

SpringSource Tool Suite 2.6.0.M2

- New and Noteworthy -

Martin Lippert

2.6.0.M2

February 28, 2011

Updated for 2.6.0.M2

ENHANCEMENTS – 2.6.0.M2

General Updates

Spring Roo 1.1.2

The newest milestone build of STS 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

The latest STS milestone builds supports Grails 1.3.7 and ships that Grails version from the 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.

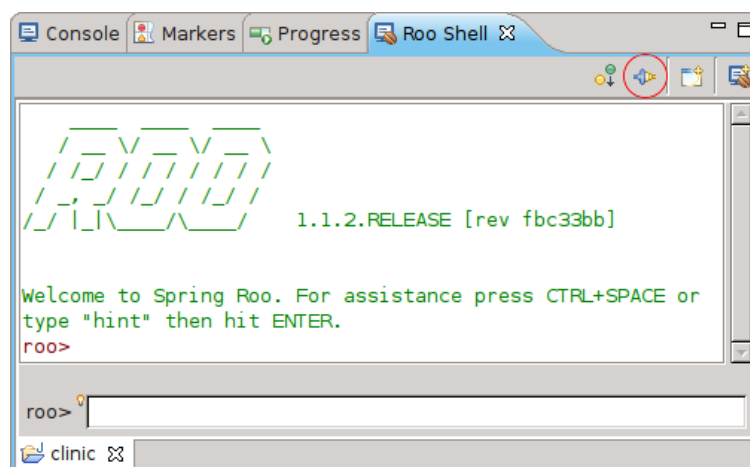
dm Server Tooling

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.

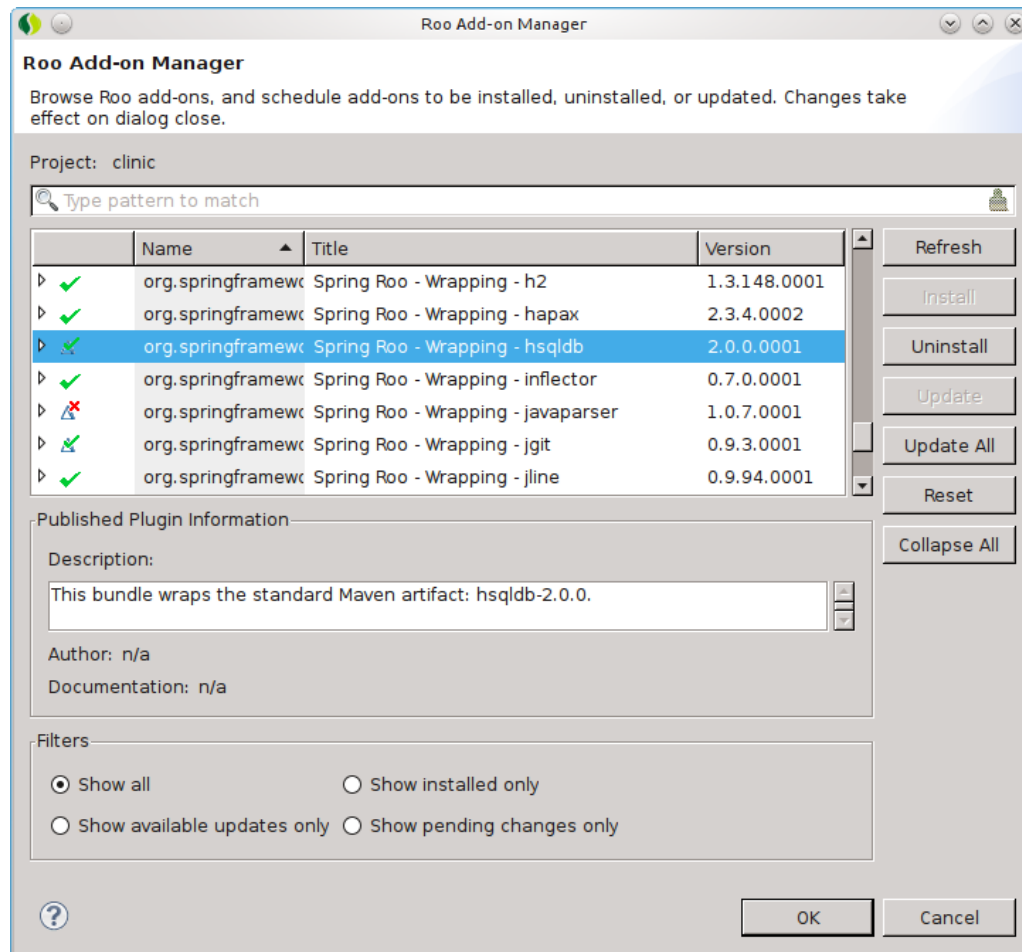
Spring Development Tools

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.



