SpringSource Tool Suite 2.6.1.SR1

- New and Noteworthy -

Martin Lippert 2.6.1.SR1 May 18, 2011 Updated for 2.6.1.SR1

UPDATES AND ENHANCEMENTS – 2.6.1.SR1

General Updates

Spring Roo 1.1.4

STS 2.6.1.SR1 ships with the just released version 1.1.4 of Spring Roo.

UPDATES AND ENHANCEMENTS – 2.6.1

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".

UPDATES AND ENHANCEMENTS – 2.6.0.SR1

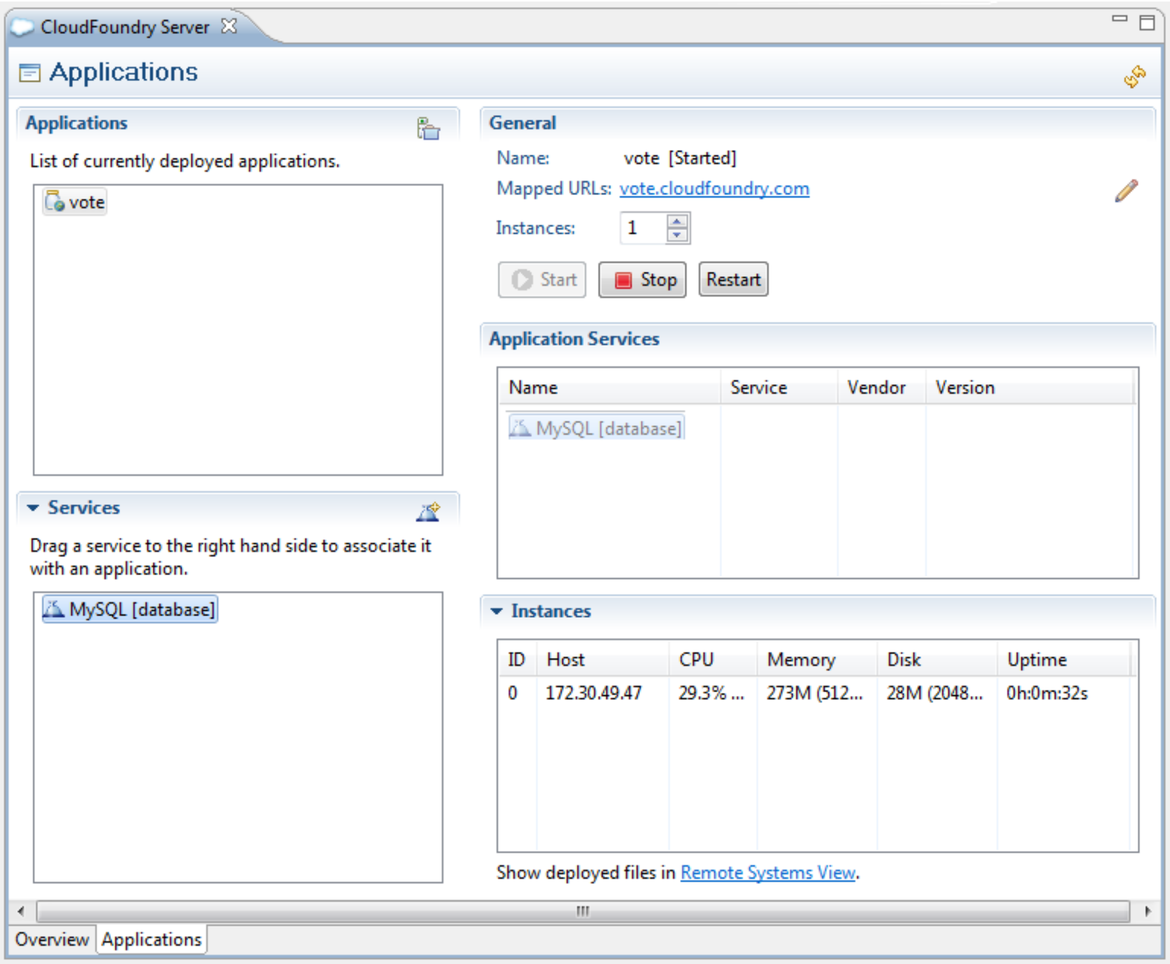
General Updates

Spring Roo 1.1.3

STS 2.6.0.SR1 ships with the just released version 1.1.3 of Spring Roo.

Cloud Foundry Support

STS now provides direct support for the just released Cloud Foundry platform. Users can install the Cloud Foundry plugin for STS from the Extension install of the Dashboard and deploy their apps via Drag & Drop into the cloud, manage services right from within STS, access the files of the running Cloud Foundry server.and much more.



List of currently deployed applications.

vote

Drag a service to the right hand side to associate it with an application.

MySQL [database]

Name: vote [Started]

Mapped URLs: vote.cloudfoundry.com

Instances: 1

Name	Service	Vendor	Version
MySQL [database]			

ID	Host	CPU	Memory	Disk	Uptime
0	172.30.49.47	29.3% ...	273M (512...	28M (2048...	0h:0m:32s

Show deployed files in Remote Systems View.

ENHANCEMENTS – 2.6.0

General Updates

Spring Roo 1.1.2

STS 2.6.0 ships with the just released version 1.1.2 of Spring Roo.

Groovy 1.7.8

The Groovy & Grails support of STS now ships with Groovy 1.7.8.

Grails 1.3.7

STS 2.6.0 supports Grails 1.3.7 and ships that Grails version from the Dashboards extensions install.

Eclipse Helios SR2 (3.6.2)

STS is now based on the freshly released second service release of Eclipse Helios and all packages of STS ship with the Eclipse IDE for Java EE Developers.

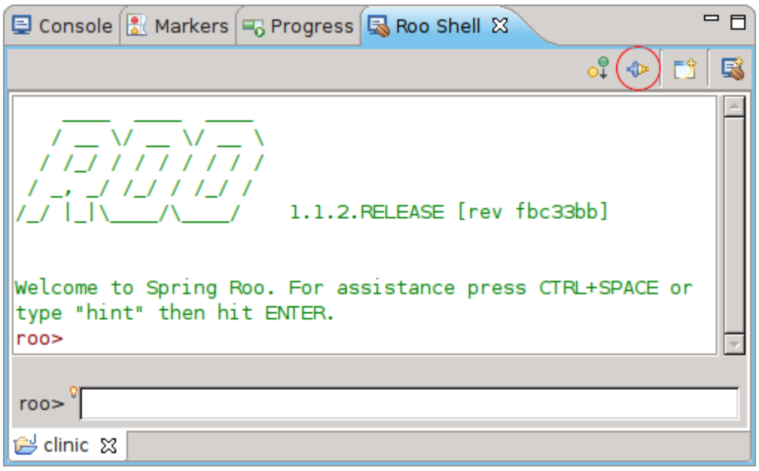
dm Server Tooling

The pre-build STS packages no longer contain the dm Server tooling by default. But you can easily add them to your installation via the Dashboards extension install.

Spring Development Tools

Roo Add-on Manager

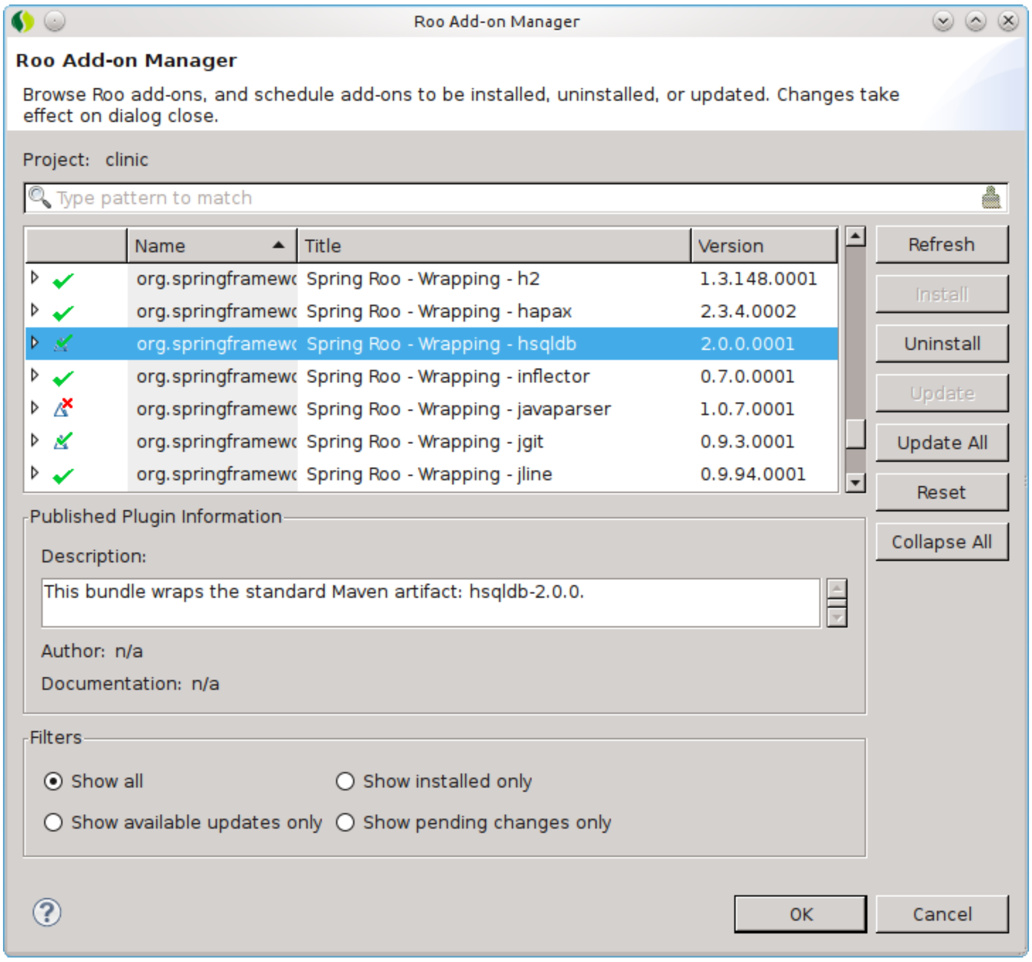
The new Roo Add-on Manager provides a convenient interface for managing Roo add-ons. It is accessible from the toolbar of the Roo Shell view and from the Spring context menu of any Roo project.



```
    ____  ____  ____
   / __ \/ __ \/ __ \
  / /_/ / / / / / / /
 / _, _/ /_/ / /_/ /
/_/ |_|\____/\____/    1.1.2.RELEASE [rev fbc33bb]


Welcome to Spring Roo. For assistance press CTRL+SPACE or
type "hint" then hit ENTER.
roo>
```

```
roo>
```

The manager dialog displays a list of installed and available add-ons. When an add-on is selected additional details are shown. The list can be filtered and searched to quickly find specific add-ons. The button bar on the right allows to select add-ons for installation, update or uninstallation. On finish all selections are applied and progress is reported in the Roo shell.



