclipse are located in the `plugin_dsl_support` class folder.

Note: Eclipse 3.6 and earlier suffer from

https://bugs.eclipse.org/bugs/show_bug.cgi?id=305172

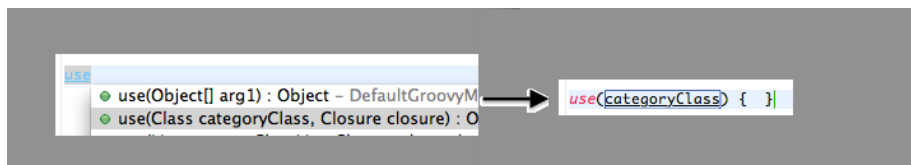
where these external classfolders will not be updated automatically. Instead, every time that you make a change to the `global_dsl_support` folder, you should go to the DSLD preferences page (Preferences -> Groovy -> DSLD) and click "Recompile Scripts":



In the coming weeks, we will be supplying several default DSLD files to handle the common builders and AST transforms inside of Groovy-Eclipse (such as Swing Builder, and HTML builder). These DSLDs will appear in the `plugin_dsl_support` folder.

Groovier way of handling closure arguments in content assist

When applying a content assist method proposal that has a closure as its last argument, Groovy-Eclipse will now automatically use the standard Groovy idiom of placing the closure expression outside of the parens. See, for example this screenshot:



Support for `$/ /$` multiline strings

Groovy-eclipse now recognizes `$/ /$` style strings that have been introduced in Groovy 1.8.

Gradle

Several bug fixes and minor enhancements since M1. For example:

- Reported problems with enabled/disabled state of Gradle menus - Fixed
- Gradle import wizard now avoids temporary build-path problems by properly sorting projects based on project dependencies before importing.
- Gradle classpath container state is now persisted to disk for faster IDE startup.

Fixed Bugs and Enhancement Requests

Here is a full list of resolved bugs and enhancement requests for the 2.7.0.M2 release:

[https://issuetracker.springsource.com/secure/IssueNavigator.jspa?reset=true&jqlQuery=project
in\('2.7.0.M2'\)AND status in\('Resolved,Closed'\)](https://issuetracker.springsource.com/secure/IssueNavigator.jspa?reset=true&jqlQuery=project%20in%20('2.7.0.M2')%20AND%20status%20in%20('Resolved','Closed'))

NEW AND NOTEWORTHY

www.springsource.com



ENHANCEMENTS – 2.7.0.M1

General Updates

vFabric tc Server 2.5.0

STS 2.6.1 ships with the just released vFabric tc Server developer edition 2.5.0. In addition to the updated package STS 2.6.1 added support for the new server type “vFabric tc Server 2.5”.

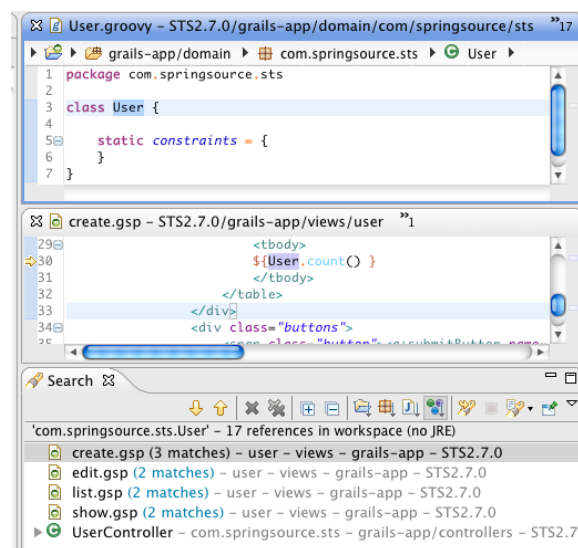
Spring Development Tools

There are no major new features implemented for the Spring tooling part of STS for 2.7.0.M1. Most of the effort was spent to fix minor bugs and to provide a great user experience for the Cloud Foundry Integration into STS - which we already shipped as extension to STS with version 2.6.0.SR1 on April 12th.

