SpringSource Tool Suites 3.0.0

- New and Noteworthy -

Martin Lippert 3.0.0 August 13, 2012 Updated for 3.0.0.RELEASE

www.springsource.com

springsource A division of vmware

ENHANCEMENTS – 3.0.0

General Updates

Spring Tool Suite & Groovy/Grails Tool Suite

Starting with 3.0.0, we are building two separate distributions of our tool suite for you: The **Spring Tool Suite** provides all the necessary bits and pieces for working with Spring projects, including the Spring framework IDE support, tc Server integration, and support for Spring Roo. The **Groovy/Grails Tool Suite** comes with all you need for working with Groovy and Grails projects, including Groovy-Eclipse and the Grails tooling pre-installed, but without the Spring framework tooling.

Moving ahead with these two distributions, you can easily install the features that are not part of the distribution into your installation from the Dashboard.

A big step: The tool suites go completely open-source

In the past the Spring IDE subproject was the only part of the SpringSource Tool Suite that was available as open-source under the Eclipse Public License. Starting with version 3.0, we have re-organized the subprojects from that we build or distributions and made all the different parts available as open-source projects under the Eclipse Public License at GitHub.

Here is an overview of the new open-source projects:

- **Spring IDE:** This brings you all the tooling for working with the Spring framework, alongside with integrations for various additional Spring-related technologies like AJDT, Spring Integration, Spring Webflow, Spring Data, Spring Security, and Spring Roo. The support for Maven and Spring Roo, that was formerly part of STS only, was contributed to this project. (https://github.com/SpringSource/spring-ide)
- **Grails IDE:** Brings you the full Grails developer tooling that was previously shipped as part of the SpringSource Tool Suite. It is build on top of the Groovy-Eclipse project. (https://github.com/SpringSource/grails-ide)
- **Eclipse Integration for tc Server:** This project includes the Eclipse tc Server integration for the different tool suites that we ship. It allows you to create new instances, use existing ones, deploy and update apps directly from your workspace, configure your tc Server instance, and enable and use Spring Insight. (https://github.com/SpringSource/eclipse-integration-tcserver)
- **Eclipse Integration for Gradle:** This integrates Gradle into Eclipse and allows you to use your Gradle build and its configuration directly in Eclipse. (https://github.com/SpringSource/eclipse-integration-gradle)
- **Eclipse Integration Commons:** This project contains the common componentry consumed by the other projects. It includes components such as the UAA support and the SpringSource Dashboard. (https://github.com/SpringSource/eclipse-integration-commons)

As an effect of this reorganization and the open-sourcing, there are fewer dependencies between those projects. Therefore you can consume them individually from the projects update sites, if you want to. For example the Eclipse integration for VMware vFabric tc Server can be installed into a plain Eclipse JEE without the need to also install Spring IDE, Grails IDE, or other components.

You can always use the Dashboard (that comes with every project, the same with UAA) to easily add other projects to your existing installation – as you might be used to from previous SpringSource Tool Suite versions.

Distribution based on Eclipse Juno (4.2.0)

STS now ships on top of the Eclipse Juno release build, including the latest m2e 1.1 that is part of Eclipse Juno.

You will also notice changes in the overall look and feel of the IDE. This is also part of the upgrade to the Eclipse 4.2 platform and customizable, if you prefer a different style.

Upgrading existing installations

Unlike with previous versions of the SpringSource Tool suite, we don't provide an automatic upgrade path via the Eclipse installation mechanism for the 3.0.0 release. So you will not see any "Updates Available" messages coming up when you use STS 2.9.x or previous versions.

Instead we recommend to start with a fresh Spring Tool Suite or Groovy/Grails Tool Suite download. You can also use a fresh Eclipse Juno or Indigo installation and add the tooling that we provide via the Eclipse Marketplace.

The automatic upgrade path will be enabled again for upgrades to an existing STS/GGTS 3.0.0 installation.

