

1. Atlassian Connector for Eclipse	2
1.1 Installation and Upgrade Guide for the Eclipse Connector	3
1.1.1 License and Copyright for the Eclipse Connector	4
1.1.2 Installing the Eclipse Connector	4
1.1.2.1 Limited Support for Specific SCMs	13
1.1.3 Upgrading the Eclipse Connector	14
1.1.4 Atlassian Connector for Eclipse - Release Notes	15
1.1.4.1 Atlassian Connector for Eclipse - v1.2 Release Notes	15
1.1.4.2 Atlassian Connector for Eclipse - v1.1 Release Notes	16
1.1.4.3 Atlassian Eclipse Connector 1.0 Release Notes	21
1.1.4.4 Atlassian Eclipse Connector 1.0 Beta Release Notes	37
1.1.4.5 Atlassian IDE Plugin 0.4 for Eclipse - Release Notes	48
1.1.4.6 Atlassian IDE Plugin 0.3 for Eclipse - Release Notes	51
1.1.4.7 Atlassian IDE Plugin 0.2 for Eclipse - Release Notes	52
1.1.4.7.1 Atlassian IDE Plugin 0.2 for Eclipse - Upgrade Notes	53
1.1.4.8 Atlassian IDE Plugin 0.1 for Eclipse - Release Notes	53
1.2 Configuring the Eclipse Connector	54
1.2.1 Collecting Usage Statistics for the Eclipse Connector	62
1.3 Using Bamboo in the Eclipse Connector	65
1.3.1 Configuring your Bamboo Options in Eclipse	66
1.3.2 Working with Bamboo Builds in Eclipse	68
1.3.2.1 Monitoring Build Statuses in Eclipse	71
1.3.2.2 Viewing Build Details in Eclipse	72
1.3.2.3 Running a Build from within Eclipse	73
1.3.2.4 Viewing Test Results in Eclipse	74
1.3.2.5 Viewing a Build Log in Eclipse	76
1.3.2.6 Adding a Task from a Failed Build in Eclipse	78
1.4 Using Crucible in the Eclipse Connector	82
1.4.1 Configuring your Crucible Options in Eclipse	83
1.4.2 Working with Crucible Reviews in Eclipse	85
1.4.2.1 Monitoring Review Updates in Eclipse	90
1.4.2.2 Viewing a Review in Eclipse	91
1.4.2.3 Managing Review Comments in Eclipse	93
1.4.2.4 Working on a Review in the Diff View in Eclipse	97
1.4.2.5 Creating a Post-Commit Review from within Eclipse	100
1.4.2.6 Creating a Patch Review from within Eclipse	102
1.4.2.7 Assigning Reviewers in Eclipse	105
1.4.2.8 Adding Changesets or Patches to Review in Eclipse	107
1.5 Using FishEye in the Eclipse Connector	110
1.5.1 Configuring your FishEye Options in Eclipse	111
1.5.2 More about Configuring FishEye Repositories in Eclipse	113
1.5.3 Working with your FishEye Repository View in Eclipse	114
1.6 Using JIRA in the Eclipse Connector	116
1.6.1 Configuring your JIRA Options in Eclipse	116
1.6.2 Working with JIRA Issues in Eclipse	119
1.7 Eclipse Connector FAQ	119
1.7.1 Configuration FAQ for the Eclipse Connector	119
1.7.1.1 Using NTLM Authentication with the Eclipse Connector	120
1.7.2 Usage FAQ for the Eclipse Connector	120
1.7.3 Installation FAQ for the Eclipse Connector	120
1.7.3.1 Error while Collecting Items to be Installed	120
1.7.3.2 Installation Error - Cannot Find a Solution Satisfying the Following Requirements	121

Atlassian Connector for Eclipse



New name: Atlassian Connector for Eclipse

We are changing the connector's name, from 'Atlassian Eclipse Connector' to '**Atlassian Connector for Eclipse**'. The new name complies with the Eclipse Foundation's [Guidelines for Eclipse Logos & Trademarks](#).

Atlassian Connector for Eclipse — Documentation

- [Installation Guide](#)
- [Upgrade Guide](#)
- [Configuration Guide](#)
- [FAQ](#)
- [Release Notes](#)

About the Atlassian Connector for Eclipse

The Atlassian Connector for Eclipse is an [Eclipse](#) plugin. It allows you to work with the Atlassian products within your Eclipse IDE. Now you don't have to switch between websites, email messages and news feeds to see what's happening to your project and your code. Instead, you can see the relevant JIRA issues, [Crucible](#) reviews and [Bamboo](#) build information right there in your development environment.



Current released version:

Version **1.2** of the Atlassian Connector for Eclipse has now been released — see the [release notes](#).

Resources

Support and bug reporting:

- If you encounter a problem using the Atlassian Connector for Eclipse, please log a [support request](#) or contact our [support team](#).
- If you find a bug, please raise a [bug report](#).

Source:

- This is an open source project. To access the connector's source code, point your SVN at: <https://studio.atlassian.com/svn/PLE/trunk>.
- If you don't already have an account, please go to <http://studio.atlassian.com> and sign up for an account. You should then be able to access the svn repository with your new account.

Other handy links:

- [Atlassian IDE Connector Forum](#)

Offline Versions of the Documentation

You can [download the connector documentation](#) in PDF, HTML or XML formats.

Looking for Other Plugins?