Roo Add-on Manager

Browse Roo add-ons, and schedule add-ons to be installed, uninstalled, or updated. Changes take effect on dialog close.

Project: clinic

Type pattern to match

	Name	Title	Version
✓	org.springframewc	Spring Roo - Wrapping - h2	1.3.148.0001
✓	org.springframewc	Spring Roo - Wrapping - hapax	2.3.4.0002
✓	org.springframewc	Spring Roo - Wrapping - hsqldb	2.0.0.0001
✓	org.springframewc	Spring Roo - Wrapping - inflector	0.7.0.0001
✗	org.springframewc	Spring Roo - Wrapping - javaparser	1.0.7.0001
✓	org.springframewc	Spring Roo - Wrapping - jgit	0.9.3.0001
✓	org.springframewc	Spring Roo - Wrapping - jline	0.9.94.0001

Published Plugin Information

Description:

This bundle wraps the standard Maven artifact: hsqldb-2.0.0.

Author: n/a

Documentation: n/a

Filters

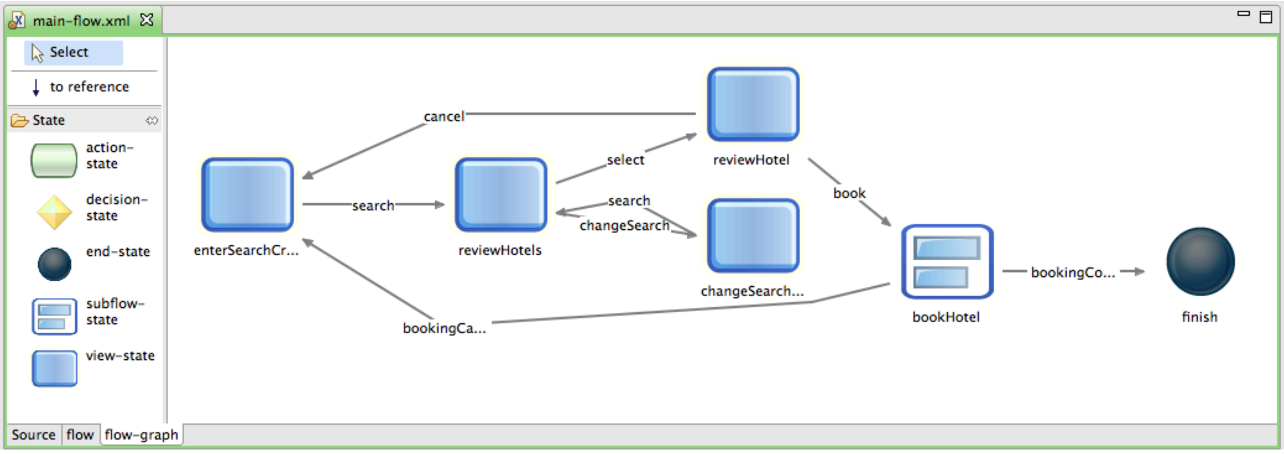
● Show all ○ Show installed only

○ Show available updates only ○ Show pending changes only

Note: The add-on manager requires Roo 1.1.2 or later and consent to the download terms of use. The status can be checked using the "download status" command in the Roo shell. If a plug-in fail to install due to a verification error the corresponding PGP key needs to be added to the trust key using the "pgp trust" or "pgp automatic trust" command.

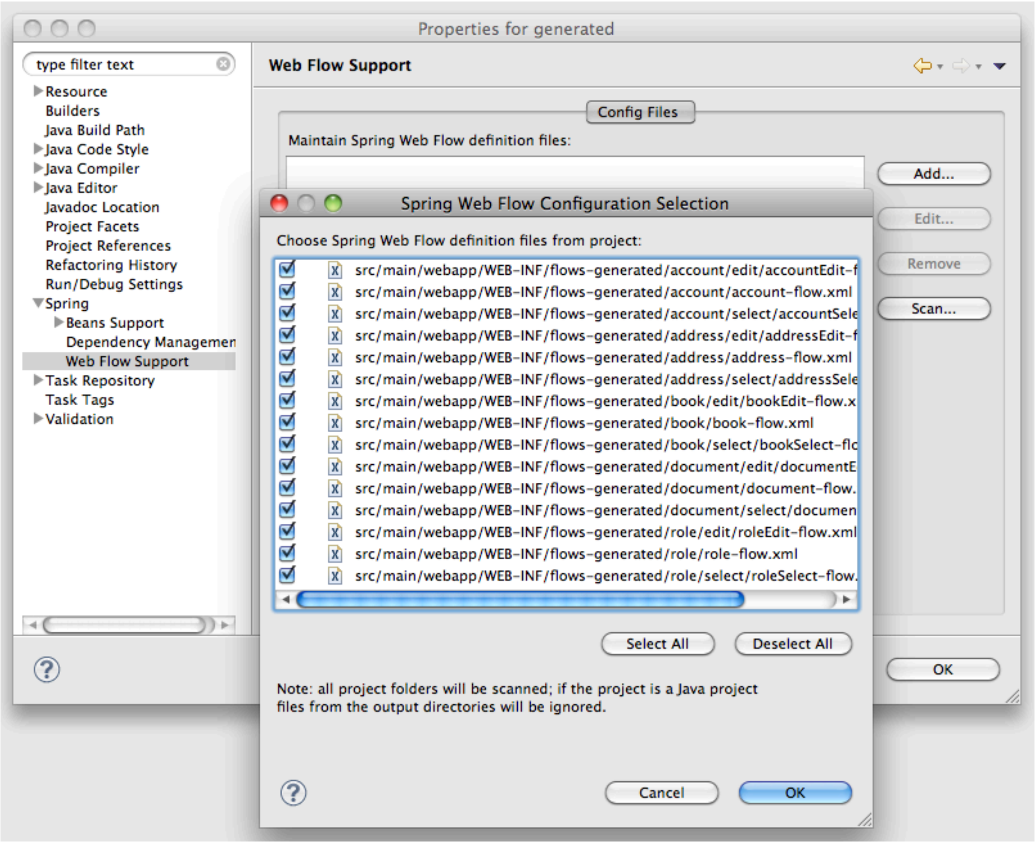
Spring Web Flow Visual Editing

STS 2.6 introduces a new visual editing mode for Spring Web Flow configuration files. The visual editor supports creation of all state types and transitions between states, and editing of state attributes can be done through the Properties view. The Spring Web Flow visual editor can be found on the flow-graph tab of the Spring Web Flow XML editor.



Scan for Spring Web Flow Files

The Web Flow Support properties page has added a Scan... button that will search the project for any Spring Web Flow definition files that are currently outside of the Web Flow model. Users can select from the resulting dialog which files they would like to add to the Web Flow model.



Maintain Spring Web Flow definition files:

Spring Web Flow Configuration Selection

Choose Spring Web Flow definition files from project:

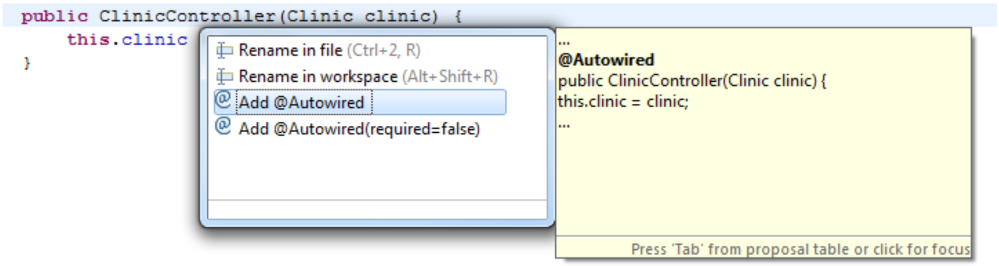
☑ src/main/webapp/WEB-INF/flows-generated/account/edit/accountEdit-f
☑ src/main/webapp/WEB-INF/flows-generated/account/account-flow.xml
☑ src/main/webapp/WEB-INF/flows-generated/account/select/accountSele
☑ src/main/webapp/WEB-INF/flows-generated/address/edit/addressEdit-f
☑ src/main/webapp/WEB-INF/flows-generated/address/address-flow.xml
☑ src/main/webapp/WEB-INF/flows-generated/address/select/addressSele
☑ src/main/webapp/WEB-INF/flows-generated/book/edit/bookEdit-flow.x
☑ src/main/webapp/WEB-INF/flows-generated/book/book-flow.xml
☑ src/main/webapp/WEB-INF/flows-generated/book/select/bookSelect-flo
☑ src/main/webapp/WEB-INF/flows-generated/document/edit/documentE
☑ src/main/webapp/WEB-INF/flows-generated/document/document-flow.
☑ src/main/webapp/WEB-INF/flows-generated/document/select/documen
☑ src/main/webapp/WEB-INF/flows-generated/role/edit/roleEdit-flow.xml
☑ src/main/webapp/WEB-INF/flows-generated/role/role-flow.xml
☑ src/main/webapp/WEB-INF/flows-generated/role/select/roleSelect-flow.