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.



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.

RequestMappings view improvements

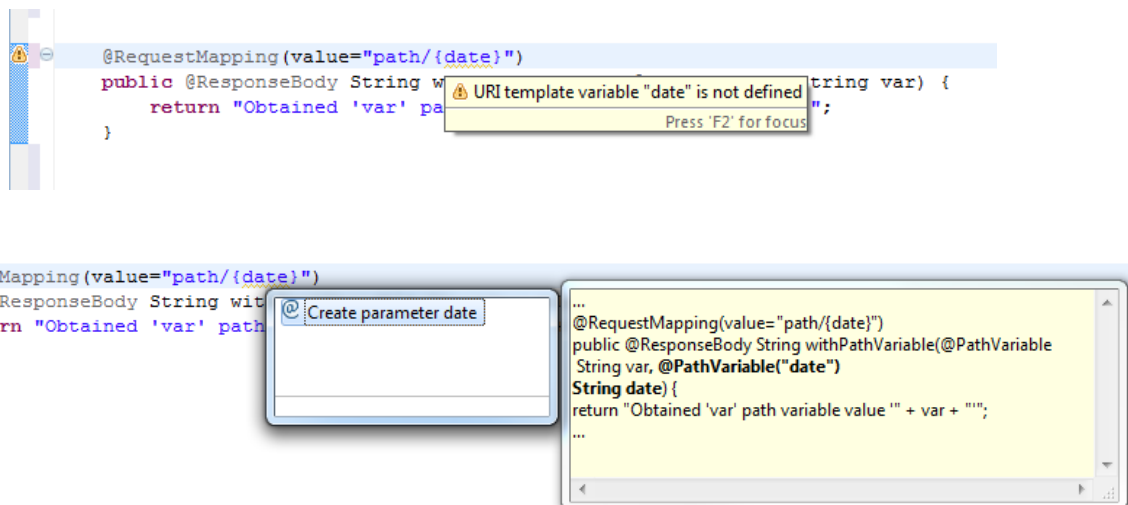
All columns in the RequestMappings view are now sortable, so users can sort the view according to URL path, Request/Method type, or Class#method signature. Additionally, the RequestMappings view now supports Link with Editor mode. When the Link with Editor button is toggled, the view will automatically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

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 ModelAndView with the model attributes for the view

Content-Assist and Quick-Fixes for @RequestMapping

We introduced content assist for URI template variable inside a @RequestMapping annotation in the last release. In this release we added as you type validation which shows warnings for variable references that are not defined as well as corresponding quick fixes:



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;
}
```

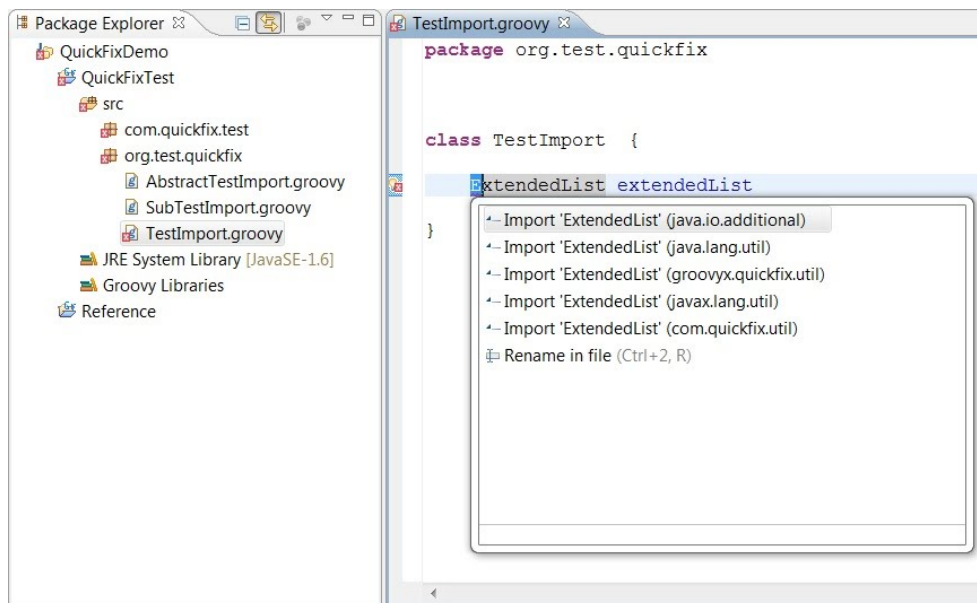
Groovy & Grails Development Tools

General Updates

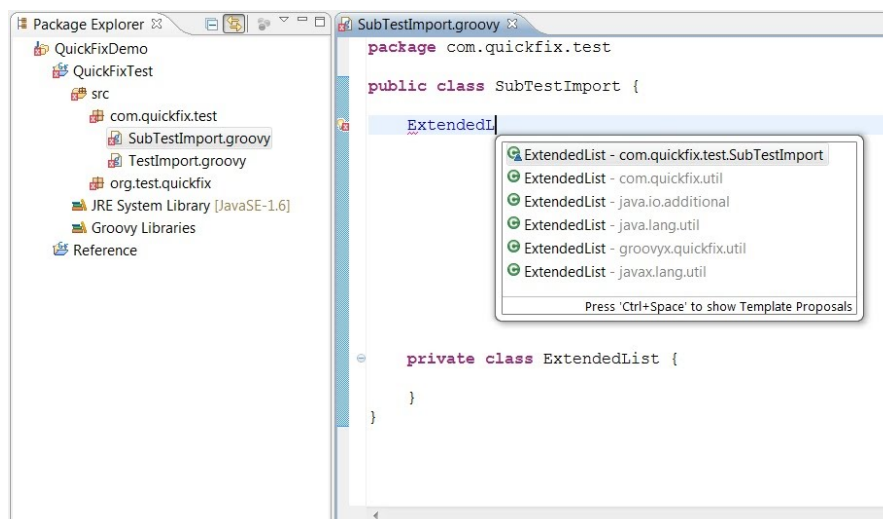
- Now includes Groovy 1.7.8
- Now supports Grails 1.3.7
- Basic support for Groovy++
- Headless Groovy-Eclipse feature
- Timers for ast transform operations now shown in Groovy Event Console

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 "Extended-List" is being imported:

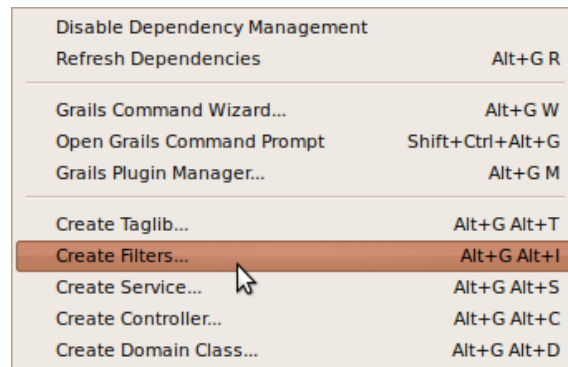


However, if importing types from similarly named packages as well as types declared in the same top level type like a private nested type, the private nested type has higher precedence, followed by public types from similarly named packages, followed by types from `java.*`, `groovy.*`, `groovyx.*`, `javax.*`, 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:



Wizards

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

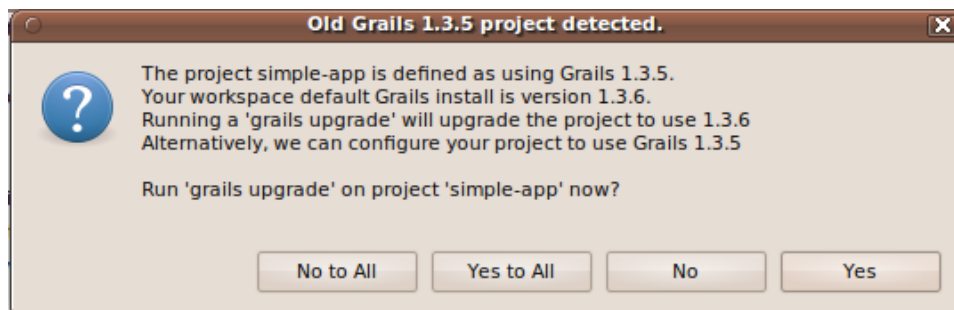


Project configuration

- 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.



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 builtin 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>
</div>