If you're in the wrong place, try these links instead:

- [Atlassian Connector for IntelliJ IDEA](#) — see [Atlassian IntelliJ Connector](#)
- [Clover IDE plugins](#) — see the [Clover documentation](#)
- [Plugins for Atlassian products](#) — see plugin information for [Confluence](#), [JIRA](#), [Crowd](#), [Bamboo](#)
- [Resources for plugin developers](#) — see [Atlassian Developer Network](#)

Table of Contents

- [Installation and Upgrade Guide for the Eclipse Connector](#)
 - [License and Copyright for the Eclipse Connector](#)
 - [Installing the Eclipse Connector](#)
 - [Limited Support for Specific SCMs](#)
 - [Upgrading the Eclipse Connector](#)
 - [Atlassian Connector for Eclipse - Release Notes](#)
 - [Atlassian Connector for Eclipse - v1.2 Release Notes](#)
 - [Atlassian Connector for Eclipse - v1.1 Release Notes](#)
 - [Atlassian Eclipse Connector 1.0 Release Notes](#)
 - [Atlassian Eclipse Connector 1.0 Beta Release Notes](#)
 - [Atlassian IDE Plugin 0.4 for Eclipse - Release Notes](#)
 - [Atlassian IDE Plugin 0.3 for Eclipse - Release Notes](#)
 - [Atlassian IDE Plugin 0.2 for Eclipse - Release Notes](#)
 - [Atlassian IDE Plugin 0.2 for Eclipse - Upgrade Notes](#)
 - [Atlassian IDE Plugin 0.1 for Eclipse - Release Notes](#)
- [Configuring the Eclipse Connector](#)
 - [Collecting Usage Statistics for the Eclipse Connector](#)
- [Using Bamboo in the Eclipse Connector](#)
 - [Configuring your Bamboo Options in Eclipse](#)
 - [Working with Bamboo Builds in Eclipse](#)
 - [Monitoring Build Statuses in Eclipse](#)
 - [Viewing Build Details in Eclipse](#)
 - [Running a Build from within Eclipse](#)
 - [Viewing Test Results in Eclipse](#)
 - [Viewing a Build Log in Eclipse](#)
 - [Adding a Task from a Failed Build in Eclipse](#)
- [Using Crucible in the Eclipse Connector](#)
 - [Configuring your Crucible Options in Eclipse](#)
 - [Working with Crucible Reviews in Eclipse](#)
 - [Monitoring Review Updates in Eclipse](#)
 - [Viewing a Review in Eclipse](#)
 - [Managing Review Comments in Eclipse](#)
 - [Working on a Review in the Diff View in Eclipse](#)
 - [Creating a Post-Commit Review from within Eclipse](#)
 - [Creating a Patch Review from within Eclipse](#)
 - [Assigning Reviewers in Eclipse](#)
 - [Adding Changesets or Patches to Review in Eclipse](#)
- [Using FishEye in the Eclipse Connector](#)
 - [Configuring your FishEye Options in Eclipse](#)
 - [More about Configuring FishEye Repositories in Eclipse](#)
 - [Working with your FishEye Repository View in Eclipse](#)
- [Using JIRA in the Eclipse Connector](#)
 - [Configuring your JIRA Options in Eclipse](#)
 - [Working with JIRA Issues in Eclipse](#)
- [Eclipse Connector FAQ](#)
 - [Configuration FAQ for the Eclipse Connector](#)
 - [Using NTLM Authentication with the Eclipse Connector](#)
 - [Usage FAQ for the Eclipse Connector](#)
 - [Installation FAQ for the Eclipse Connector](#)
 - [Error while Collecting Items to be Installed](#)
 - [Installation Error - Cannot Find a Solution Satisfying the Following Requirements](#)

Installation and Upgrade Guide for the Eclipse Connector

- [License and Copyright for the Eclipse Connector](#)
- [Installing the Eclipse Connector](#)

- [Upgrading the Eclipse Connector](#)
- [Atlassian Connector for Eclipse - Release Notes](#)

License and Copyright for the Eclipse Connector

Open Source

This is an open source project. You can get the source code from our [SVN repository](#).

Copyright Statement

© Atlassian 2008

License

Licensed under the Eclipse Public License, Version 1.0 (the "License").

You may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.eclipse.org/legal/epl-v10.html>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Installing the Eclipse Connector

Below are the system requirements and installation instructions for the Atlassian Connector for Eclipse.


On this page:



- [System Requirements](#)
- [Eclipse Update Sites](#)
- [Offline Installation](#)
- [Installation for Eclipse 3.5 \(Galileo\)](#)
 - [Installation via Mylyn Connector Discovery Wizard](#)
 - [Installation via Eclipse 3.5 Software Updates Manager](#)
- [Installation for Eclipse 3.4 \(Ganymede\)](#)
- [Installation for Eclipse 3.3 \(Europa\)](#)
- [Troubleshooting the Installation](#)
 - [Installation Error - Cannot Find a Solution Satisfying the Following Requirements](#)
 - [Error while Collecting Items to be Installed](#)
 - [Full List of Installation FAQ](#)

System Requirements

Version Matrix

The Atlassian Connector for Eclipse is compatible with the following software versions:

	Atlassian Connector for Eclipse — version 1.1.x
Bamboo	1.x 2.x recommended
Crucible	1.6 or later  We recommend that you upgrade to the latest version of Crucible 1.6.x or 2.x for best results. However, please note that the new features of Crucible 2.0 are not reflected in the connector, such as iterative reviews, read/unread support, etc. Existing Crucible 1.6.x functionality will work when you connect to a Crucible 2.0 server.
Eclipse	3.3 (Europa) 3.4 (Ganymede) 3.5 (Galileo)

FishEye	1.6 or later  We recommend that you upgrade to the latest version of FishEye 1.6.x or 2.x for best results.
JIRA	3.4 or later 3.7 or later recommended  JIRA 4.0 is still in beta and so is not supported. We will try to work with you if you use the connector on JIRA 4 Beta, but JIRA 4.0 Beta itself is not supported. Please note that the new features of JIRA 4.0 are not reflected in the connector, such as JQL. Existing JIRA 3.x functionality should work when you connect to a JIRA 4.0 server.