Eclipse compatibility notes

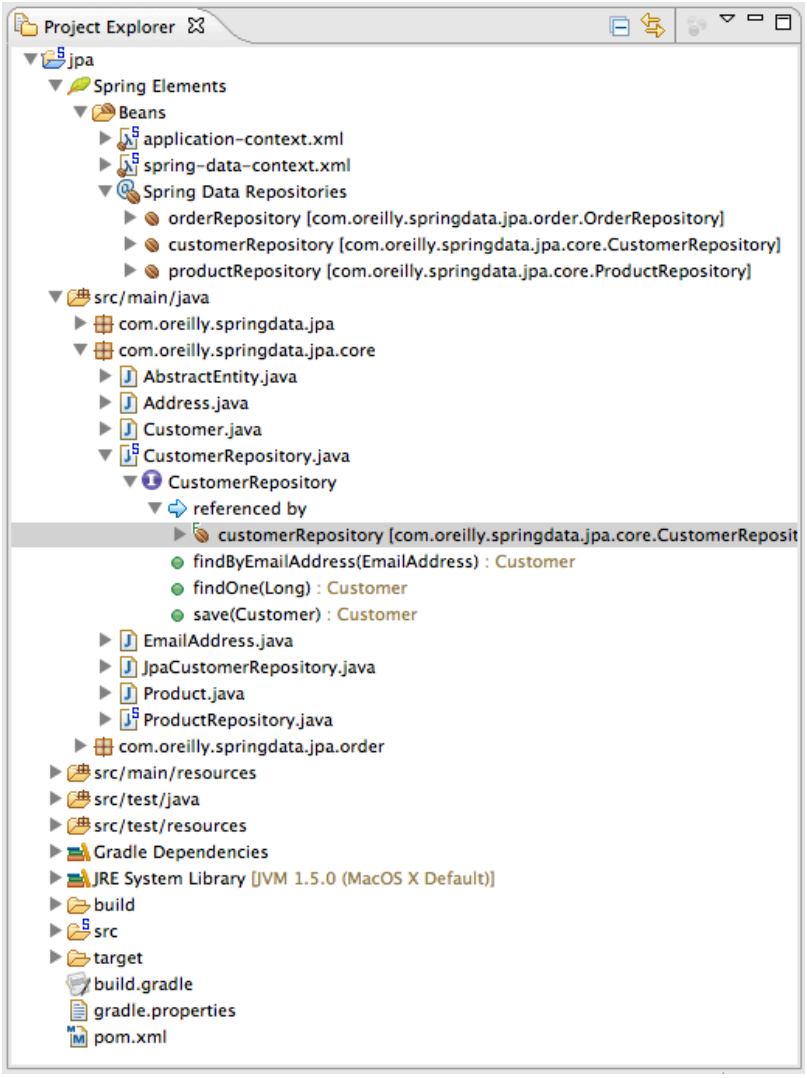
In addition to the distribution on top of Eclipse Juno (4.2), we also provide an Eclipse Indigo (3.7) compatible update site for STS, so that Indigo users can continue to use their favorite environment together with STS.

From STS 3.0.0 on we will stop supporting Eclipse Helios (3.6). There will be no STS 3.0.0 update repository available for Eclipse Helios.

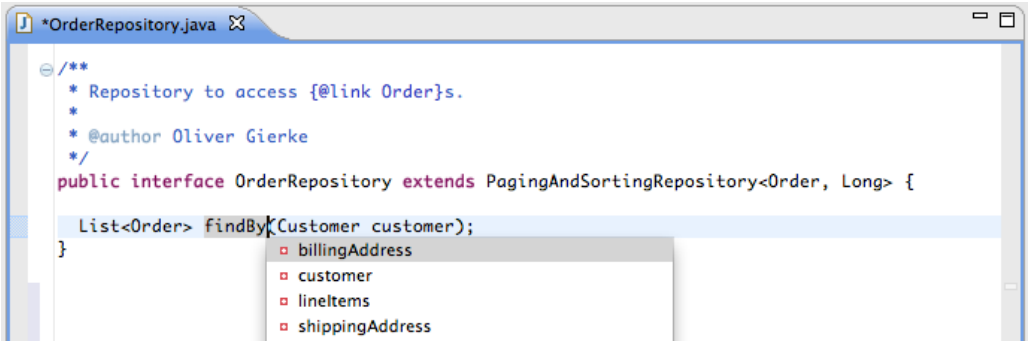
Spring Development Tools

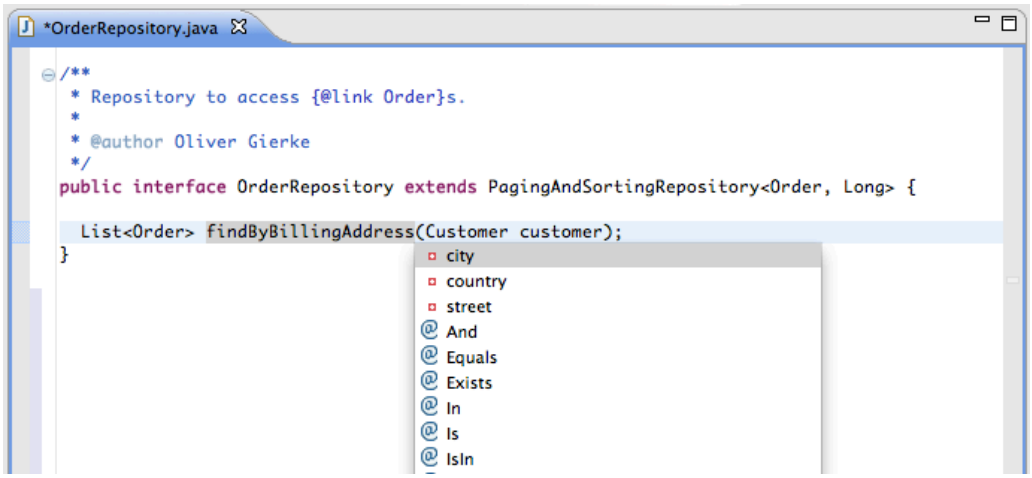
Support for Spring Data

The Spring tooling inside STS now includes the features for supporting Spring Data. Repository definitions are now shown as part of the Spring Elements in the Project Explorer, including references between code and bean definitions:

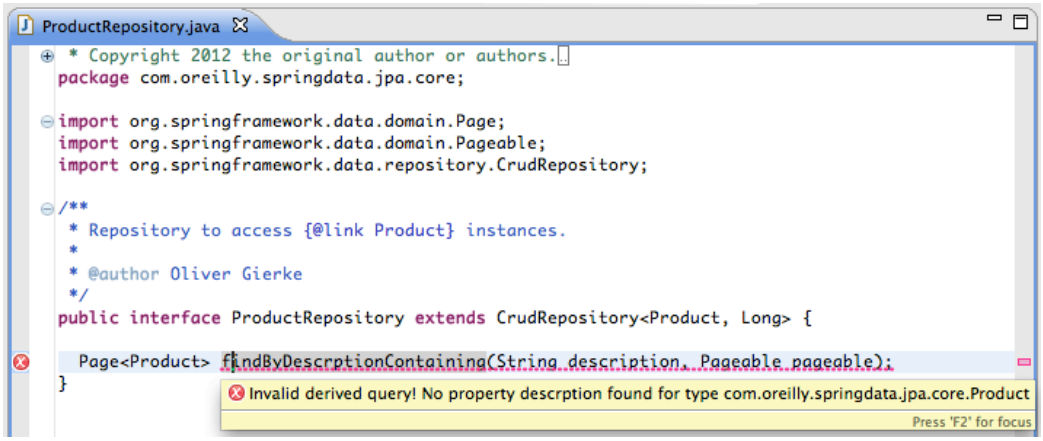


In addition to that there is code completion available for properties and keywords when writing repository methods:





The tooling also provides validation for derived query method names:



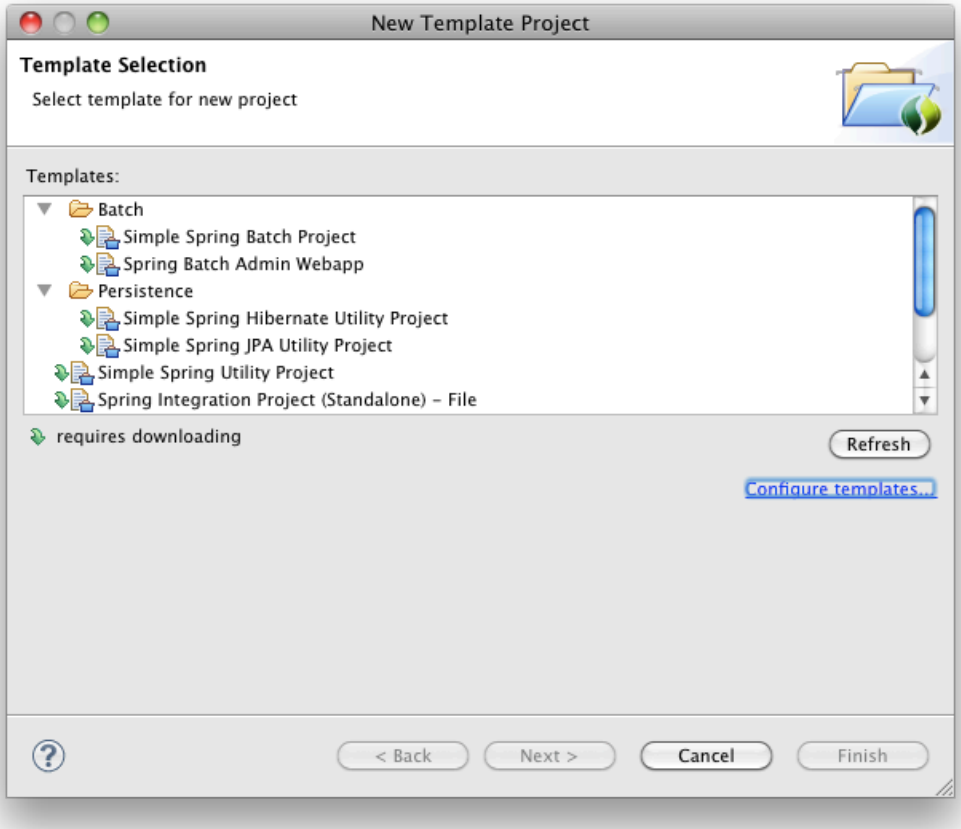
Easy Install for Spring Example Projects from GitHub

We have added a few detailed, fully featured example projects to the Dashboard. Those are the general example projects that are hosted by SpringSource on GitHub, but the Dashboard access gives you an easy way to install them directly into your IDE, just by clicking on the link on the Dashboard. This provides an easy way to run the example projects yourself, learn from them, and/or modify them.