Note: all project folders will be scanned; if the project is a Java project files from the output directories will be ignored.

Content-Assist and Quick-Fixes for @Autowired and @Qualifier

It is becoming a common practice to configure spring applications using annotations. In this release we added some content assist and quick fix proposals to provider more support for editing these annotations.

@Autowired can now be quickly added to constructors, methods and fields by invoking quick fix on the declarations which brings up an option to add the @Autowired annotation.

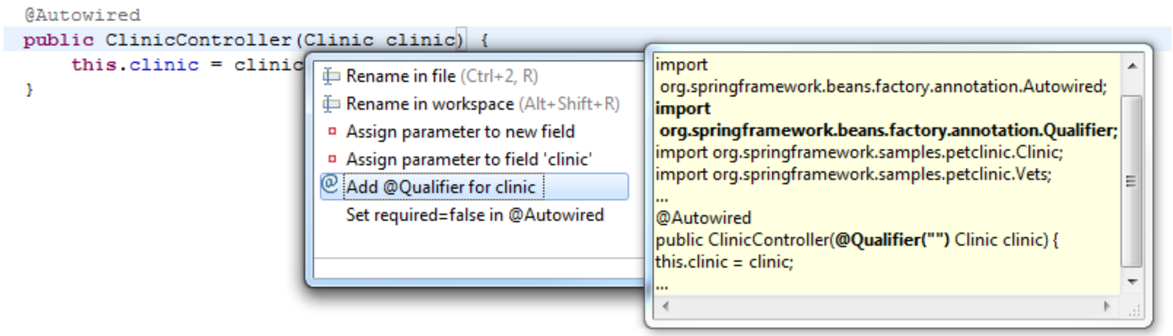


```
public ClinicController(Clinic clinic) {
    this.clinic
}
```

Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)
Add @Autowired
Add @Autowired(required=false)
Press 'Tab' from proposal table or click for focus

```
...
@Autowired
public ClinicController(Clinic clinic) {
this.clinic = clinic;
...
```

Autowiring by type may lead to multiple candidates. There is now quick fix support for adding @Qualifier annotation, as well as content assist for matching bean names.



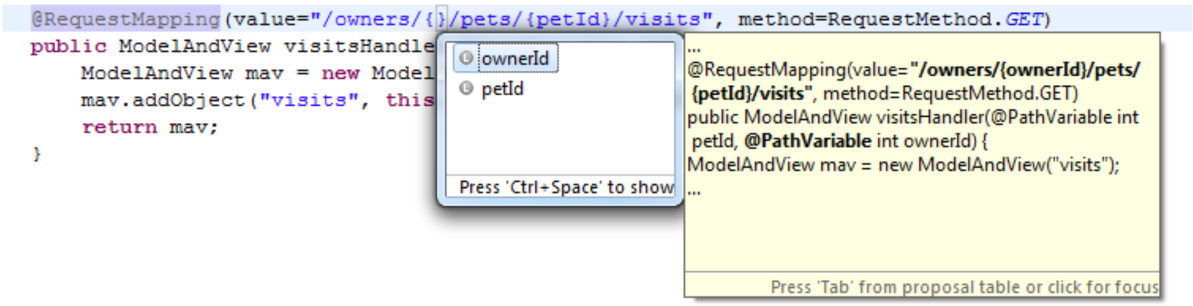
```
@Autowired
public ClinicController(Clinic clinic) {
    this.clinic = clinic
}
```

Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)
Assign parameter to new field
Assign parameter to field 'clinic'
Add @Qualifier for clinic
Set required=false in @Autowired

```
import
 org.springframework.beans.factory.annotation.Autowired;
import
 org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.samples.petclinic.Clinic;
import org.springframework.samples.petclinic.Vets;
...
@Autowired
public ClinicController(@Qualifier("") Clinic clinic) {
this.clinic = clinic;
...
```

Content-Assist and Quick-Fixes for @RequestMapping

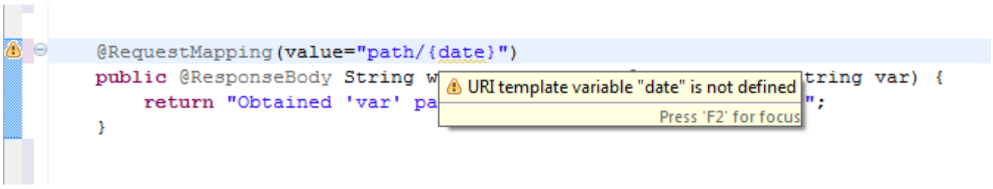
Another new feature is content assist for specifying URI template variable. When editing a @RequestMapping annotation on a method declaration, content assist is available for matching method parameters, as well as adding @PathVariable to the matching parameter to indicate that the parameter should be bound to the value of a URI template variable.



```
@RequestMapping(value="/owners/{}/pets/{petId}/visits", method=RequestMethod.GET)
public ModelAndView visitsHandle
    ModelAndView mav = new Model
    mav.addObject("visits", this
    return mav;
}
```

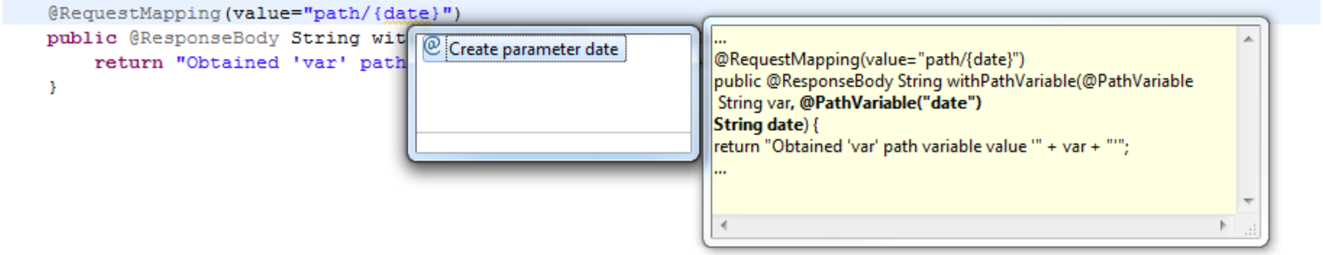
```
...
@RequestMapping(value="/owners/{ownerId}/pets/
{petId}/visits", method=RequestMethod.GET)
public ModelAndView visitsHandler(@PathVariable int
 petId, @PathVariable int ownerId) {
ModelAndView mav = new ModelAndView("visits");
...
```

We also added as you type validation which shows warnings for variable references that are not defined as well as corresponding quick fixes:



```
@RequestMapping(value="path/{date}")
public @ResponseBody String w                                  tring var) {
    return "Obtained 'var' pa                                  ";
}
```

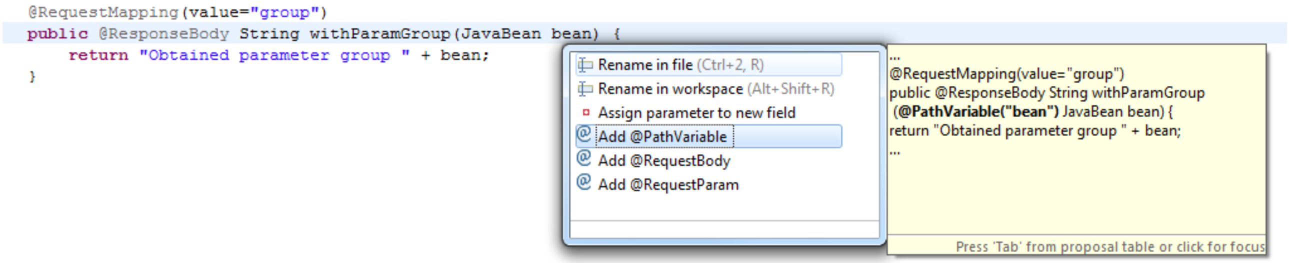
URI template variable "date" is not defined



```
@RequestMapping(value="path/{date}")
public @ResponseBody String wit
    return "Obtained 'var' path
}
```

```
...
@RequestMapping(value="path/{date}")
public @ResponseBody String withPathVariable(@PathVariable
 String var, @PathVariable("date")
String date) {
return "Obtained 'var' path variable value '" + var + "'";
...
```

Quick fixes are now available on method parameters to show basic annotations (@PathVariable, @RequestBody and @RequestParam) that can be added to the method parameter:



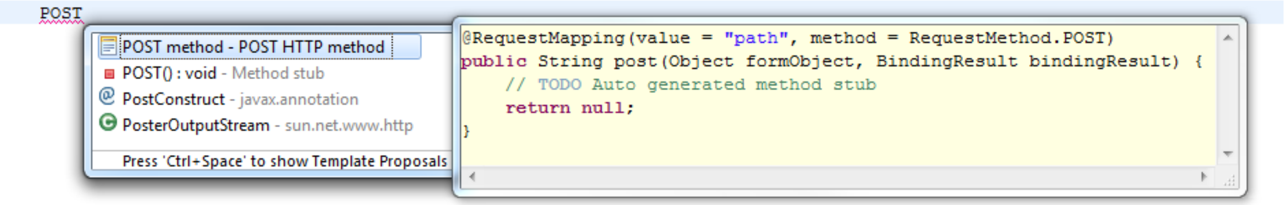
```
@RequestMapping(value="group")
public @ResponseBody String withParamGroup(JavaBean bean) {
    return "Obtained parameter group " + bean;
}
```

Rename in file (Ctrl+2, R)
Rename in workspace (Alt+Shift+R)
Assign parameter to new field
Add @PathVariable
Add @RequestBody
Add @RequestParam

```
...
@RequestMapping(value="group")
public @ResponseBody String withParamGroup
 (@PathVariable("bean") JavaBean bean) {
return "Obtained parameter group " + bean;
...
```

Press 'Tab' from proposal table or click for focus

We also added templates for creating basic method with annotations, like GET, POST and PUSH request mapping methods as well as @InitBinder and @ExceptionHandler methods. Starting typing the first few characters of the method you would like to add (for example ini... for @InitBinder method) and then hit CTRL + SPACE and you can see the template in the pop up dialog.



