

# SpringSource Tool Suite 2.7.0.M1

- New and Noteworthy -

Martin Lippert

2.7.0.M1

May 3, 2011

Updated for 2.7.0.M1

## 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".

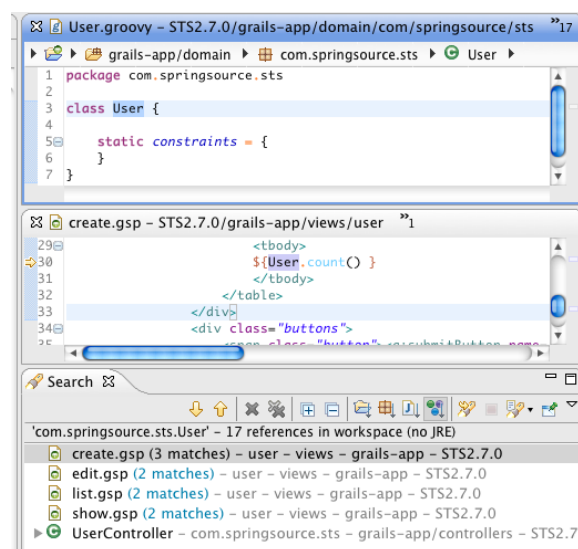
### Spring Development Tools

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 12<sup>th</sup>.

### 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:

```

1 package com.springsource.sts
2
3 class UserTagLib {
4     static namespace = 'user'
5     def some = { }
6 }
7
28 <table>
29 <tbody>
30 <user:some>
31 </user:some/>
32 </user:some>
33 </tbody>
34 </table>

```

Search: 'com.springsource.sts.UserTagLib.some' - 3 references in workspace (no J...)

create.gsp (3 matches) - user - views - grails-app - STS2.7.0

## 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:

```

1 package com.springsource.sts
2
3 class User {
4
5     def spleenService
6
7     sp
8     ▲ specialService - New property
9     ▲ springSecurityService - New property
10    ▲ sprocketService - New property
11 }

```

## Conditional breakpoints for Groovy code

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

```

3 class UserController {
4
5     static allowedMethods = [save: "POST", update: "POST", delete: "POST"]
6
7     def index = {
8         redirect(action: "list", params: params)
9     }
10
11    def list = {
12        params.max = Math.min(params.max ? params.int('max') : 10, 100)
13        [userInstanceList: User.list(params), userInstanceTotal: User.count()]
14    }
15
16    def create {
17        def u = userInstance
18        return u
19    }
20
21
22    def save {
23        def u = userInstance
24        if (u) {
25            u.save()
26        }
27    }
28    else {
29        u.save()
30    }
}

```

Properties for com.springsource.sts.UserController\$\_closure2 [line:12] - list

type filter text

Breakpoint Properties

Filtering

Line Breakpoint

Type: com.springsource.sts.UserController\$\_closure2

Line Number: 12

Member: list

Enabled

Hit count:   Suspend thread  Suspend VM

Conditional  Suspend when 'true'  Suspend when value changes

User.count() > 1

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/>.

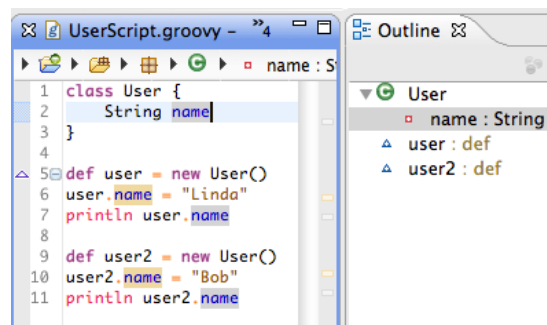
### Groovy 1.8 support

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:

```

1 class User {
2     String name
3 }
4
5 def user = new User()
6 user.name = "Linda"
7 println user.name
8
9 def user2 = new User()
10 user2.name = "Bob"
11 println user2.name
12

```

Search 🔍

'name' - 5 occurrences in 'UserScript.groovy'

- 1: String name
- 5: user.name = "Linda"
- 6: println user.name
- 9: user2.name = "Bob"
- 10: println user2.name