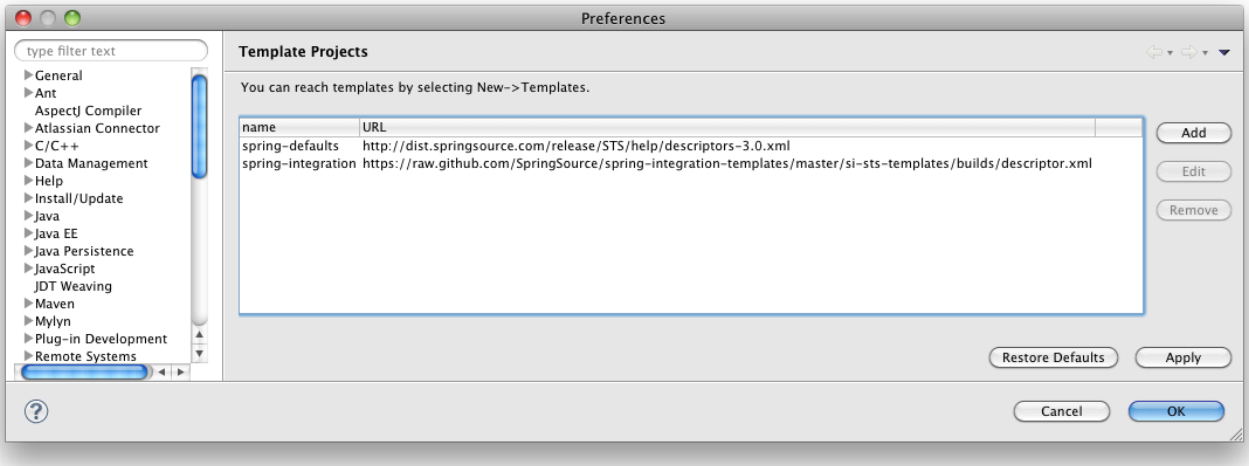
In addition to that you can customize the list of example projects yourself via the Example Projects preferences.

Template Project Enhancements

Template projects can now be categorized, giving you a better overview of the different template projects that are available.



In addition to that you can now easily customize from where to consume those template projects via a preference page:



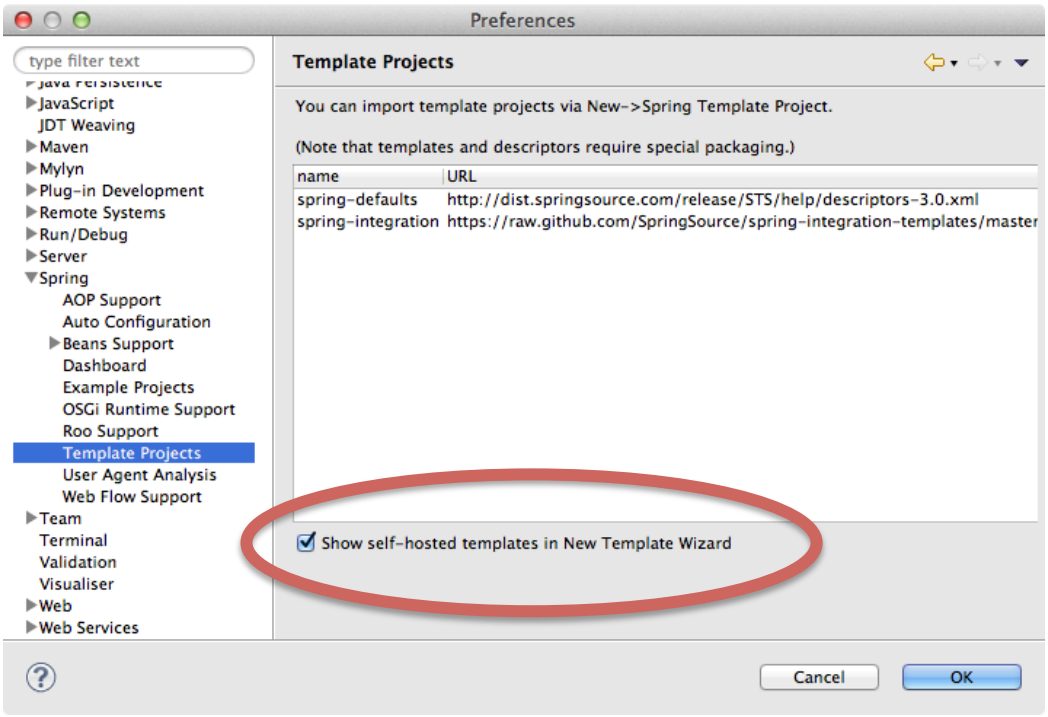
You can use this page to add your own template projects, or perhaps company-specific template projects.