POST

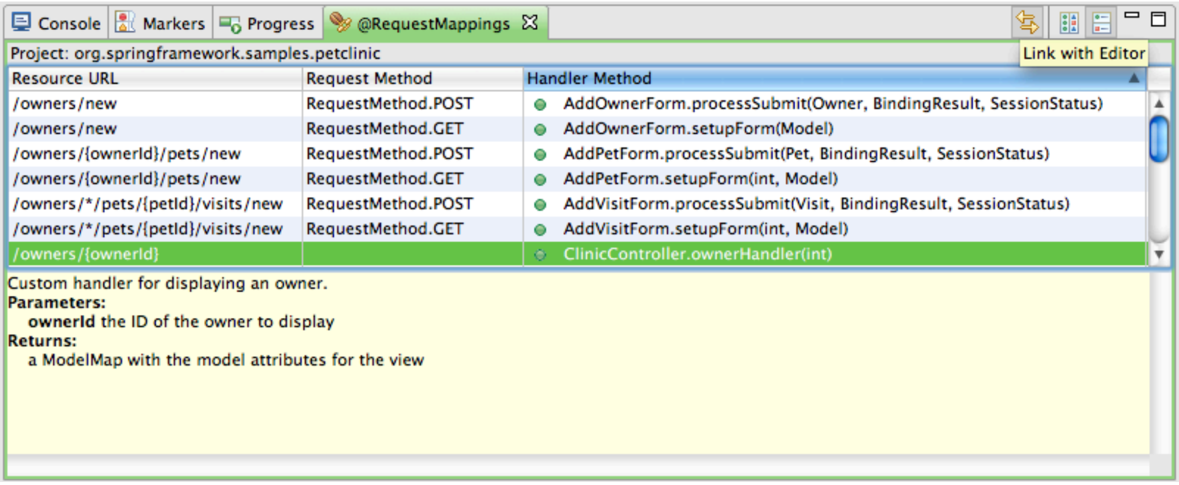
- POST method - POST HTTP method
- POST() : void - Method stub
- PostConstruct - javax.annotation
- PosterOutputStream - sun.net.www.http

Press 'Ctrl+Space' to show Template Proposals

```
@RequestMapping(value = "path", method = RequestMethod.POST)
public String post(Object formObject, BindingResult bindingResult) {
    // TODO Auto generated method stub
    return null;
}
```

RequestMappings view improvements

All columns in the RequestMappings view are now sortable, so users can sort the view according to URL path, RequestMethod type, or Class#method signature. Additionally, the RequestMappings view now supports Link with Editor mode. When the Link with Editor button is toggled, the view will automatically populate with all RequestMappings for the project containing the selected editor.



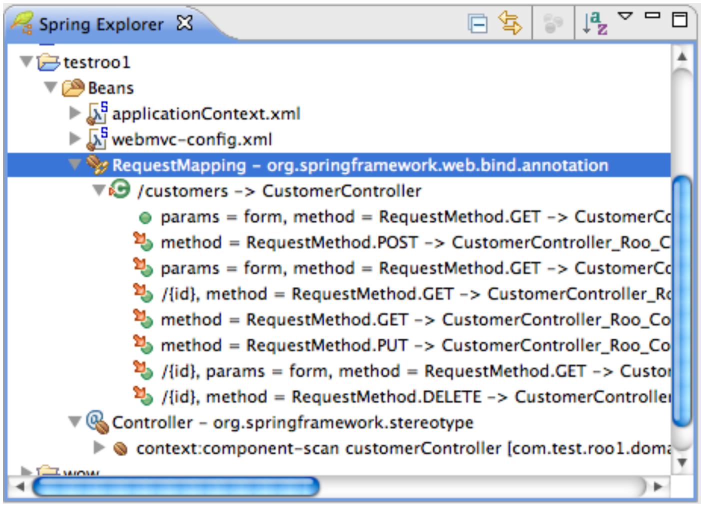
Project: org.springframework.samples.petclinic

Resource URL	Request Method	Handler Method
/owners/new	RequestMethod.POST	AddOwnerForm.processSubmit(Owner, BindingResult, SessionStatus)
/owners/new	RequestMethod.GET	AddOwnerForm.setupForm(Model)
/owners/{ownerId}/pets/new	RequestMethod.POST	AddPetForm.processSubmit(Pet, BindingResult, SessionStatus)
/owners/{ownerId}/pets/new	RequestMethod.GET	AddPetForm.setupForm(int, Model)
/owners/*/pets/{petId}/visits/new	RequestMethod.POST	AddVisitForm.processSubmit(Visit, BindingResult, SessionStatus)
/owners/*/pets/{petId}/visits/new	RequestMethod.GET	AddVisitForm.setupForm(int, Model)
/owners/{ownerId}		ClinicController.ownerHandler(int)

Custom handler for displaying an owner.
Parameters:
ownerId the ID of the owner to display
Returns:
a ModelMap with the model attributes for the view

Support for ITDs in Spring Explorer and Request-Mapping-View

A common situation in Roo projects is that aspects introduce standard controller methods via intertype declarations. STS 2.6.0 now supports this way of adding controller methods by visualizing them in the Spring Explorer and the Request-Mapping-View. You can also directly navigate from the Spring Explorer to the aspect by double-clicking on the mapping item.

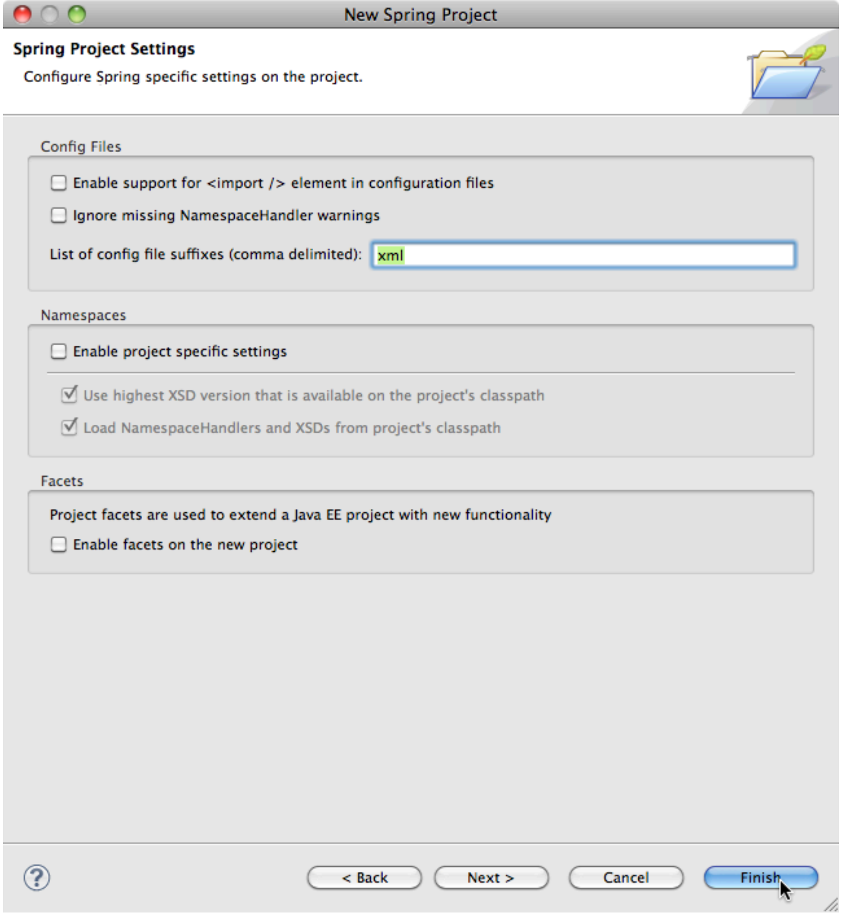


```
testroo1
  Beans
    applicationContext.xml
    webmvc-config.xml
    RequestMapping - org.springframework.web.bind.annotation
      /customers -> CustomerController
        params = form, method = RequestMethod.GET -> CustomerCo
        method = RequestMethod.POST -> CustomerController_Roo_C
        params = form, method = RequestMethod.GET -> CustomerCo
        /{id}, method = RequestMethod.GET -> CustomerController_Ro
        method = RequestMethod.GET -> CustomerController_Roo_Co
        method = RequestMethod.PUT -> CustomerController_Roo_Co
        /{id}, params = form, method = RequestMethod.GET -> Custor
        /{id}, method = RequestMethod.DELETE -> CustomerControlle
    Controller - org.springframework.stereotype
      context:component-scan customerController [com.test.roo1.doma
wow
```

Please note, that the ITD request mappings are not shown until you rebuild the project. So if you don't see them immediately, just clean the project.