Other Dependencies

- The Atlassian Connector for Eclipse makes use of [Mylyn](#)'s task-focused interface. Mylyn is shipped with the more recent versions of Eclipse. The Atlassian Connector for Eclipse requires **Mylyn 3.2.x**.
 - If you do not already have Mylyn, it will be installed automatically when you install one of the Atlassian Connector's features.
 - If you already have Mylyn installed, please check your version of Mylyn. If your version of Mylyn is earlier than 3.2, please upgrade to [Mylyn 3.2.x](#).
 - The Atlassian Connector for Eclipse now includes the [JIRA Mylyn connector](#), which was previously available as a JIRA-only connector.
 - The Mylyn documentation includes a [quick start guide](#).
- For Crucible reviews and FishEye source repository viewing, the connector integrates with [Subversion](#) using [Subclipse](#) or [Subversive](#) for source control and file retrieval.



Limited Support for CVS

The Atlassian Connector for Eclipse provides limited support for [CVS](#). See [Limited Support for Specific SCMs](#).



Note: Install either Subclipse or Subversive integration option, not both

When selecting components of the Atlassian Connector for Eclipse, please select either **Subclipse** or **Subversive**, not both. (Installation details are below.) Installing one or the other of these two integration options will allow you to make full use of the Atlassian Connector's integration with Crucible for code reviews and FishEye for source repository viewing, when versioning your project under SVN.

Eclipse Update Sites

Add one of the following URLs to your Eclipse update sites, depending on your version of Eclipse:

- For Eclipse 3.3 — <http://update.atlassian.com/atlassian-eclipse-plugin/e3.3>
- For Eclipse 3.4 and 3.5 — <http://update.atlassian.com/atlassian-eclipse-plugin/e3.4>

There are also weekly update sites available, for those who are keen to get the latest version of the connector. Builds are executed every Sunday at approximately 2am:

- For Eclipse 3.3 — <http://update.atlassian.com/atlassian-eclipse-plugin/weekly/e3.3/site.xml>
- For Eclipse 3.4 and 3.5 — <http://update.atlassian.com/atlassian-eclipse-plugin/weekly/e3.4/site.xml>

Offline Installation

If you cannot use our update site (e.g. due to firewall issues) or just prefer local installation, we offer a zipped version of the entire update site, including the Mylyn dependencies. This installation supports **Eclipse 3.4 and 3.5** only.

- First install the [Subclipse](#) or [Subversive](#) plugin for Eclipse, if you version your project under [SVN](#) and if you do not already have one of these plugins installed in your IDE. This will allow you to take advantage of the connector's Crucible and FishEye integration.
- Download the Atlassian connector's [archived update site](#).
- In Eclipse, open **'Help', 'Install New Software', 'Add'**, click the **'Archive'** button and select the downloaded zip file.
- Continue with the installation steps as described for the regular update site — see instructions below.




Please note: In the zipped update site we do not include Subclipse or Subversive. You will need to install them separately **before** installing Atlassian Connector, if you want to take advantage of the connector's Crucible and FishEye integration.

Installation for Eclipse 3.5 (Galileo)

If you are using Eclipse 3.5 (Galileo), you can install the connector via the Mylyn Connector Discovery wizard or via the Eclipse software updates manager. Both methods are described below.

Installation via Mylyn Connector Discovery Wizard

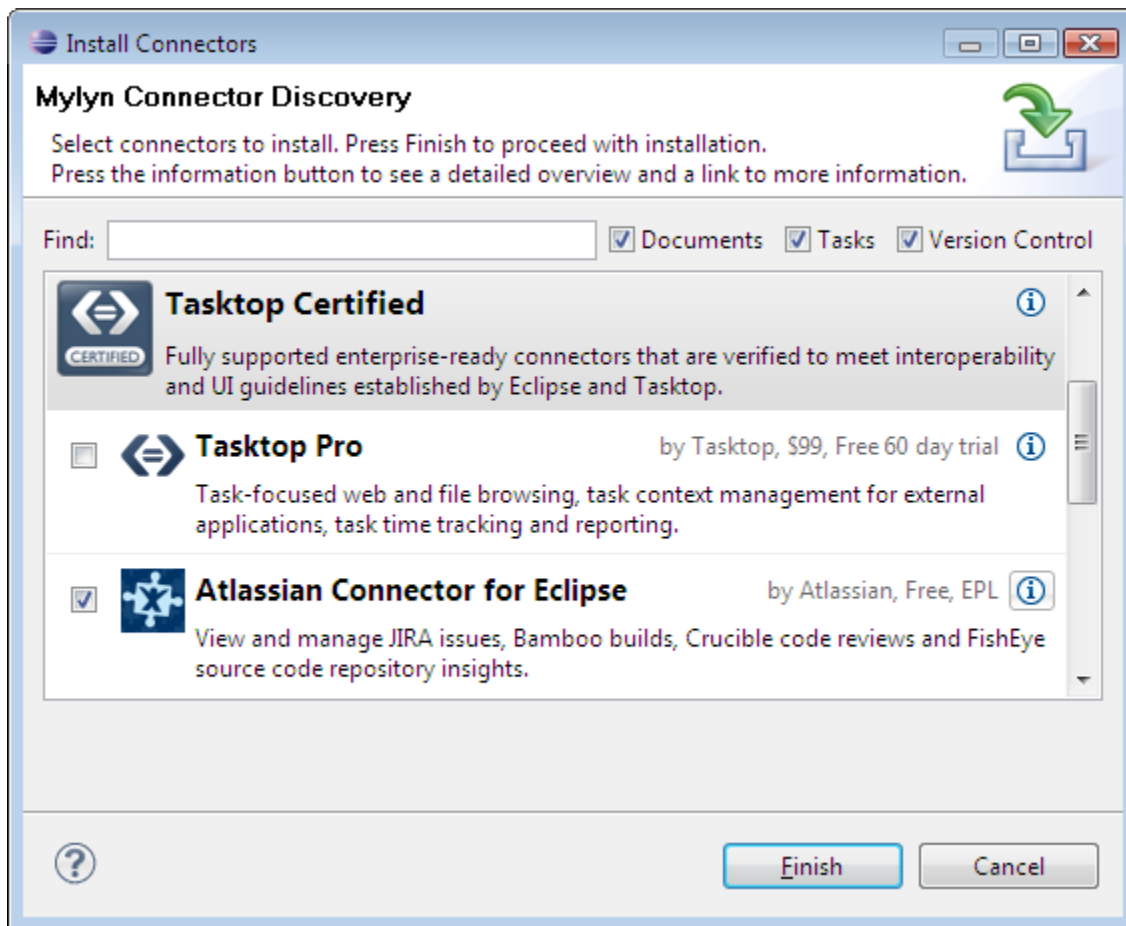
Mylyn 3.2 introduced the new Connector Discovery wizard that provides simple installation for Mylyn extensions.