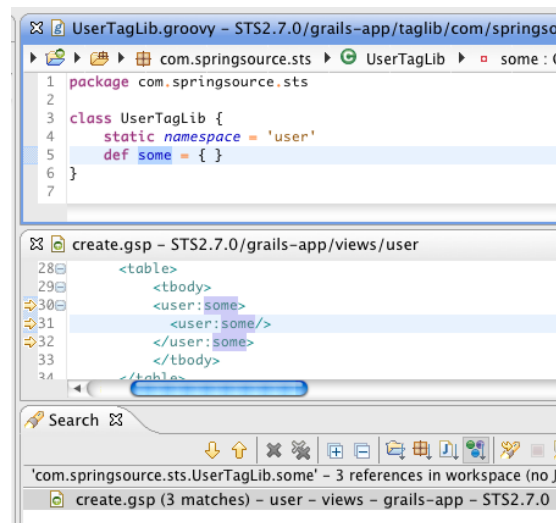
Groovy & Grails Development Tools

GSP Search

The Eclipse search engine will now look inside GSP files for search results. References inside of scriptlets will be found:

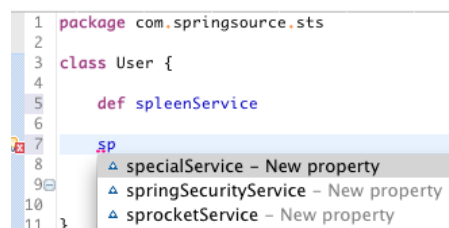


Also, references to tags will be found:



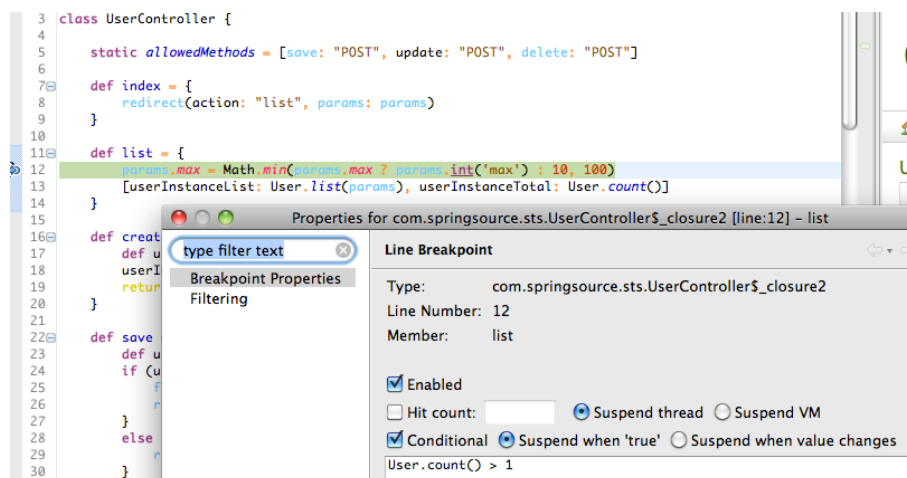
Service field declaration content assist

Content assist inside of Grails artifacts at a location where a new field can be declared will now display all available services as proposals:



Conditional breakpoints for Groovy code

Conditional breakpoints are now supported in Groovy files using groovy syntax:



This is built on top of the same infrastructure that the Groovy debug support uses. So, enhanced debug support must be activated and JDT Weaving must be enabled. See the blog post about Groovy Debug support in STS for more information:

<http://blog.springsource.com/2010/11/30/new-groovy-debug-support-in-sts-2-5-1/>.

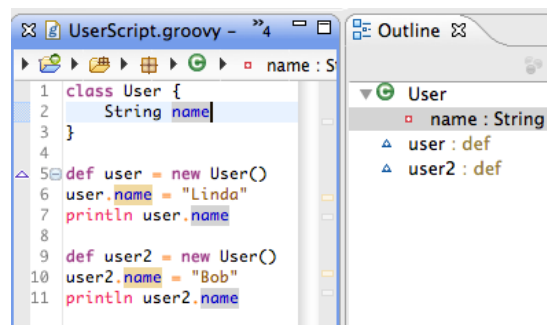
Groovy 1.8 support

Currently, Groovy-Eclipse supports Groovy 1.8.rc2. Groovy 1.8 final will be available shortly in a development snapshot of Groovy-Eclipse. Groovy 1.8 will not be installed by default with the rest of Groovy-Eclipse. Instead, it is available as a separate feature that can be installed from the Groovy-Eclipse update site. For information on how to do this, see Compiler Switching in Groovy-Eclipse:

<http://groovy.codehaus.org/Compiler+Switching+within+Groovy+Eclipse>

Script outline view

When working with scripts, the outline view displays contents in a more Groovy-like way. The run, and main methods are hidden. Instead, script variable declarations are displayed.