Updated Spring Project Wizard

The Spring project wizard has been updated to function like the Java project wizard including all the options you are used to from the Java project wizard, and the Spring project settings page has been expanded to include more options.



Spring Project Settings

Configure Spring specific settings on the project.

Config Files

□ Enable support for <import /> element in configuration files

□ Ignore missing NamespaceHandler warnings

List of config file suffixes (comma delimited): xml

Namespaces

□ Enable project specific settings

☑ Use highest XSD version that is available on the project's classpath

☑ Load NamespaceHandlers and XSDs from project's classpath

Facets

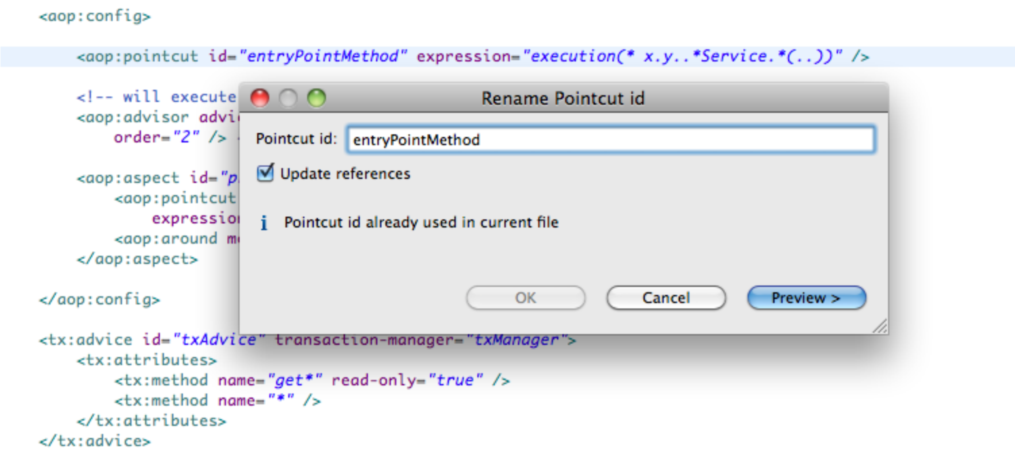
Project facets are used to extend a Java EE project with new functionality

□ Enable facets on the new project

Spring project settings can be modified any time after the project has been created by going to the project properties dialog and entering the Spring section.

Refactoring Improvements

In addition to a number of small bug fixes in this area STS 2.6.0 now supports automated re-naming of Advice-IDs and Pointcut-IDs.



```
<aop:config>

    <aop:pointcut id="entryPointMethod" expression="execution(* x.y..*Service.*(..))" />

    <!-- will execute
    <aop:advisor advi
        order="2" />

    <aop:aspect id="p
        <aop:pointcut
            expressio
        <aop:around m
    </aop:aspect>

</aop:config>

<tx:advice id="txAdvice" transaction-manager="txManager">
    <tx:attributes>
        <tx:method name="get*" read-only="true" />
        <tx:method name="*" />
    </tx:attributes>
</tx:advice>
```

Pointcut id: entryPointMethod

☑ Update references

i Pointcut id already used in current file

AspectJ and AJDT improvements

A detailed list of improvements for the new AspectJ and AJDT versions can be found here:

http://eclipse.org/ajdt/whatsnew212/

http://eclipse.org/aspectj/doc/released/README-1611.html

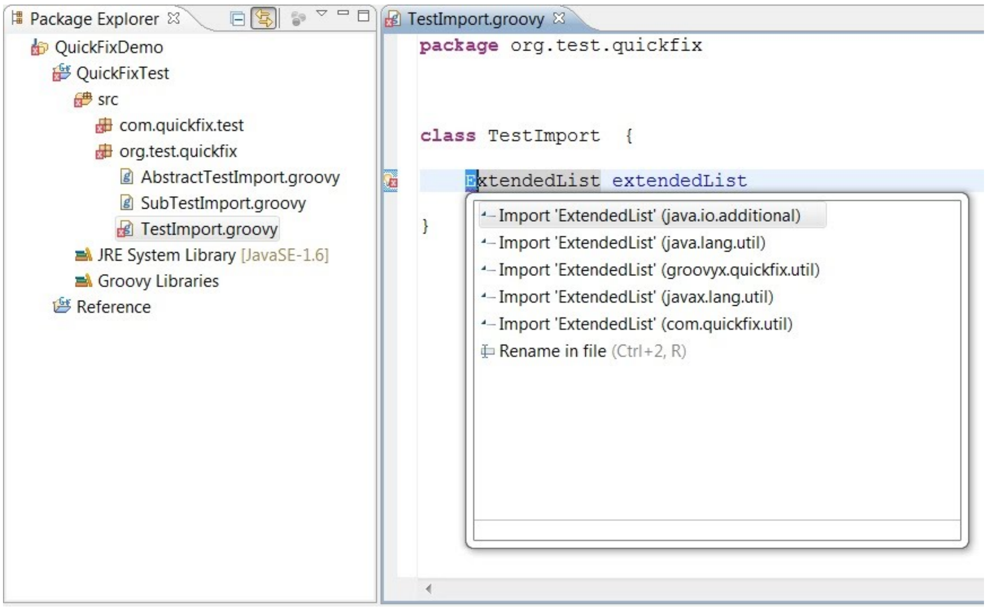
Groovy & Grails Development Tools

General Updates

- Now includes Groovy 1.7.8
- Now supports Grails 1.3.7
- Basic support for Groovy++
- Headless Groovy-Eclipse feature
- Timers for ast transform operations now shown in Groovy Event Console

Better ordering for Quick fixes and content assist

Suggested types to import in both quick fix and content assist are now ordered based on various criteria. Types imported from java.*, groovy.*, groovyx.*, and javax.* have higher priority, in that order, than types from other packages as seen in this screen shot where the type "ExtendedList" is being imported:



Package Explorer

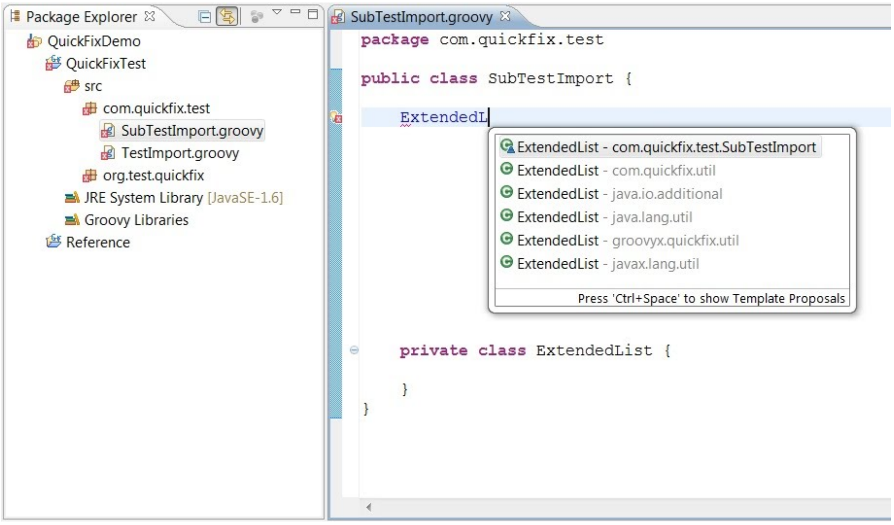
- QuickFixDemo
  - QuickFixTest
    - src
      - com.quickfix.test
      - org.test.quickfix
        - AbstractTestImport.groovy
        - SubTestImport.groovy
        - TestImport.groovy
    - JRE System Library [JavaSE-1.6]
    - Groovy Libraries
  - Reference

TestImport.groovy

```
package org.test.quickfix


class TestImport  {

    ExtendedList extendedList

}
```

- Import 'ExtendedList' (java.io.additional)
- Import 'ExtendedList' (java.lang.util)
- Import 'ExtendedList' (groovyx.quickfix.util)
- Import 'ExtendedList' (javax.lang.util)
- Import 'ExtendedList' (com.quickfix.util)
- Rename in file (Ctrl+2, R)

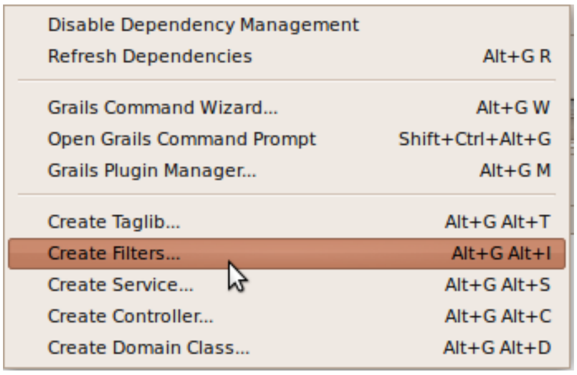
However, if importing types from similarly named packages as well as types declared in the same top level type like a private nested type, the private nested type has higher precedence, followed by public types from similarly named packages, followed by types from java.*, groovy.*, groovyx.*, javax.*, and finally types from other packages. The following screen shows the private type "ExtendedList" appearing first in the content assist suggestion, followed by a type from a similarly named package, followed by the remaining suggestions:


Package Explorer
QuickFixDemo
QuickFixTest
src
com.quickfix.test
SubTestImport.groovy
TestImport.groovy
org.test.quickfix
JRE System Library [JavaSE-1.6]
Groovy Libraries
Reference
SubTestImport.groovy
ExtendedList - com.quickfix.test.SubTestImport
ExtendedList - com.quickfix.util
ExtendedList - java.io.additional
ExtendedList - java.lang.util
ExtendedList - groovyx.quickfix.util
ExtendedList - javax.lang.util
Press 'Ctrl+Space' to show Template Proposals

```
package com.quickfix.test

public class SubTestImport {

    ExtendedL

    private class ExtendedList {

    }
}
```

Wizards

There is now a 'new Filters' menu entry alongside other Grails specific 'new wizard' menu items.