 The Mylyn Connector Discovery wizard is available in Eclipse 3.5 only.

To install the Atlassian Connector for Eclipse via the Mylyn Connector Discovery wizard:

1. Ensure that you have already installed Mylyn 3.2.x. (If you are using an Eclipse package from the [Eclipse download site](#), Mylyn 3.2 is already included in any package except the Classic download.)
2. In Eclipse Mylyn, open the '**Task Repositories**' view. (In Eclipse, click '**Windows**', '**Show View**', '**Other**' and select the '**Task Repositories**' view from the '**Tasks**' category.)
3. Click the '**Add Task Repository**' icon.
4. The '**Add Task Repository**' screen appears. Click the '**Install More Connectors**' button.
5. The '**Mylyn Connector Discovery**' screen appears, as shown below.
6. Select the Atlassian Connector and click '**Finish**' to install it.

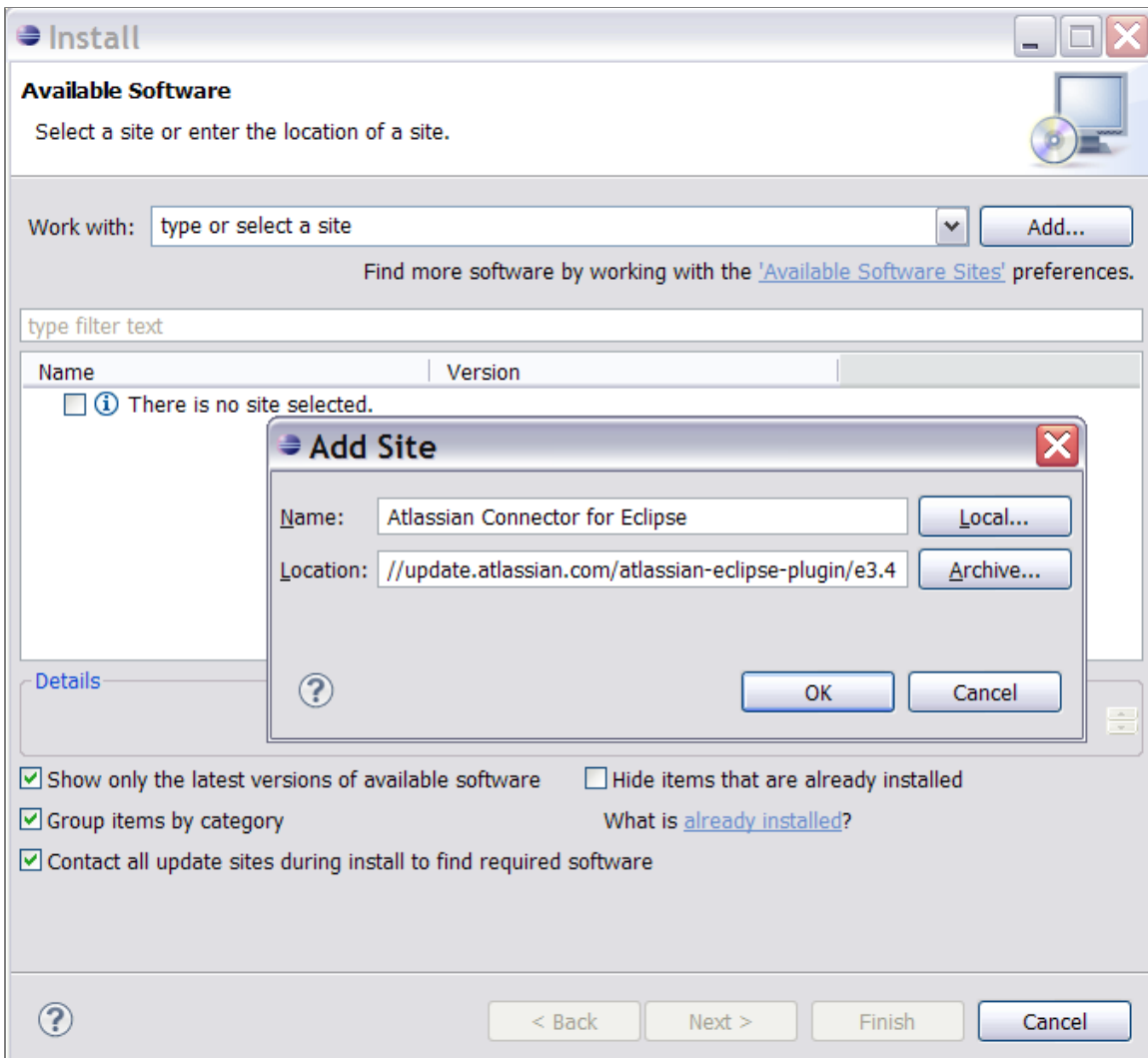
Screenshot: The Mylyn Connector Discovery wizard