## 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

```

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:

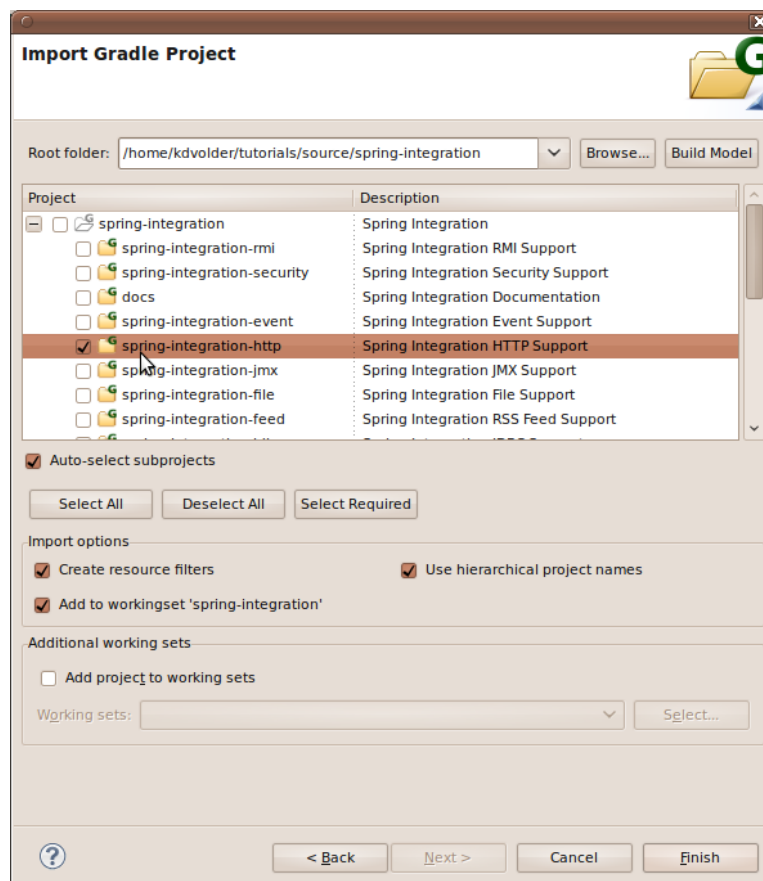
### First set of features

- 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.

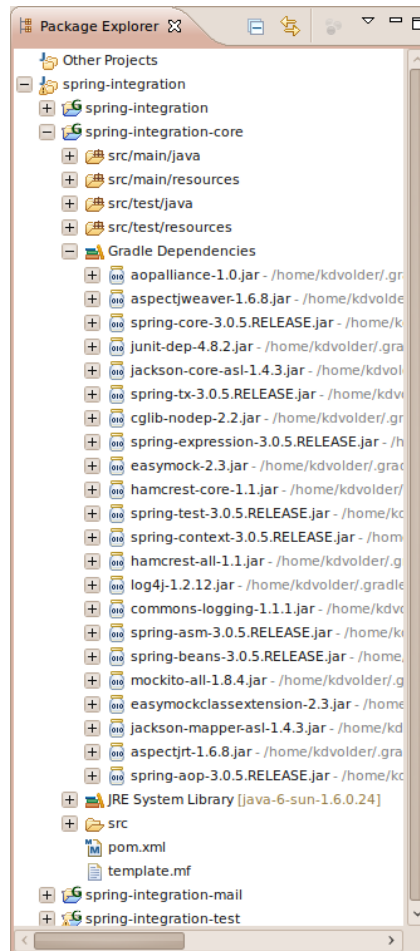
- Manage Gradle dependencies: (so projects can "build/compile" using the Eclipse JDT tools)
  - Classpath container that manages Gradle Jar dependencies on the Eclipse classpath
  - configure source folders automatically
  - configure project dependencies automatically
- Task Execution UI allows users 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:

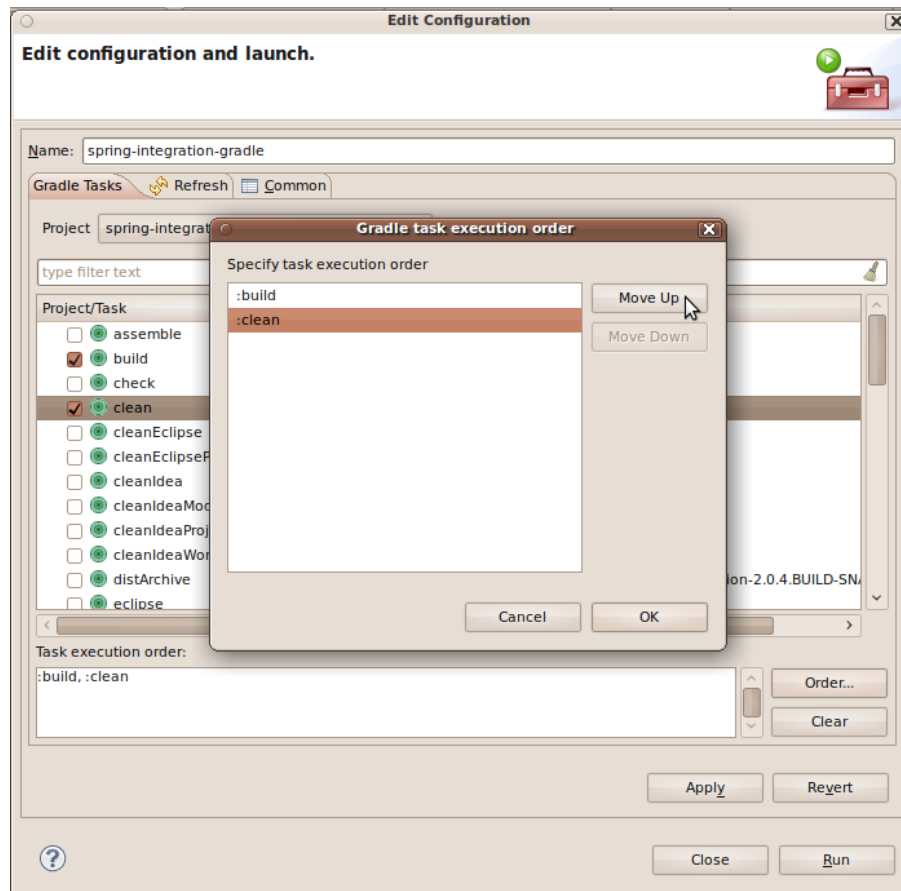
<http://static.springsource.org/sts/docs/2.7.0.M1/reference/html/gradle/index.html>



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+'2.7.0.M1'+AND+status+in+%28Resolved,+Closed%29)