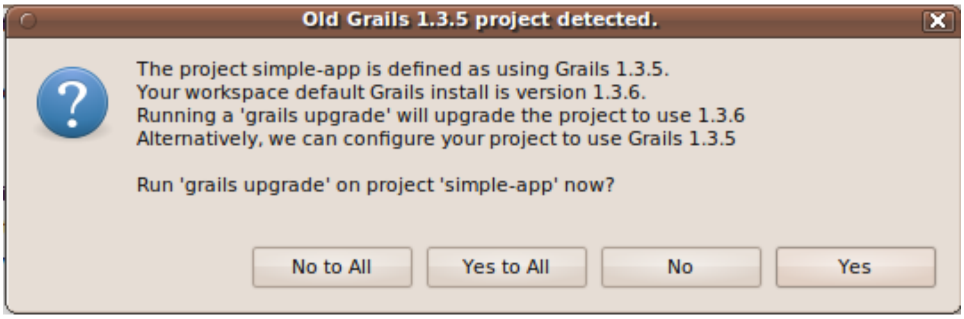

Disable Dependency Management
Refresh Dependencies Alt+G R
Grails Command Wizard... Alt+G W
Open Grails Command Prompt Shift+Ctrl+Alt+G
Grails Plugin Manager... Alt+G M
Create Taglib... Alt+G Alt+T
Create Filters... Alt+G Alt+I
Create Service... Alt+G Alt+S
Create Controller... Alt+G Alt+C
Create Domain Class... Alt+G Alt+D

Project configuration

- grails-app/util is automatically added as a source folder to the classpath of new Grails projects.
- aid in reconfiguring projects after changing the default grails install

Problem: When the user configures a new default Grails install they will also have to reconfigure any existing projects that implicitly use the default install.

Solution: When the workspace default is changed, STS will pop up a dialog offering the user the option to upgrade affected projects.



Old Grails 1.3.5 project detected.

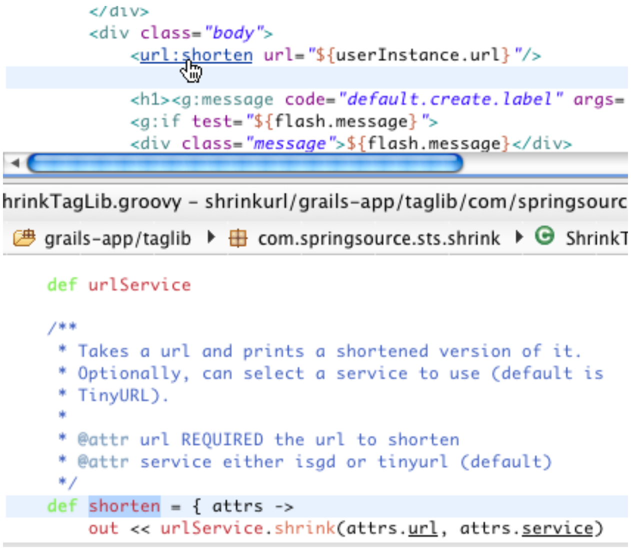
The project simple-app is defined as using Grails 1.3.5.
Your workspace default Grails install is version 1.3.6.
Running a 'grails upgrade' will upgrade the project to use 1.3.6
Alternatively, we can configure your project to use Grails 1.3.5

Run 'grails upgrade' on project 'simple-app' now?

If the user declines to upgrade the project, then the project configuration will be changed to explicitly use the non-default install.

GSP Support - Navigation to GSP definition

Pressing F3 or CTRL-Click on a GSP tag will now open the tag's definition. This will work for built-in tags as well as custom tags.



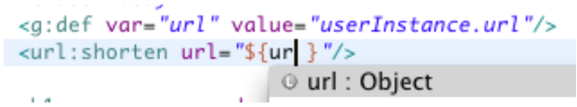
```
</div>
<div class="body">
    <url:shorten url="${userInstance.url}"/>

    <h1><g:message code="default.create.label" args=
    <g:if test="${flash.message}">
    <div class="message">${flash.message}</div>
```

hrinkTagLib.groovy – shrinkurl/grails-app/taglib/com/springsourc

```
def urlService

/**
 * Takes a url and prints a shortened version of it.
 * Optionally, can select a service to use (default is
 * TinyURL).
 *
 * @attr url REQUIRED the url to shorten
 * @attr service either isgd or tinyurl (default)
 */
def shorten = { attrs ->
    out << urlService.shrink(attrs.url, attrs.service)
```

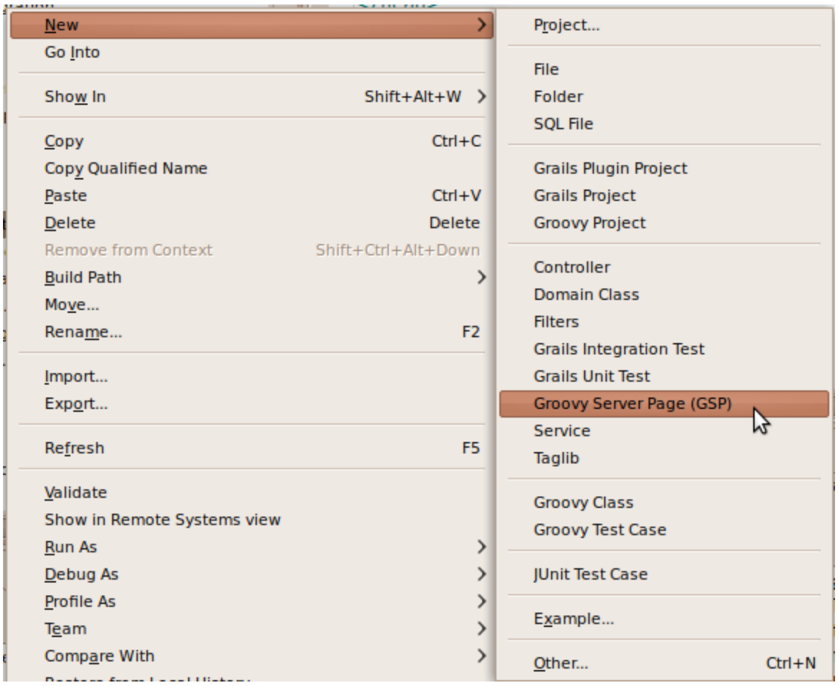
GSP Support - Content assist for g:def defined variables

After using the g:def tag to define a variable in your GSP file, this variable will now be available in scriptlets that are in the scope of the tag.



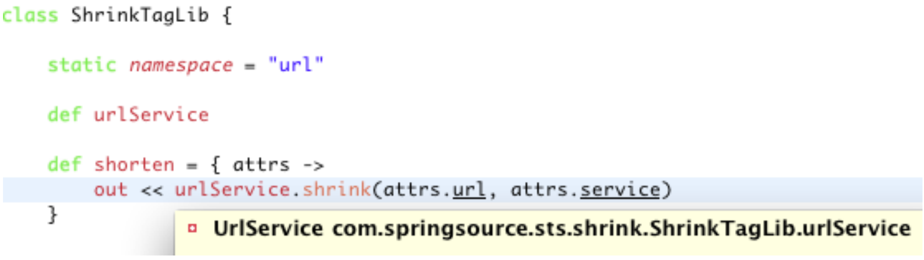
```
<g:def var="url" value="userInstance.url"/>
<url:shorten url="${ur}"/>
```

'New Groovy Server Page' wizard now accessible directly from Grails Explorer's 'new' context menu.



Groovy Editor - Code completion for autowired service fields

The types of Grails autowired services are now correctly inferred in the Groovy Editor. You can see an example of this in the screenshot below where the type of urlService is inferred to be UrlService, which is a service class in the same Grails project.

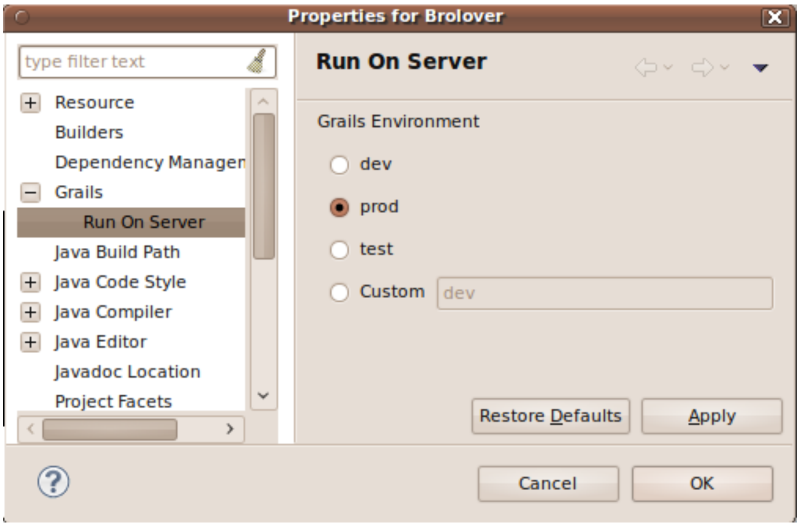


```
class ShrinkTagLib {

    static namespace = "url"

    def urlService

    def shorten = { attrs ->
        out << urlService.shrink(attrs.url, attrs.service)
    }
```

UrlService com.springsource.sts.shrink.ShrinkTagLib.urlService

Support setting 'env' parameters for deploying Grails Apps on Tc server

There is now a property page for Grails App projects allowing the user to set the 'env' parameter used to deploy to the embedded Tc Server instance. Right-click on a Grails project and go to "Properties >> Grails >> Run On Server".