```

shrinkTagLib.groovy - shrinkurl/grails-app/taglib/com/springsource

grails-app/taglib ▶ com.springsource.sts.shrink ▶ ShrinkT

```

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.

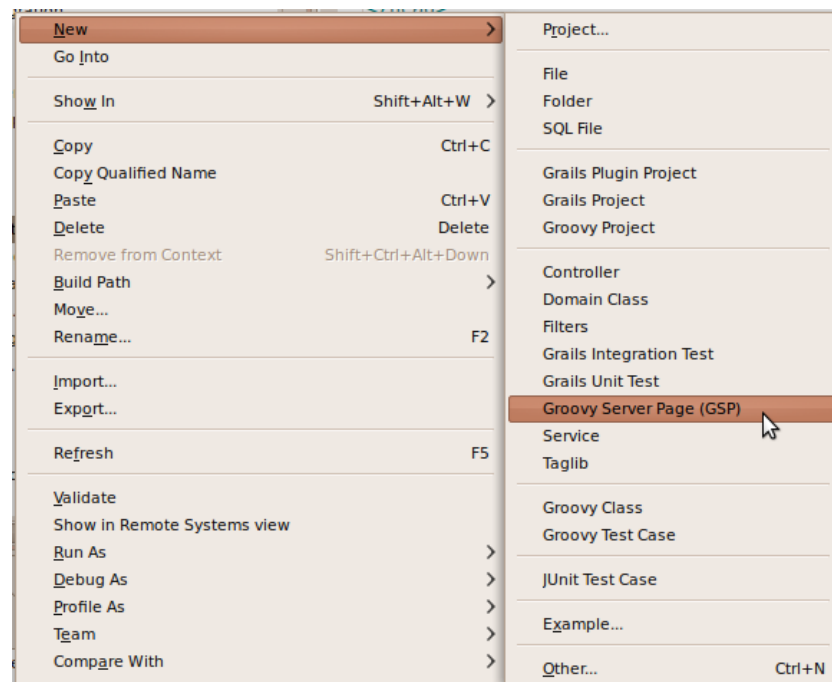
```