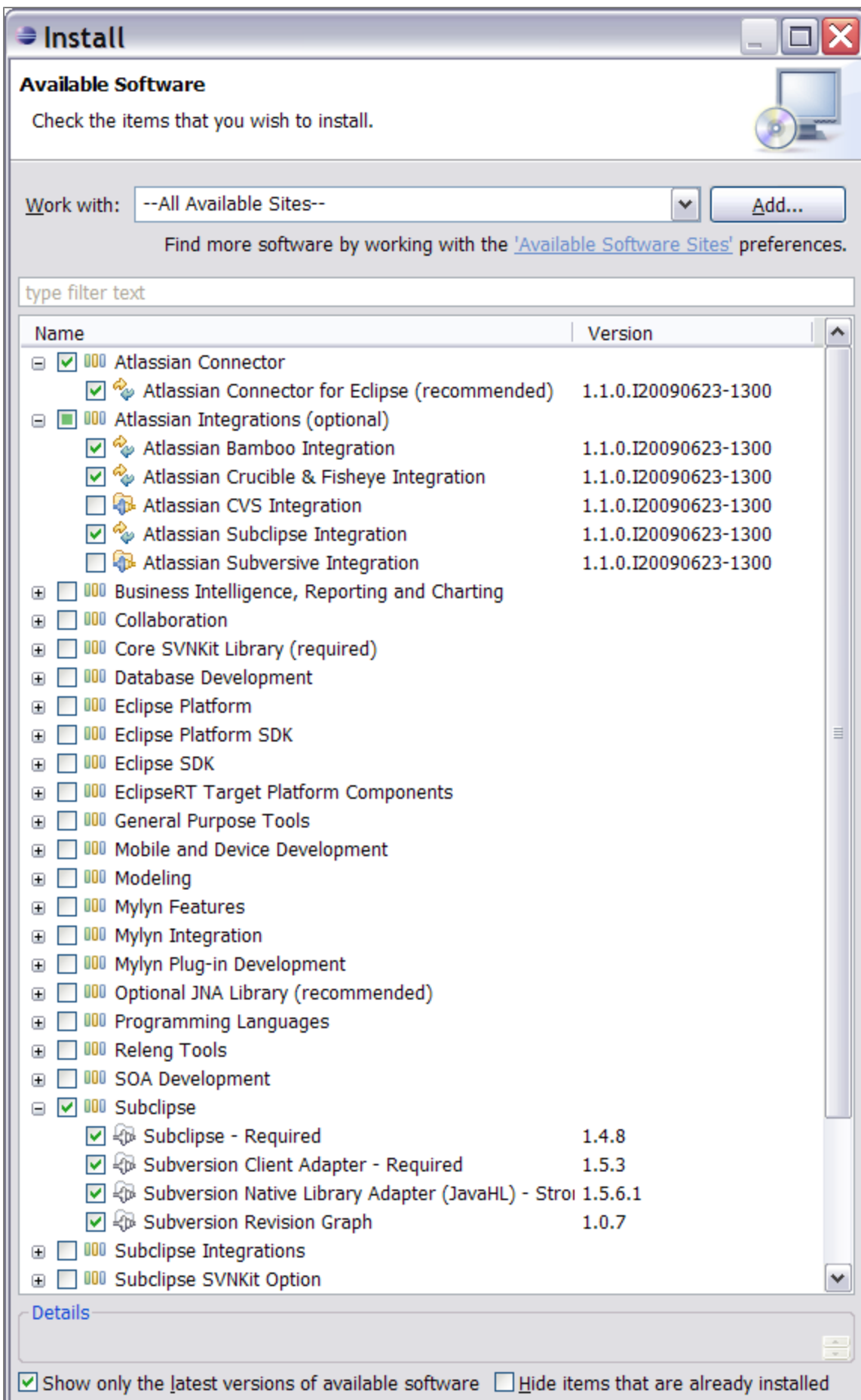
Installation via Eclipse 3.5 Software Updates Manager

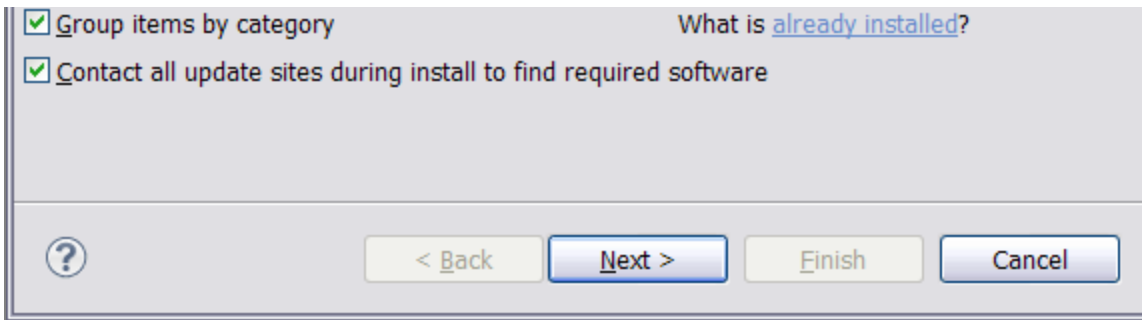
To install the connector on a clean Eclipse 3.5 (Galileo) IDE for Java Developers:

1. Open the Eclipse software updates manager via '**Help**', '**Install New Software**'.
2. Click '**Add**', enter the details of the update site, as shown in the screenshot below:



3. Click 'OK'.
4. The 'Available Software' screen appears again. Select '~~All Available Sites~~' in the 'Work with' dropdown list.
5. Select both 'Atlassian Connector' categories. Also select either 'Subclipse' or 'Subversive', depending on which of the two you are using, as shown in the screenshot below:



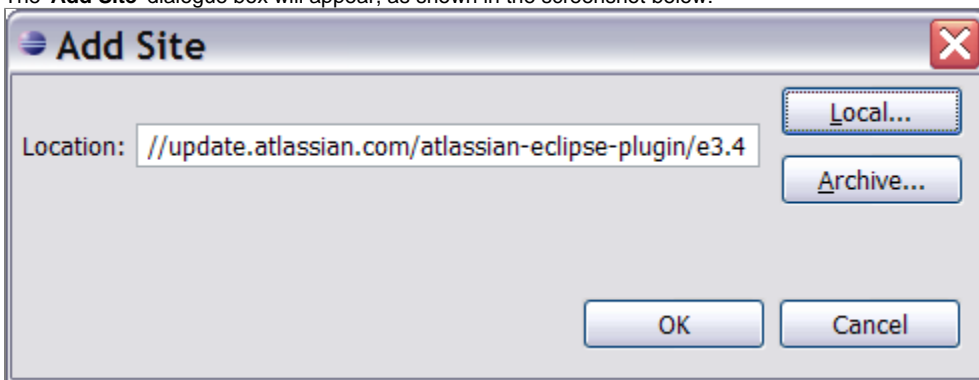


6. Click '**Next**' and follow the installation prompts.
7. Restart Eclipse when prompted.

Installation for Eclipse 3.4 (Ganymede)

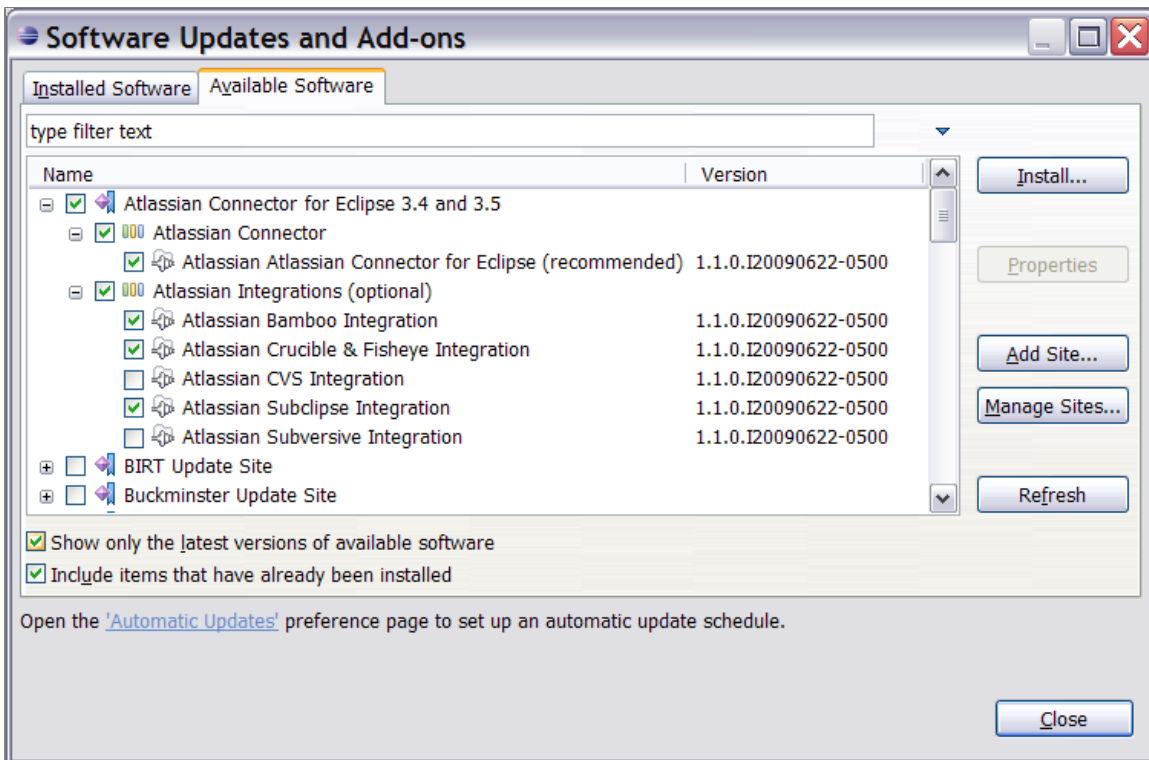
The steps below show the full installation procedure in Eclipse 3.4 (Ganymede):

1. If you already have Mylyn installed:
 - Check your version of Mylyn as described under dependencies [above](#).
 - Open the Eclipse software updates manager via '**Help**', '**Software Updates**', '**Installed Software**' and click '**Update**' to get the latest Mylyn updates.
 - Restart Eclipse.
2. Open the Eclipse software updates manager via '**Help**', '**Software Updates**', '**Available Software**'.
3. Click the '**Manage Sites**' button.
4. Scroll down to find the '**Mylyn Extras**' update site some where near the bottom, enable it and click '**OK**'.
5. Back on the '**Available Software**' tab, click the '**Add Site**' button.
6. The '**Add Site**' dialogue box will appear, as shown in the screenshot below:



Enter the URL of the Atlassian Connector update site:
<http://update.atlassian.com/atlassian-eclipse-plugin/e3.4>