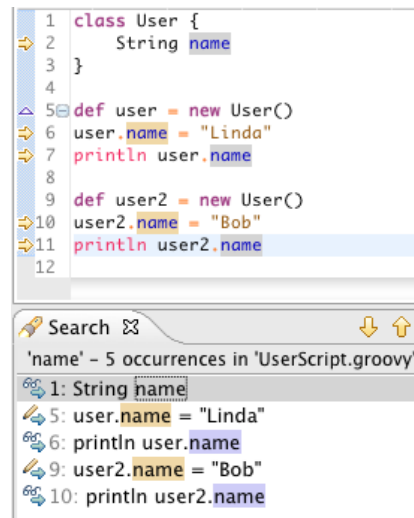


The enhanced outline view is part of Groovy-Eclipse and is available for third party plugins to use their own outline extender. For more information, see the feature request here:

GRECLIPSE-1029: <http://jira.codehaus.org/browse/GRECLIPSE-1029>

Read access and Write access differentiation when searching

When doing a find references in the Groovy editor (CTRL-Shift-U on Windows/Linux), read occurrences are highlighted differently from write occurrences:



Better inferencing for Generics

Groovy-Eclipse now recognizes more complex forms of generics to help seed type inferencing. For example, in the following code, Groovy-Eclipse can determine that the second call to "get()" is declared by the WeakReference class and returns a java.lang.Class object:

```

3 import java.lang.ref.WeakReference
4 class MyMap extends HashMap<String,WeakReference<Class>> { }
5 MyMap cache;
6 cache.get("name").get().canonicalName
7

```

The tooltip for the second call to "get()" shows the result: **Class java.lang.ref.Reference.get()**

For more information, see GRECLIPSE-997: <http://jira.codehaus.org/browse/GRECLIPSE-997>

Gradle

STS 2.7.M1 includes an initial, basic set of features to support working with Gradle projects inside of STS. The following list of features are provided:

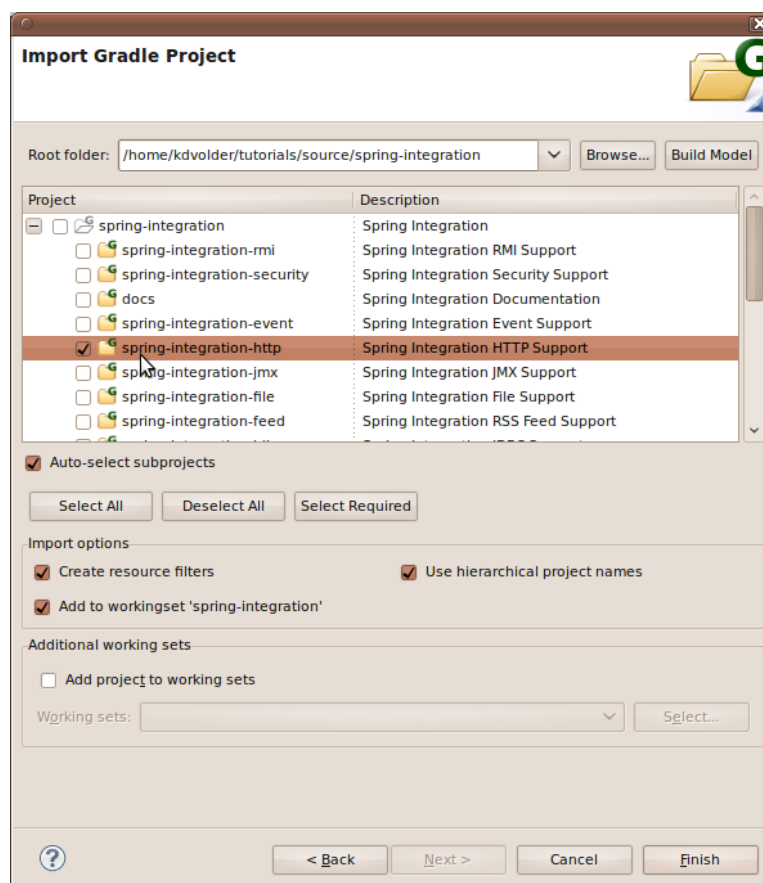
First set of features

- Supports working with nested Gradle "multi projects":
 - mapping hierarchical structure onto a flat set of "linked" Eclipse projects with appropriate automatically configured resource filters.
 - flexible name mapping, allows user to adopt two different default name mapping schemes and also rename individual projects in the workspace without breaking the tools.
- Gradle Import Wizard:
 - ease importing existing Gradle (Multi) Projects with a number of subprojects.
 - ease the importing of a subset of "interesting projects", without breaking project dependencies.
 - allow incrementally importing additional projects after initially importing only a subset.