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

```

url : Object

'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

Fixed Bugs and Enhancement Requests

Here is a full list of resolved bugs and enhancement requests for the 2.6.0.M2 release:

Sub-tasks

- STS-1372 - Content assist for special GSP attributes
- STS-1382 - Add content assist for variables in `@RequestMapping`
- STS-1383 - Add validation for variables in `@RequestMapping`
- STS-1482 - create-filter wizard should be available where appropriate

- STS-1525 - Implement add @Controller quickfix
- STS-1550 - Add @RequestParam quickfix
- STS-1551 - Add @InitBinder quick fix
- STS-1553 - Add @ExceptionHandler quick fix
- STS-1558 - Add code template for creating @InitBinder method
- STS-1559 - Add code template for creating @ExceptionHandler method
- STS-1566 - Improve @PathVariable content assist by showing all method params
- STS-1568 - labels should be fixed width, and behave like the SI graph
- STS-1569 - changing state labels should refactor the XML to keep transitions in place
- STS-1570 - icons should not have borders around them
- STS-1571 - update iconography
- STS-1578 - update transition creation tools
- STS-1579 - update if transitions
- STS-1584 - Add quick fix for creating method parameter when URI template variable does not match
- STS-1585 - layout items according to their order in the XML
- STS-1586 - Exclude standard arguments in path variable quick fix
- STS-1588 - Add quick fix for @ResponseBody for method with @RequestMapping
- STS-1589 - Add quick fix for @RequestMapping parameter
- STS-1603 - implement a "link with editor" feature for RequestMappings view
- STS-1615 - display warning when trying to enable insight on for tc Server instances that have a space in the path

Enhancements

- STS-871 - [Grails] Type inferencing of autowired fields
- STS-1315 - Code completion for autowired dependencies
- STS-1408 - [GSP] Ability to navigate to declaration of GSP tag
- STS-1485 - Service support - recognition for code assist and inferencing
- STS-1491 - [gsps] content assist for variables
- 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-1599 - As a user, I'd like to create a GSP easier from a context menu

Defects