In addition to this you can now also host your template projects on GitHub, for example, using a plain project layout. There is no longer a need to setup a build and create template.zip files for them.

Template project self-hosting

The authoring of your own custom Spring Template Projects gets a lot easier now by offering the so-called "self-hosting" mode. This allows you to author template projects inside your workspace and immediately test it by creating a new project based on your template project – as if this new template project would have been deployed somewhere. This reduces turn-around cycles when creating your own project templates tremendously.

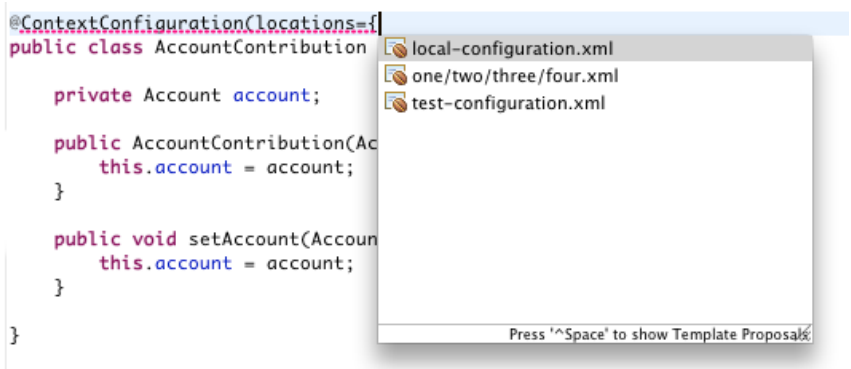
All you need to do is to enable the self-hosting mode in the Spring Template preferences:



A more comprehensive step-by-step blog-post about this will follow shortly.

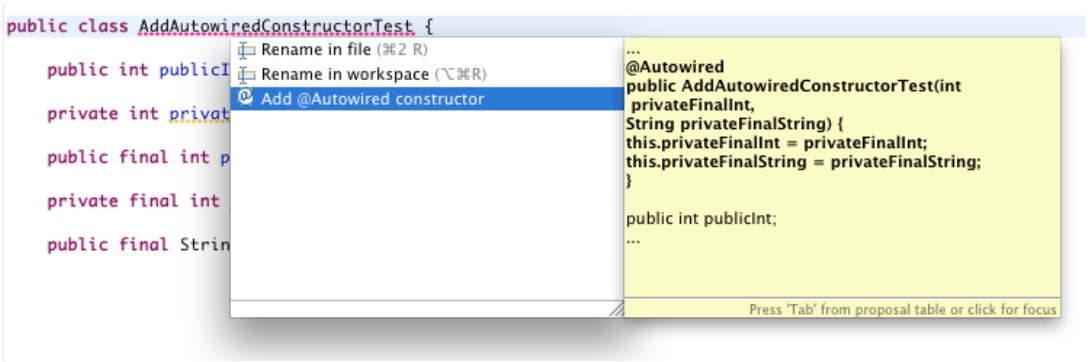
Location autocomplete for @ContextConfiguration

STS now supports auto-completion for the locations (or its alias value) in a @ContextConfiguration annotation:



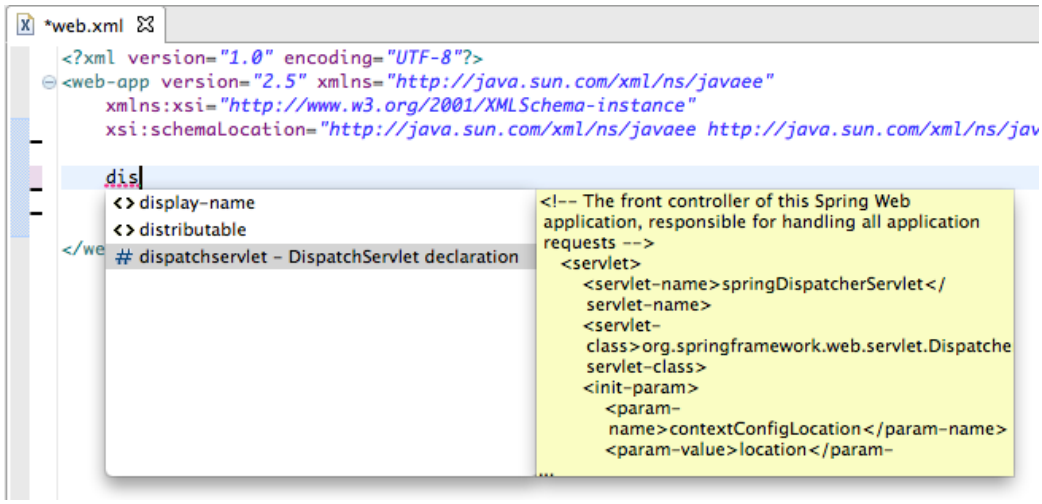
Quickfix to create autowired constructor

There is now a quickfix to create an autowired constructor for all final, non-static, non-initialized variables. Quickfix will present this option if there is no constructor, or if there is only the default (i.e. no-argument) constructor.