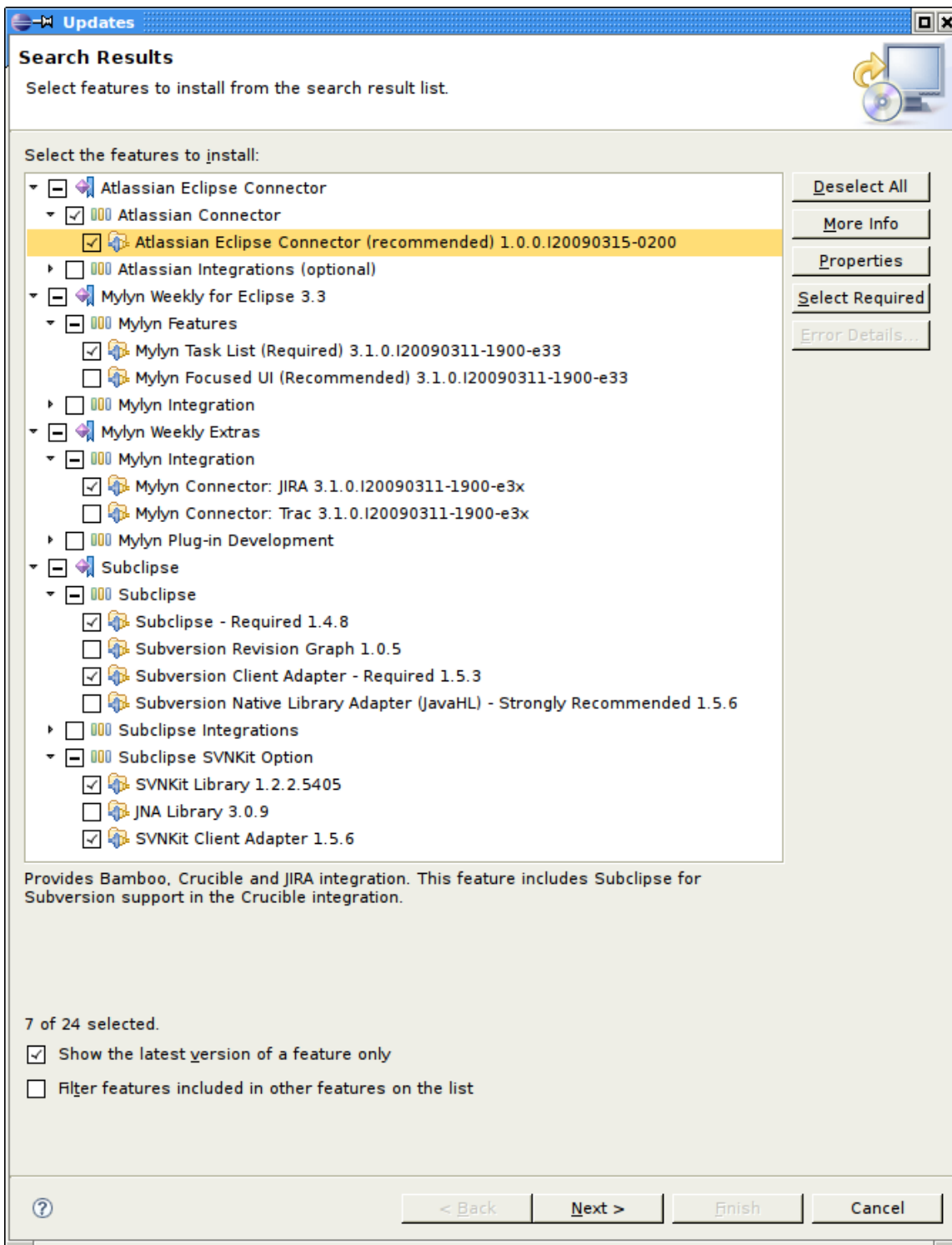
7. Click '**OK**'.
8. The 'Atlassian Connector for Eclipse' will appear in the list of available software, as shown in the screenshot below:



9. Select the required features for the connector:
 - Atlassian Connector for Eclipse — Required. This is the core of the Atlassian Connector.
 - Bamboo Integration — Allows you to view and manage your [Bamboo](#) builds within your Eclipse IDE.
 - Crucible and FishEye Integration — Allows you to view and manage your [Crucible](#) code reviews within your Eclipse IDE, and to move easily to a [FishEye](#) view of your repository.
 - CVS Integration — Select this integration option if you version your project under [CVS](#).
 - Subclipse Integration — Select this integration option if you version your project under [SVN](#) with [Subclipse](#).
 - Subversive Integration — Select this integration option if you version your project under [SVN](#) with [Subversive](#).
 - The installation process will automatically provide the connector's JIRA integration, allowing you to view and manage your [JIRA](#) issues within your Eclipse IDE.
10. Click the 'Install' button, then click 'Finish' when asked to confirm the installation.
11. If prompted, accept the license agreement and click the 'Next' button.
12. Restart Eclipse when prompted.
13. The Atlassian Connector for Eclipse has now been installed. Your next step is to add your Bamboo, Crucible and JIRA task repositories in Eclipse. See [Configuring the Eclipse Connector](#).

Installation for Eclipse 3.3 (Europa)

We have not included detailed instructions for installation under Eclipse 3.3 (Europa). The screenshot below shows the selection of required dependencies:



Troubleshooting the Installation

Installation Error - Cannot Find a Solution Satisfying the Following Requirements

Symptoms

During installation you may receive an error message similar to one of these:

- Cannot complete the request. See the details.
Cannot find a solution satisfying the following requirements Match[requiredCapability:
org.eclipse.equinox.p2.iu/org.eclipse.swt/[3.4.0.v3448f,3.4.0.v3448f]].