- STS-1400 - "Create Service..." - Wizard Keyboard-ShortCut (Alt+G Alt+S) does not work
- STS-1411 - Exception logged to error log when gsp tidyup running for deleted project
- STS-1432 - Grails version mismatch problems arise from changing default Grails install
- STS-1464 - Can't move refactor in Grails Explorer
- STS-1489 - [gsps] highlighting breaks down when single char reference is used
- STS-1518 - Problem deploying Grails App that uses the audit-log plugin on Tc Server
- STS-1520 - Grails plugin manager fails to uninstall plugin
- STS-1521 - Checkout an existing Grails (non eclipse) project from SVN just creates a new

Grails project

- STS-1527 - After upgrading Grails, Grails Dependencies classpath container still reference jars from the old Grails distributions
- STS-1530 - Grails refresh dependencies (Error executing script Compile: org.apache.xerces.dom.DocumentImpl.getXmlStandalone()Z)
- STS-1539 - Interactions between STS Groovy compiler and grails war command
- STS-1544 - investigate deployment errors when connections are idle for a while
- STS-1545 - error dialog displayed when hovering over link in Console and internal browser is not available
- STS-1554 - Problem with Spring IDE User Agent Analysis
- STS-1564 - Adding a new Spring namespace using the UI corrupts schemaLocation
- STS-1565 - The overview page of the Spring config editor should display details for new/unknown namespaces.
- 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-1577 - NPE when deploying VMforce applications
- STS-1582 - Hover text of XML attribute of Batch schema uses wrong schema version?
- STS-1583 - Bogus validation on int:poller attribute if not the first child element in a config file
- STS-1587 - move dm Server feature to Extension install
- STS-1595 - NPE when no default Roo runtime is configured
- STS-1598 - Cannot rename non Java/Groovy files
- STS-1601 - Unexpected runtime error while computing a text hover
- STS-1608 - grails-app/utis should be included in the Build Path as Source Folder
- STS-1609 - [SI graph] tcp-inbound-channel-adapter is rendered incorrectly
- STS-1616 - UAA 'Value too long' stack trace
- IDE-607 - Refactor support for change method signature
- IDE-1168 - Classloader "previously initiated loading for a different type" error when using custom NamespaceResolvers