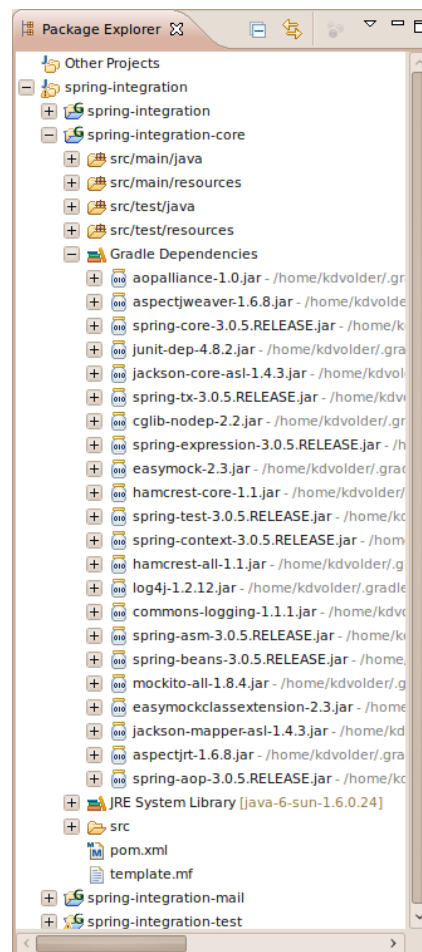
- Manage Gradle dependencies: (so projects can "build/compile" using the Eclipse JDT tools)
 - Classpath container that manages Gradle Jar dependencies on the Eclipse classpath
 - configure source folders automatically
 - configure project dependencies automatically
- Task Execution UI allows users to...
 - create, store, edit, manage and execute lists of Gradle tasks as Eclipse launch configurations from within STS.
 - Since this leverages the standard Eclipse launching framework, you can also
 - relaunch from a list of favorites
 - relaunch from a "recent history" list
 - store launch configurations in files that can be shared via a code repository like SVN, CVS or git.

For more information see the documentation:

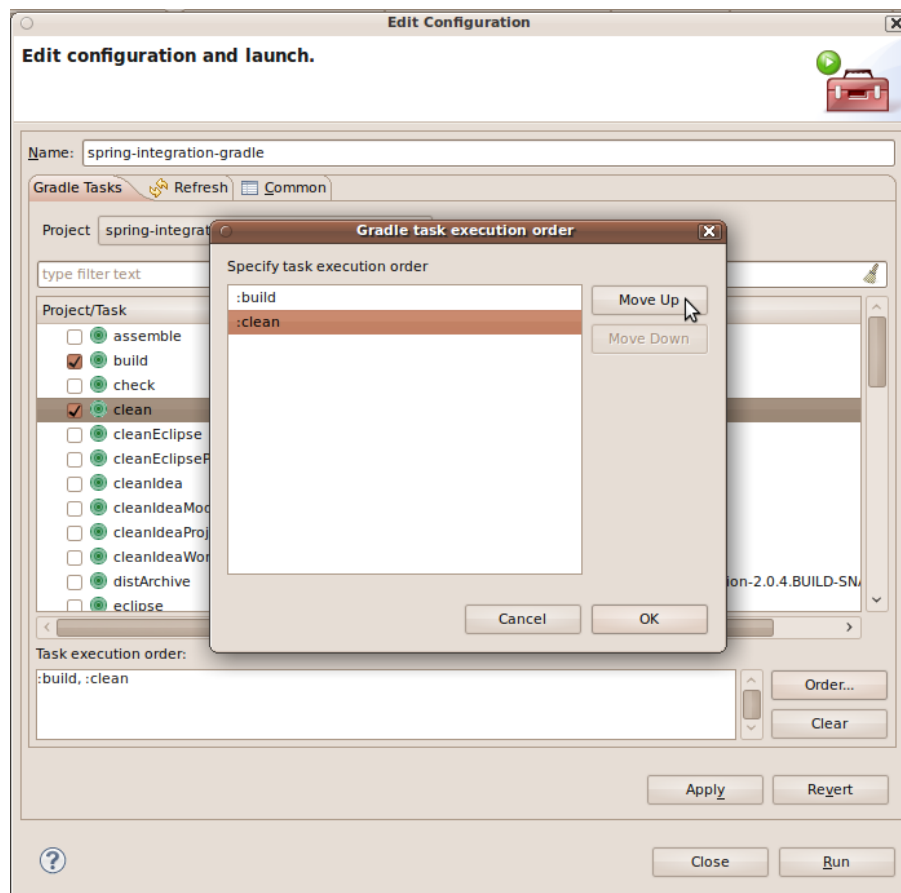
<http://static.springsource.org/sts/docs/2.7.0.M1/reference/html/gradle/index.html>



Gradle Import Wizard



Gradle Classpath Container



Gradle Launch Configuration Editor

Fixed Bugs and Enhancement Requests

Here is a full list of resolved bugs and enhancement requests for the 2.7.0.M1 release:

[https://issuetracker.springsource.com/secure/IssueNavigator.jspa?reset=true&jqlQuery=project
 +=+STS+AND+fixVersion+=+"2.7.0.M1"+AND+status+in+%28Resolved,+Closed%29](https://issuetracker.springsource.com/secure/IssueNavigator.jspa?reset=true&jqlQuery=project+%3D+STS+AND+fixVersion+%3D+%272.7.0.M1%27+AND+status+in+%28Resolved,+Closed%29)