Run On Server

Grails Environment

○ dev

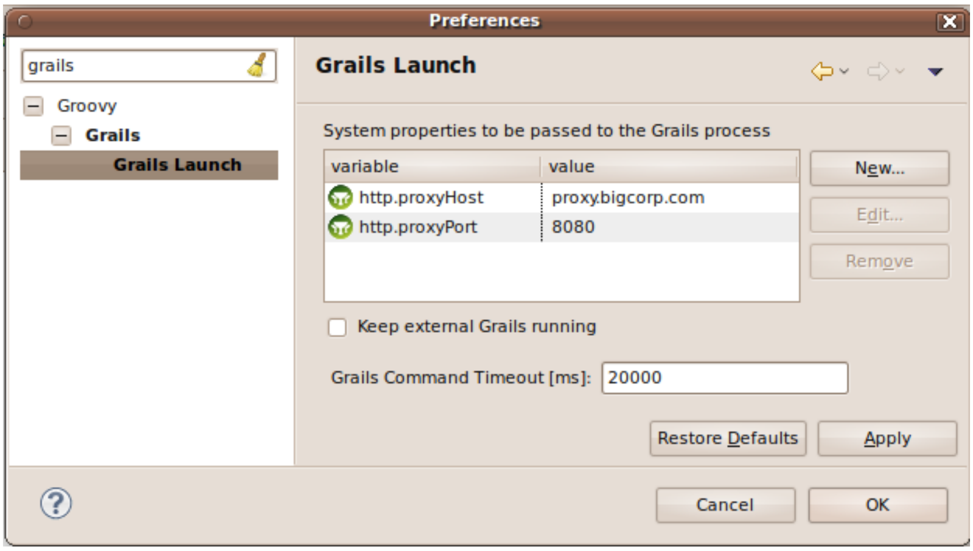
● prod

○ test

○ Custom dev

UI support for setting System properties passed to Grails Commands

A preferences page under "Grails >> Launch" has been added. It allows defining system properties that will be passed automatically to any Grails command executed from within STS.



Grails Launch

System properties to be passed to the Grails process

variable	value
http.proxyHost	proxy.bigcorp.com
http.proxyPort	8080

☐ Keep external Grails running

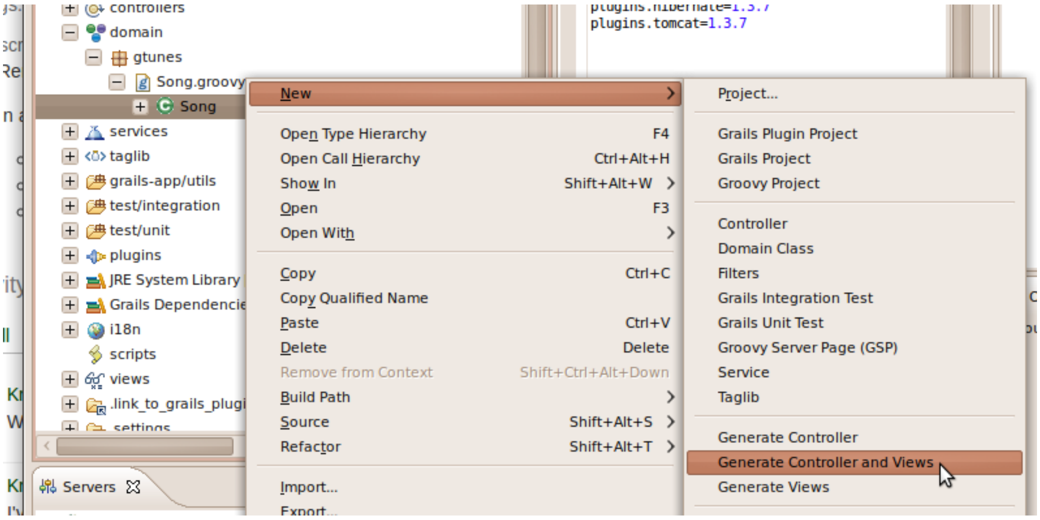
Grails Command Timeout [ms]: 20000

Upgrade Grails Projects when project settings changed

If the user changes a Grails Project's project specific settings to use different Grails install, then it is reasonable to assume they will probably also want to upgrade the project contents in accordance with the new Grails version. STS will now popup a dialog offering to run the Grails upgrade command on the user's behalf.

Additional Right-click functionality in project explorer

It is now possible to invoke the grails commands "generate-all", "generate-views" and "generate-controller" from the "new" context menu.



Context sensitive initialization of Grails Command Wizard

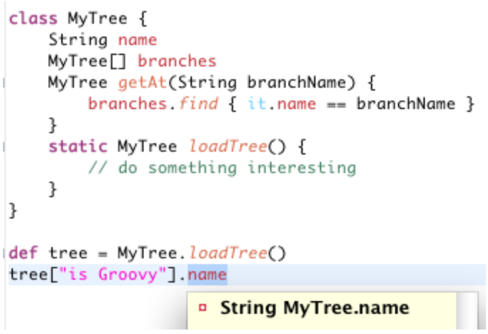
Many Grails related menu items (new controller, new domain class, etc.) open the grails command wizard to execute a specific grails command. The logic for initializing the dialog has been extended. Now STS will try to guess useful values to put into the dialog, based on the element selected in the project explorer. For example, when invoking "generate views" the command expects a domain class as a parameter. So if a controller class was selected, the fully qualified name of the corresponding domain class will be placed into the dialog automatically.

Additional improvements

- Compilation of Grails projects will no longer produce warnings in the problems view for dependent plugin source files.
- Overriding the plugin install location (grails.project.plugins.dir) in a Grails project will now correctly be reflected in any linked source folders created for plugins.

Type inferencing for getAt

Custom getAt(String) methods will now be used to feed the inferencing engine when using '[ ]'. For example:



```
class MyTree {
    String name
    MyTree[] branches
    MyTree getAt(String branchName) {
        branches.find { it.name == branchName }
    }
    static MyTree loadTree() {
        // do something interesting
    }
}

def tree = MyTree.loadTree()
tree["is Groovy"].name
```

In the preceeding code, notice how the reference to the name field is correctly resolved to coming from the MyTree class.

Better unicode support

Unicode escape sequences (e.g., \uE096) now have correct offsets and lengths in the parse tree, so you can use unicode escape sequences freely in your Groovy code and source locations in your file will not be corrupted.

Speedier type inferencing

Where appropriate, the inferencing engine will not visit the entire file to determine the type of a single expression. Rather, it will prune the branches of the AST and only visit the method or field declaration that contains the expression to be inferred. This has a noticeable affect on content assist invocation speeds.

Optional non-jar locking behaviour on Windows

On windows there is now an option to try and avoid Groovy-Eclipse locking jars and preventing them from being updated (or removed, in the case of attempting to uninstall a plugin in a grails project). Specify the system property -Dgreclipse.nonlocking=true when starting STS to activate the use of non locking classloaders. If no problems come up over the next few releases this may be made the default mode of operation. Locks should also be released (regardless of that system property setting) if the offending project is closed.

More Groovy-Eclipse improvements

A detailed list of improvements for Groovy-Eclipse can be found here:

http://groovy.codehaus.org/Groovy-Eclipse+2.1.2+New+and+Noteworthy

Some highlights are:

- Code assist performance has been improved.
- Find occurrences in groovy files (CTRL-Shift-U) now works.
- Mark occurrences now works.
- Improved syntax highlighting.
- Better type inferencing for closures inside DGM calls.

- Ability to set Method entry breakpoints.
- Fixed some issues preventing the setting of breakpoints.

Fixed Bugs and Enhancement Requests

Here is a full list of resolved bugs and enhancement requests for the 2.6.0 release (including the 2.6.0 milestone builds):

Enhancements

- STS-1610 - Ensure STS includes Grails 1.3.7
- STS-1599 - As a user, I'd like to create a GSP easier from a context menu
- STS-1546 - update STS to Eclipse 3.6.2
- STS-1531 - Roo Plugin Manager
- STS-1524 - killing grails commands due to a timeout should give the user a good idea of what happened and where to change the value
- STS-1523 - Update Grails dependencies takes longer than the timeout of 15000
- STS-1514 - Long Running Grails Process should support system properties set in GrailsCommand
- STS-1491 - [gsps] content assist for variables
- STS-1485 - Service support - recognition for code assist and inferencing
- STS-1478 - Handle type inferencing for inserted TagLib variables
- STS-1473 - Merge the New Spring Project & New Java Project wizards
- STS-1471 - Add "Scan" button to discover all Web Flow definition files in my project
- STS-1466 - Parsing of updates.xml should understand platform filter
- STS-1465 - @RequestMapping display improvements
- STS-1457 - Collect ideas for improving the @RequestMapping view
- STS-1455 - STS: Roo Shell: Tab completion does not work
- STS-1414 - Hovering over gsp tags could show the available attributes
- STS-1410 - Deploying Grails applications using a different environment than production using "Run on server..."
- STS-1408 - [GSP] Ability to navigate to declaration of GSP tag
- STS-1315 - Code completion for autowired dependencies
- STS-1272 - As a user, I want an easy way to set system properties for any grails command I execute
- STS-1121 - Syntax coloring issues
- STS-871 - [Grails] Type inferencing of autowired fields
- IDE-1166 - GRAILS Deployment should be done in DEV mode
- IDE-812 - Hyperlinking for p namespace missing