- IDE-1173 - Constant reloading of Spring XML files after full build

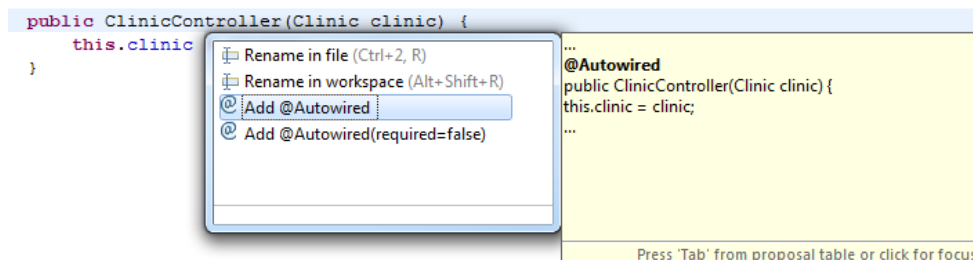
ENHANCEMENTS – 2.6.0.M1

Spring Development Tools

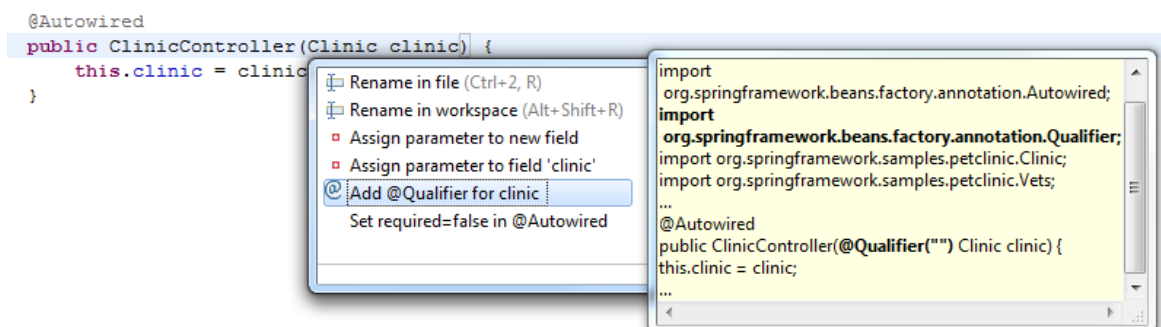
Content-Assist and Quick-Fixes for @Autowired and @Qualifier

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 provide 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.

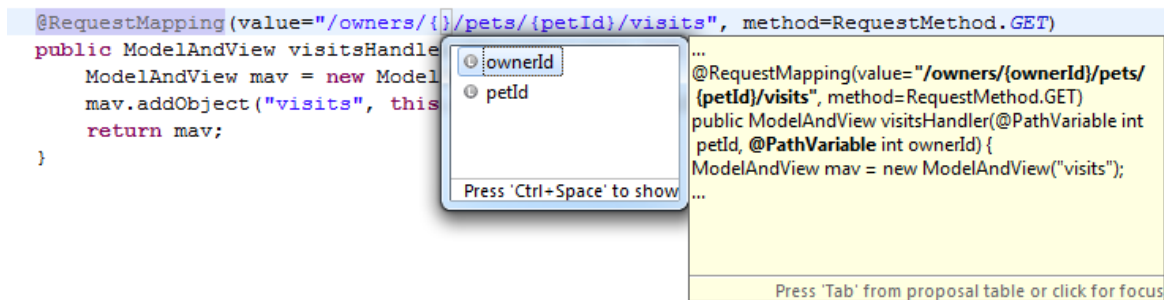


Autowiring by type may lead to multiple candidates. There is now quick fix support for adding @Qualifier annotation, as well as content assist for matching bean names.



Content-Assist and Quick-Fixes for @RequestMapping

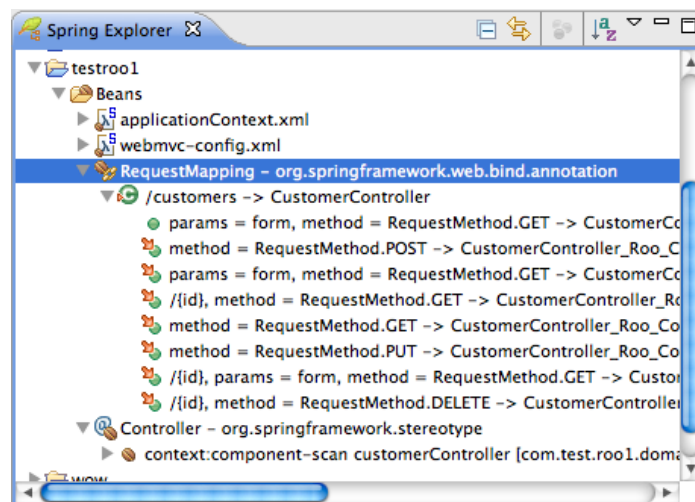
Another new feature is content assist for specifying URI template variable. When editing a @RequestMapping annotation on a method declaration, content assist is available for matching method parameters, as well as adding @PathVariable to the matching parameter to indicate that the parameter should be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



These are just the first steps towards supporting annotation-based Spring programming. Expect more to come in future milestones and releases.

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

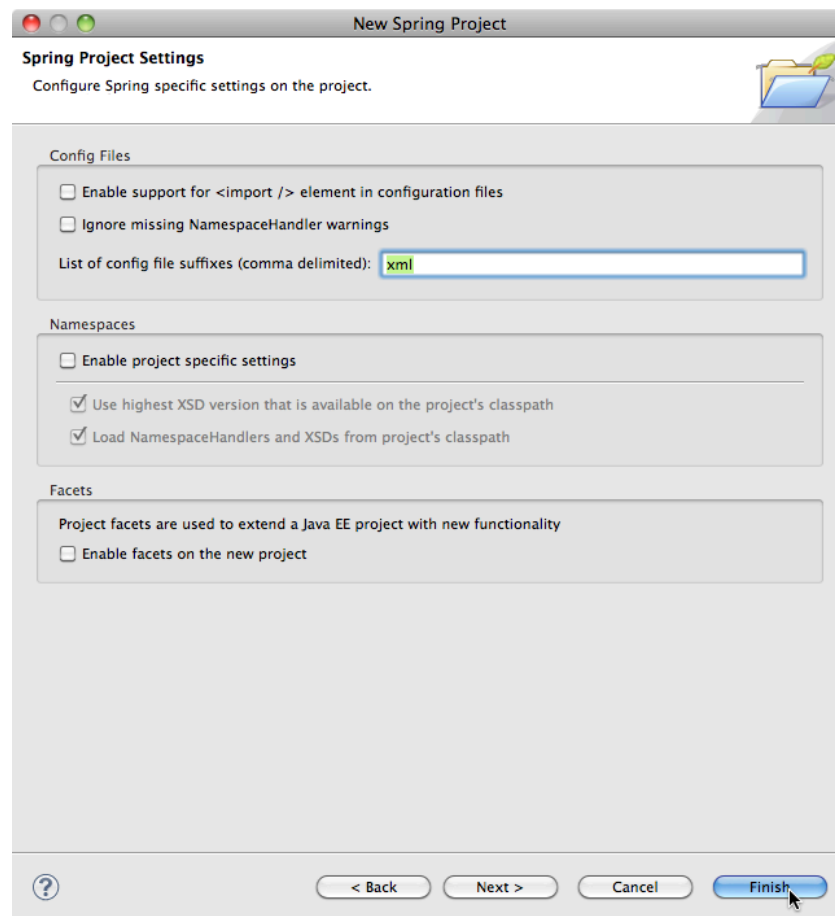
A common situation in Roo projects is that aspects introduce standard controller methods via intertype declarations. STS 2.6.0.M1 now supports this way of adding controller methods by visualizing them in the Spring Explorer and the Request-Mapping-View. You can also directly navigate from the Spring Explorer to the aspect by double-clicking on the mapping item.



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

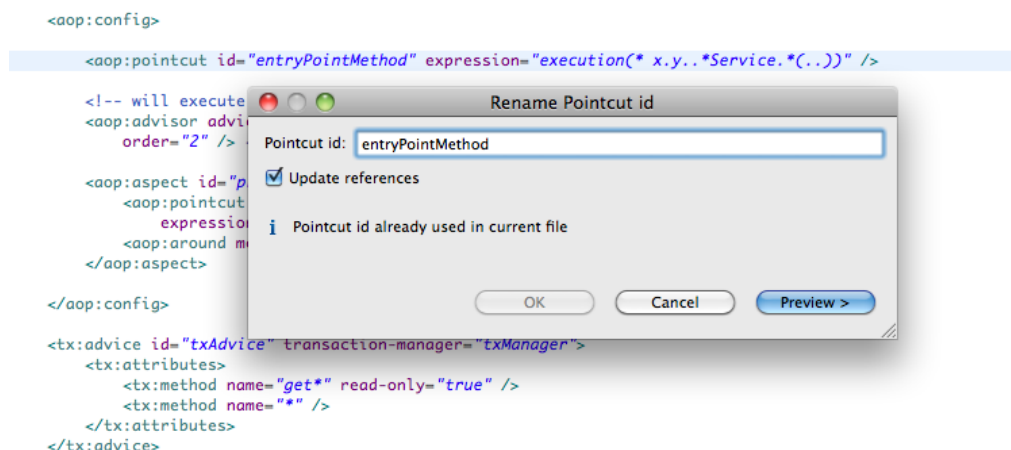
The Spring project wizard has been updated to function like the Java project wizard including all the options you are used to from the Java project wizard, and the Spring project settings page has been expanded to include more options.



Spring project settings 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

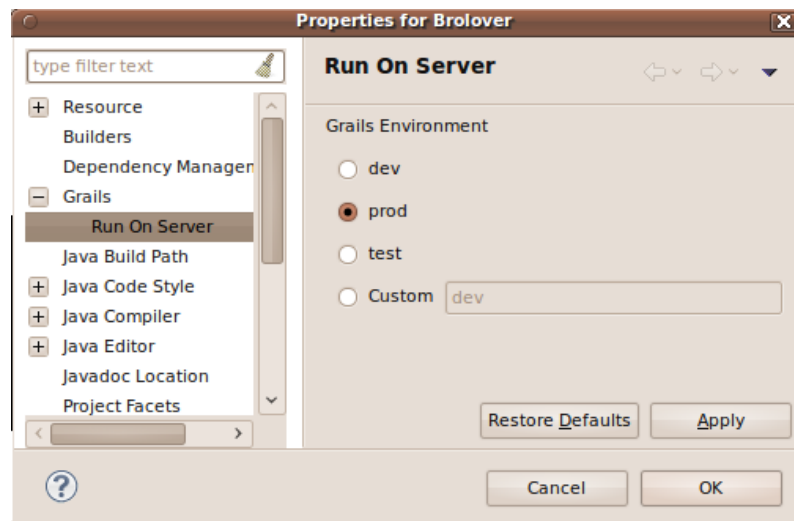
In addition to a number of small bug fixes in this area STS 2.6.0.M1 now supports automated renaming of Advice-IDs and Pointcut-IDs.