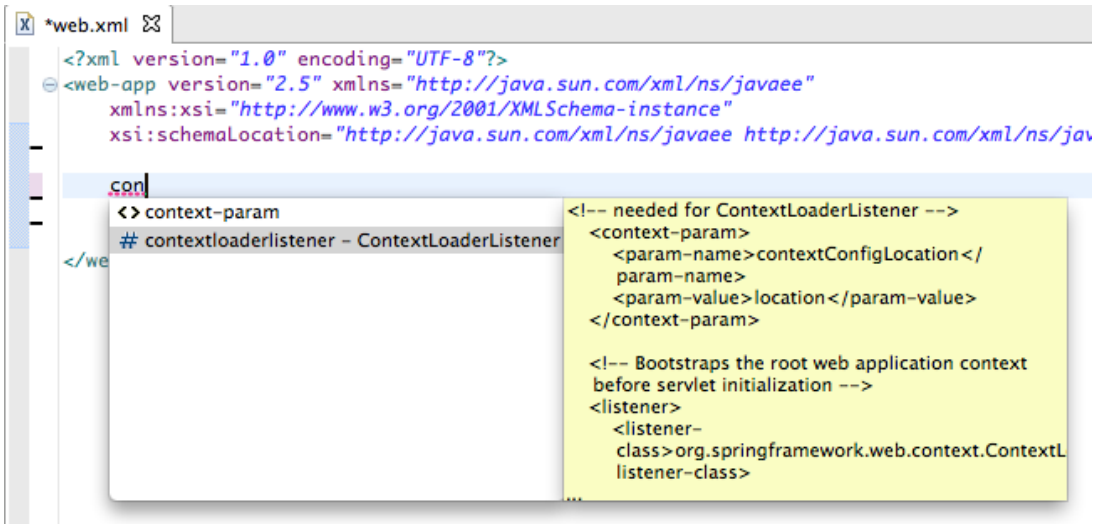


Generate DispatcherServlet and ContextLoaderListener definitions

There are two new content-assist code templates that help you working within the web.xml. The first one generates the typical definition for the DispatcherServlet for you:



The second one generates the default definition for the ContextLoaderListener:



Grails Development Tools

Grails versions

Grails 2.1 is now the version installed from the extension dashboard and is the version supported by the tools. Grails 2.1 is compatible with Groovy 1.8 (not Groovy 2.0) and so Groovy 1.8 is the version of Groovy included in the GGTS distribution. If you wish to try out Groovy 2.0 you can install it into GGTS at any time.

Important project migration notes

With the move to open source for the 3.0.0 version of the tools, many internal names have had to change in the grails tools (package names, plugin names, etc). This has an impact on some of the project metadata maintained by the tools (metadata that you may be sharing in your source code repository). For example, the prefix for preference names is no longer com.springsource but org.grails.

In order to ease the transition, if you open the 3.0.0.M2 version of the tools against a workspace created with an older set of tools, a migration wizard will popup that will help you in migrating your project by performing all the renaming for you. This wizard will only need to run once to migrate the projects in your workspace. If you don't want to run it immediately you can cancel out and run it later via the Project context menu **Configure > Migrate legacy STS projects...**

Obviously you need to be careful committing the migrated changes to your VCS if different members of your team are on different levels of the tools. They should really only be committed once everyone has upgraded to 3.0.0.M2 or later.

Groovy Development Tools

Groovy-Eclipse 2.7.0 release

GGTS now includes the Groovy-Eclipse 2.7.0 release – or it is available from the dashboard if using STS. The full new and noteworthy for Groovy-Eclipse 2.7.0 can be found here:

http://docs.codehaus.org/display/GROOVY/Groovy-Eclipse+2.7.0+New+and+Noteworthy

Groovy 2.0 Compiler

Groovy-Eclipse 2.7.0 now also offers support for the Groovy 2.0 compiler. This is not installed by default in GGTS because GGTS includes the appropriate level for the bundled Grails. GGTS 3.0.0 still includes Grails 2.1 and so GGTS includes Groovy 1.8. For either GGTS or STS the 2.0 compiler support is available through the dashboard.

Groovy 2.0 includes the new support for type checking and static compilation, along with other enhancements, which are all covered in detail here:

http://www.infoq.com/articles/new-groovy-20

Groovy Compiler level checking

In Groovy-Eclipse, each workspace is configured for a single version of the Groovy compiler. It is now possible to set the expected compiler level for each project. This way, if you accidentally import the project into a workspace with an incompatible compiler level, an error marker will be placed on this project.