Tasks

- STS-1403 - Right click functionality for "Domain Classes"

- STS-1355 - Implement java editor tool support for @RequestMapping
- STS-1266 - [Grails] Add automated tests that involve multiple grails installs

Sub-tasks

- STS-1625 - Improve the presentation of transitions
- STS-1615 - display warning when trying to enable insight on for tc Server instances that have a space in the path
- STS-1614 - update discovery.jar for 2.6
- STS-1611 - Extend "link with editor" to support link with selection
- STS-1603 - implement a "link with editor" feature for RequestMappings view
- STS-1596 - Add content assist for @RequestMapping method parameter type
- STS-1589 - Add quick fix for @RequestMapping parameter
- STS-1588 - Add quick fix for @ResponseBody for method with @RequestMapping
- STS-1586 - Exclude standard arguments in path variable quick fix
- STS-1585 - layout items according to their order in the XML
- STS-1584 - Add quick fix for creating method parameter when URI template variable does not match
- STS-1579 - update if transitions
- STS-1578 - update transition creation tools
- STS-1571 - update iconography
- STS-1570 - icons should not have borders around them
- STS-1569 - changing state labels should refactor the XML to keep transitions in place
- STS-1568 - labels should be fixed width, and behave like the SI graph
- STS-1566 - Improve @PathVariable content assist by showing all method params
- STS-1563 - fix SpringIDE compile errors on Eclipse 3.7
- STS-1559 - Add code template for creating @ExceptionHandler method
- STS-1558 - Add code template for creating @InitBinder method
- STS-1553 - Add @ExceptionHandler quick fix
- STS-1552 - Add @ModelAttribute quick fix
- STS-1551 - Add @InitBinder quick fix
- STS-1550 - Add @RequestParam quickfix
- STS-1549 - Implement @PathVariable refactoring
- STS-1525 - Implement add @Controller quickfix
- STS-1482 - create-filter wizard should be available where appropriate
- STS-1383 - Add validation for variables in @RequestMapping
- STS-1382 - Add content assist for variables in @RequestMapping
- STS-1372 - Content assist for special GSP attributes
- STS-671 - [grails] Ability to run grails commands in-process

Defects

- STS-1662 - Damaged launch configuration on Windows - affects ability to run grails commands from the prompt

- STS-1660 - NPE in UAA
- STS-1659 - TcServer enable insight with spaces in path warning dialog is misbehaving
- STS-1654 - ArrayIndexOutOfBoundsException when opening console for application that is not running
- STS-1653 - Errors running builder 'Spring Project Builder' on project. URI scheme is not "file"
- STS-1652 - connection times out when authenticating proxy is specified
- STS-1651 - Annotation @org.codehaus.groovy.grails.plugins.metadata.GrailsPlugin is not allowed on element ANNOTATION
- STS-1649 - Visual editor selections are not updating the properties view
- STS-1644 - fix platform filter for extension install
- STS-1643 - fix icon in STS support editor
- STS-1630 - Roo shell is inconsistent with roo.bat cmd line
- STS-1629 - Update installed Roo addon with Add-on Manager
- STS-1624 - Breakpoint not being installed when running with app-engine plugin installed
- STS-1623 - The gsp editor does not recognize the TagLibs of plugins any longer
- STS-1621 - Wrong wizard validation: tc Server requires JDK
- STS-1620 - Trying to open template.mf fails with NoClassDefFoundError
- STS-1616 - UAA 'Value too long' stack trace
- STS-1609 - [SI graph] tcp-inbound-channel-adapter is rendered incorrectly
- STS-1608 - grails-app/utils should be included in the Build Path as Source Folder
- STS-1607 - STS 2.5.2 deadlocks on loading Spring XML
- STS-1604 - GrailsVersion mismatch problem may arise when user changes project preferences
- STS-1601 - Unexpected runtime error while computing a text hover
- STS-1598 - Cannot rename non Java/Groovy files
- STS-1595 - NPE when no default Roo runtime is configured
- STS-1594 - reopening STS-1510 - build taking a long time parsing spring bean configs
- STS-1587 - move dm Server feature to Extension install
- STS-1583 - Bogus validation on int:poller attribute if not the first child element in a config file
- STS-1582 - Hover text of XML attribute of Batch schema uses wrong schema version?
- STS-1577 - NPE when deploying VMforce applications
- STS-1573 - Roo wizard should not enable Finish unless valid Roo runtime is selected
- STS-1567 - Cannot complete the install because one or more required items could not be found. SpringSource Tool Suite Runtime Error Analysis Support 2.5.2.201101081000-RELEASE
- STS-1565 - The overview page of the Spring config editor should display details for new/unknown namespaces.
- STS-1564 - Adding a new Spring namespace using the UI corrupts schemaLocation
- STS-1561 - As a user, I would like to install STS on Eclipse 3.7 (Indigo)
- STS-1554 - Problem with Spring IDE User Agent Analysis
- STS-1545 - error dialog displayed when hovering over link in Console and internal browser is not available

- STS-1544 - investigate deployment errors when connections are idle for a while
- STS-1543 - Updates to jars containing new versions of Spring XSDs don't seem to pickup
- STS-1542 - Spring -> Update Maven Dependencies blows away m2Eclipse projects
- STS-1539 - Interactions between STS Groovy compiler and grails war command
- STS-1538 - Need to find where this comes from and stop it coming out: [Fatal Error] :1:1: Premature end of file.
- STS-1530 - Grails refresh dependencies (Error executing script Compile: org.apache.xerces.dom.DocumentImpl.getXmlStandalone()Z)
- STS-1528 - STS 2.6.0.M1 fails to start on Mac OS X Leopard
- STS-1527 - After upgrading Grails, Grails Dependencies classpath container still reference jars from the old Grails distributions
- STS-1521 - Checkout an existing Grails (non eclipse) project from SVN just creates a new Grails project
- STS-1520 - Grails plugin manager fails to uninstall plugin
- STS-1519 - STS 2.5.2 is *really* slow at building
- STS-1518 - Problem deploying Grails App that uses the audit-log plugin on Tc Server
- STS-1515 - STS adds import statements for auto-imported packages' types also
- STS-1512 - Attributes displayed with content assist for <g:remoteLink> tag don't seem correct
- STS-1511 - IDE crash with STS 2.5.2
- STS-1508 - NPE on startup when an editor factory fails to load
- STS-1506 - installing a plugin that pulls in another plugin looks to have issues
- STS-1505 - Spring Roo Project creation wizard hangs infinitely
- STS-1502 - Problems with a refresh dependencies not quite getting the right answer (on first run, works on second)
- STS-1501 - NPE with Beans from an external JAR
- STS-1489 - [gsps] highlighting breaks down when single char reference is used
- STS-1486 - UAA throws spurious exceptions from "library usage monitor"
- STS-1480 - "The filename or extension is too long" Error When DataNucleus Enhancer is Run Within STS
- STS-1479 - "Building workspace" takes extremely long for simple compile
- STS-1468 - Constantly asked to save changes in a run config when there aren't any
- STS-1464 - Can't move refactor in Grails Explorer
- STS-1463 - Grails artifacts show up twice in Grails Explorer if refactored into default package
- STS-1462 - Code completion on groovy is very sluggish.
- STS-1461 - GC overhead limit exceeded
- STS-1459 - Spurious error in GSP file with scriplets nested in comments
- STS-1458 - Web flow graph UI & labelling updates
- STS-1456 - STS hangs
- STS-1452 - Exception thrown from usage monitor
- STS-1451 - Grails Projects imported as 'linked' projects can't be deployed by "Run On Server".

- STS-1449 - GrailsVersion class for parsing Grails version String doesn't handle build snapshot versions
- STS-1444 - Issues in Importing Grails Projects into STS from SVN
- STS-1432 - Grails version mismatch problems arise from changing default Grails install
- STS-1431 - Groovy file text colors change while editing
- STS-1411 - Exception logged to error log when gsp tidyup running for deleted project
- STS-1400 - "Create Service..." - Wizard Keyboard-ShortCut (Alt+G Alt+S) does not work
- STS-1280 - Grails project displays warnings from plugin's sources
- STS-1114 - Syntax higlighting of grails controller stops halfway the file
- STS-796 - [grails] Grails run command and dependency calculation requires JDK
- IDE-1173 - Constant reloading of Spring XML files after full build
- IDE-1171 - Deadlock in bean model access
- IDE-1170 - Workbench has not been created yet - Error occured starting Spring UAA usage monitor
- IDE-1169 - Rename ID doesn't find references in config-files that import the defining one
- IDE-1168 - Classloader "previously initiated loading for a different type" error when using custom NamespaceResolvers
- IDE-1167 - STS does not recognize Groovy syntax in GSPs
- IDE-1166 - GRAILS Deployment should be done in DEV mode
- IDE-1164 - @RequestMapping annotations don't appear in Spring Explorer or Request-Mapping view for ITD methods
- IDE-1161 - Duplicates of config files when maven projects have "Project Reference"
- IDE-1159 - Font Size in STS Navigation Panels - Mac OSX
- IDE-998 - Unable to rename advice declared with tx namepspace
- IDE-812 - Hyperlinking for p namespace missing
- IDE-608 - refactor support broken if you move/collapse a whole packages
- IDE-607 - Refactor support for change method signature