Groovy & Grails Development Tools

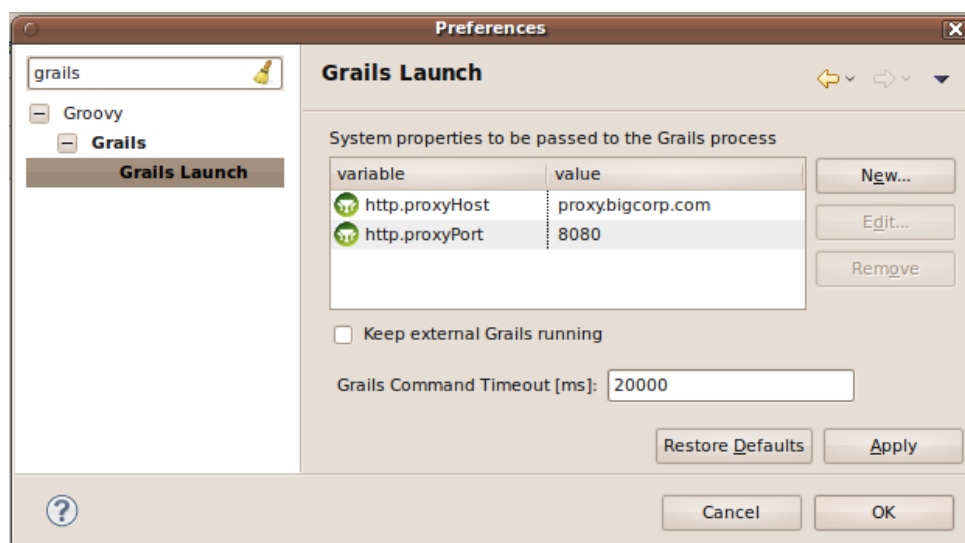
Support setting 'env' parameters for deploying Grails Apps on Tc server

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".



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.



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.

Groovy-Eclipse

- 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.M1 release:

Sub-tasks

- [STS-487] - [grails] Ability to run grails commands in-process

Enhancements

- [STS-1514] - Long Running Grails Process should support system properties set in GrailsCommand
- [STS-1478] - Handle type inferencing for inserted TagLib variables
- [STS-1473] - Merge the New Spring Project & New Java Project wizards
- [STS-1466] - Parsing of updates.xml should understand platform filter
- [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-1272] - As a user, I want an easy way to set system properties for any grails command I execute
- [STS-1121] - Syntax coloring issues
- [IDE-1166] - GRAILS Deployment should be done in DEV mode

Defects

- [STS-1519] - STS 2.5.2 is *really* slow at building
- [STS-1515] - STS adds import statements for auto-imported packages' types also
- [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-1486] - UAA throws spurious exceptions from "library usage monitor"
- [STS-1468] - Constantly asked to save changes in a run config when there aren't any
- [STS-1463] - Grails artifacts show up twice in Grails Explorer if refactored into default package
- [STS-1459] - Spurious error in GSP file with scriptlets nested in comments
- [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-1431] - Groovy file text colors change while editing
- [STS-1280] - Grails project displays warnings from plugin's sources
- [STS-1114] - Syntax highlighting of grails controller stops halfway the file
- [STS-796] - [grails] Grails run command and dependency calculation requires JDK
- [IDE-1170] - Workbench has not been created yet - Error occurred starting Spring UAA usage monitor
- [IDE-1169] - Rename ID doesn't find references in config-files that import the defining one
- [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 namespace
- [IDE-608] - refactor support broken if you move/collapse a whole packages