Once the problem occurs there are a few ways to solve it. The simplest is to select the error in the Markers or Problems view and press Ctrl-1 (Cmd-1 on Mac). This will bring up the quick fix dialog offering multiple solutions to the problem:

- Manually change the expected compiler level for the project so that it matches the workspace level.
- Automatically change the expected compiler level for the project to match the workspace level.
- Open the workspace compiler preferences page and change the workspace compiler level to the project level.
- Or, also in the workspace compiler preferences page, disable the checking for version mismatches.

Organize imports removes implicit imports

Implicitly declared imports, like java.util.* and groovy.lang.* will be removed after performing organize imports on a Groovy file.

Assign statement to new local variable quick fix

There is a new quick fix to assign a statement to a local variable. To invoke the quick fix, select a statement and press CTRL-1 (or CMD-1 on Mac) and the quick fix will be available.

Convert local variable to field refactoring

There is a new 'Convert local variable to field' refactoring.

AspectJ Development Tools

AJDT 2.2.0/AspectJ 1.7.0

Both STS and GGTS now include the AJDT 2.2.0 release, which includes the recently released AspectJ 1.7.0. AspectJ 1.7.0 is the version of AspectJ supporting Java 7 – the features are covered in more detail here:

http://eclipse.org/aspectj/doc/released/README-170.html

Java 7 support

AJDT now fully supports Java7. This includes compiling AspectJ/Java source that exploits Java 7 features, and weaving into Java7 code pre-compiled elsewhere.

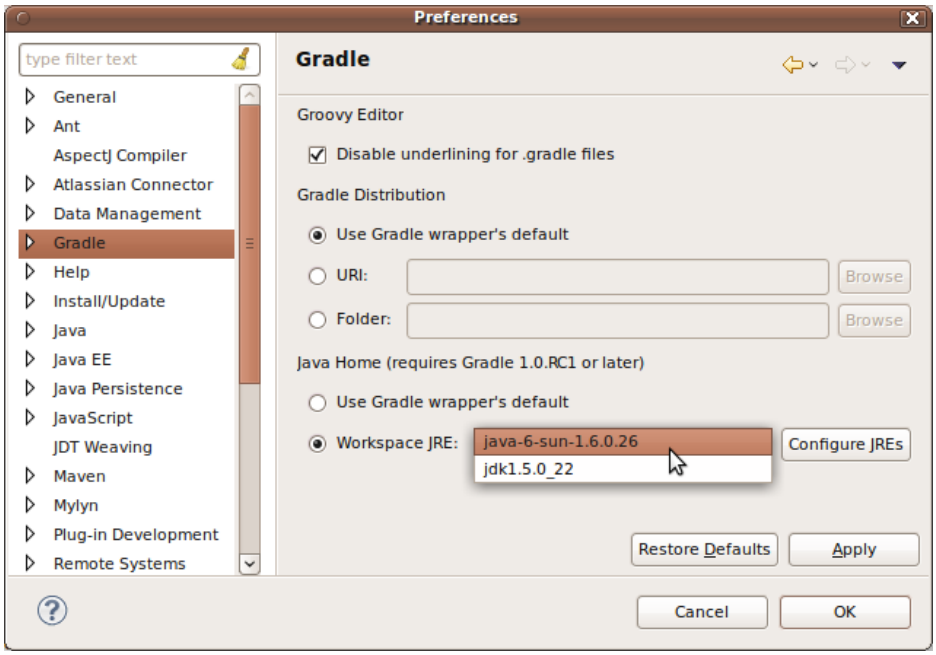
Gradle Tools

Migration

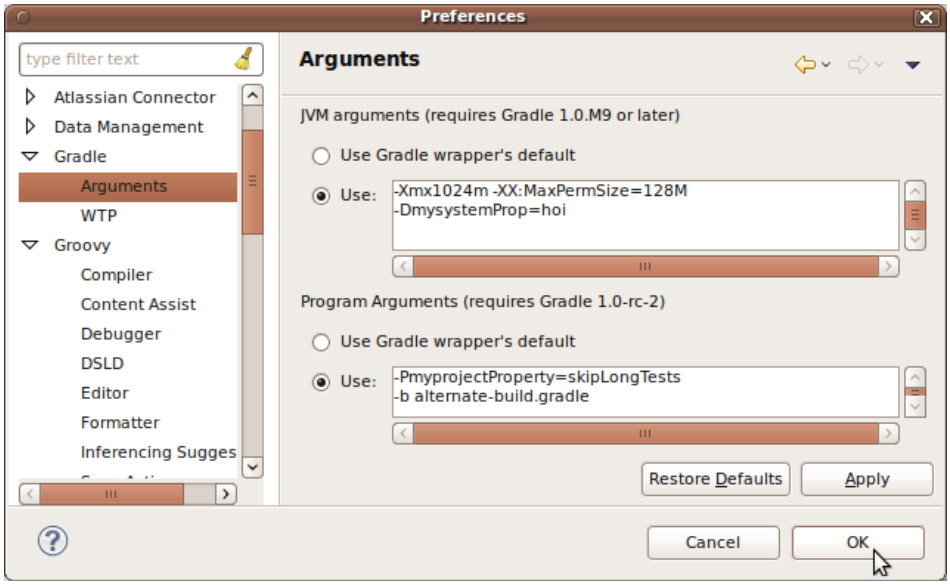
The Gradle tools are now open source and as part of the open sourcing process the package name of the tools has changed to org.springsource.ide.eclipse.gradle. As with the rename that occurred to package/preference names in the Grails tools for this release, this has affected some project metadata like the names of preferences. The Gradle Tools include migration support and will prompt to perform the migration when the old style projects are encountered.

New preferences/options

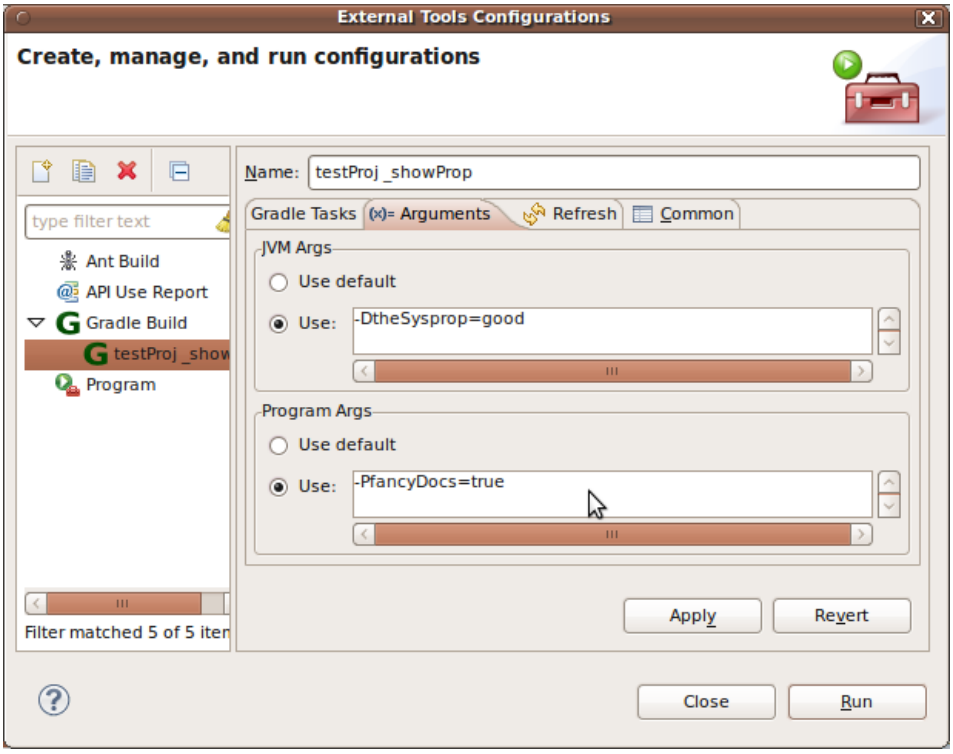
Setting 'JAVA_HOME' as a global preference:



Setting JVM and Gradle program arguments (via global preference):



Setting JVM and program arguments for a specific Launch configuration (this overrides the global settings):



Fixed Bugs and Enhancement Requests

Here is a full list of resolved bugs and enhancement requests for the 3.0.0 release:

Toolsuite issuetracker:

https://issuetracker.springsource.com/secure/IssueNavigator.jspa?reset=true&jqlQuery=project+%3D+STS+AND+fixVersion+in+%2811807%2C+11808%2C+10592%2C+12299%2C+11806%2C+11805%29+AND+status+in+%28Resolved%2C+Closed%29

Spring IDE isssuetracker:

https://jira.springsource.org/secure/IssueNavigator.jspa?reset=true&jqlQuery=project+%3D+IDE+AND+fixVersion+in+%2812626%2C+12627%2C+10788%2C+13011%2C+12625%2C+12624%29+AND+status+in+%28Resolved%2C+Closed%29

New & Noteworthy of previous releases

STS 2.9.x:

http://download.springsource.com/release/STS/doc/STS-new_and_noteworthy-2.9.2.RELEASE.pdf

STS 2.8.x:

http://download.springsource.com/release/STS/doc/STS-new_and_noteworthy-2.8.1.RELEASE.pdf

STS 2.7.x:

http://download.springsource.com/release/STS/doc/STS-new_and_noteworthy-2.7.2.RELEASE.pdf

STS 2.6.x:

http://download.springsource.com/release/STS/doc/STS-new_and_noteworthy-2.6.1.SR1.pdf

STS 2.5.x and before:

http://download.springsource.com/release/STS/doc/STS-new_and_noteworthy-2.5.2.SR1.pdf