- Cannot complete the request. See the details.
Cannot find a solution satisfying the following requirements org.eclipse.pde.feature.jar [3.4.0.v20080603-7T7U-E_EkMNRNkb39WcMC].
- Cannot complete the install because one or more required items could not be found.
Software being installed: Atlassian Connector for Eclipse (recommended) 1.1.0.v20090624 (com.atlassian.connector.eclipse.feature.group 1.1.0.v20090624)
Missing requirement: Atlassian Connector for Eclipse (recommended) 1.1.0.v20090624 (com.atlassian.connector.eclipse.feature.group 1.1.0.v20090624) requires 'org.eclipse.mylyn.jira_feature.feature.group [3.2.0.I20090529,3.3.0]' but it could not be found

Cause

This is most likely an error from P2 (Eclipse's update mechanism in Eclipse 3.4 and higher). The problem seems to be that the Mylyn update site is not enabled in your copy of Eclipse. There is a bug in Eclipse that these sites may not be automatically enabled even though we specify that we require them for installation.

If any of the installed features have unsatisfied dependencies or if features were previously installed from the extras or incubator update site, P2 may fail with an error similar to the one above.

Resolution

Follow these steps to enable the required sites:

1. In Eclipse, open **'Help', 'Software Updates'**.
2. Click the **'Available Software'** tab.
3. Click **'Manage Sites'**.
4. Look for either the **'Mylyn Extras'** update site or a URL that looks like this:

```
http://download.eclipse.org/tools/mylyn/update/extras
```

5. Make sure that the site or URL is checked.
6. Look for either the **'Mylyn For Eclipse 3.4'** update site or a URL that looks like this:

```
http://download.eclipse.org/tools/mylyn/update/e3.4
```

7. Make sure that the site or URL is checked.
8. Now you should update all installed features to the latest version. Click the **'Installed Software'** tab.
9. Click **'Update'**.
10. Restart if prompted.
11. Try to install the Atlassian Connector for Eclipse again.

More Information

- Issue [PLE-576](#).
- Issue [PLE-393](#).
- Issue [PLE-506](#).
- The [Mylyn FAQ](#).

Error while Collecting Items to be Installed

Symptoms

During installation, after the license agreement screen, you may receive an error message similar to this:

```
An error occurred while collecting items to be installed
No repository found containing: com.atlassian.connector.commons/osgi.bundle/0.5.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.subclipse/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.subclipse.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.thoughtworks.xstream/osgi.bundle/1.3.0.v20090224-1640
No repository found containing: org.apache.commons.io/osgi.bundle/1.3.2.v20080604-1500
```


No repository found containing: org.jdom_jaxen/osgi.bundle/1.0.0.20081203-1100
No repository found containing: org.joda.time/osgi.bundle/1.6.0.v20081202-0100

Cause

P2 (Eclipse's update mechanism in Eclipse 3.4 and higher) has cached some incorrect information about the update site where the plugins reside for the Atlassian Connector for Eclipse.

Resolution

Follow these steps to clear the cached information:

1. Remove the Atlassian update site:

```
http://update.atlassian.com/atlassian-eclipse-plugin/e3.4
```

2. Add the Atlassian update site again:

```
http://update.atlassian.com/atlassian-eclipse-plugin/e3.4
```

3. Try to install the Atlassian Connector for Eclipse again.

More Information

See issue [PLE-393](#) in our bug tracker.

Full List of Installation FAQ

Above we have listed some common problems and their solutions in full. If your problem is not described above, please take a look at the list of installation FAQ, in case additional FAQs have been added since this document was written:

- [Error while Collecting Items to be Installed](#)
- [Installation Error - Cannot Find a Solution Satisfying the Following Requirements](#)

RELATED TOPICS

- [Configuring the Eclipse Connector](#)
- [License and Copyright for the Eclipse Connector](#)
- [Installing the Eclipse Connector](#)
- [Upgrading the Eclipse Connector](#)
- [Atlassian Connector for Eclipse - Release Notes](#)

[IDE Connector Documentation](#)

Limited Support for Specific SCMs

Currently the Atlassian Connector for Eclipse provides *full* support for code reviews where you version your project under [SVN](#) using [Subclipse](#) or [Subversive](#). We provide *limited* support for [CVS](#), as described below.



We appreciate your feedback

If you are using CVS, please let us know of any problems or give us other feedback. You can raise issues and requests on our [issue tracker](#).

On this page:

- [Full Support for SVN with Subclipse or Subversive](#)
- [Limited Support for CVS](#)
- [Other SCMs not yet Supported](#)

Full Support for SVN with Subclipse or Subversive

The Atlassian Connector for Eclipse provides *full* support for [Subclipse](#) or [Subversive](#) where you version your project under [SVN](#). For a list of supported features, please refer to the [version 1.0](#) and [version 1.1](#) of the Atlassian Connector for Eclipse.

- Only **Subversive 0.7.8 or later** is supported.

Limited Support for CVS

The Atlassian Connector for Eclipse provides *limited* support for Crucible code reviews where you version your project under [CVS](#):

- You can open review files and add comments.
- You **cannot** add new reviews or changesets.

The connector's FishEye functionality provides *full* support for CVS.

Other SCMs not yet Supported

We intend to add support for other SCMs via the connector's integration with Crucible, so that the connector will support the SCMs which Crucible supports.

RELATED TOPICS

[Installing the Eclipse Connector](#)
[IDE Connector Documentation](#)

Upgrading the Eclipse Connector

This page tells you how to upgrade the Atlassian Connector for Eclipse. If you have never installed the connector, please refer to the [installation guide](#) instead.

Replacing Early Versions of the 'Atlassian IDE Plugin'

If you have previously installed version **0.4 or earlier** of the **Atlassian IDE Plugin**:

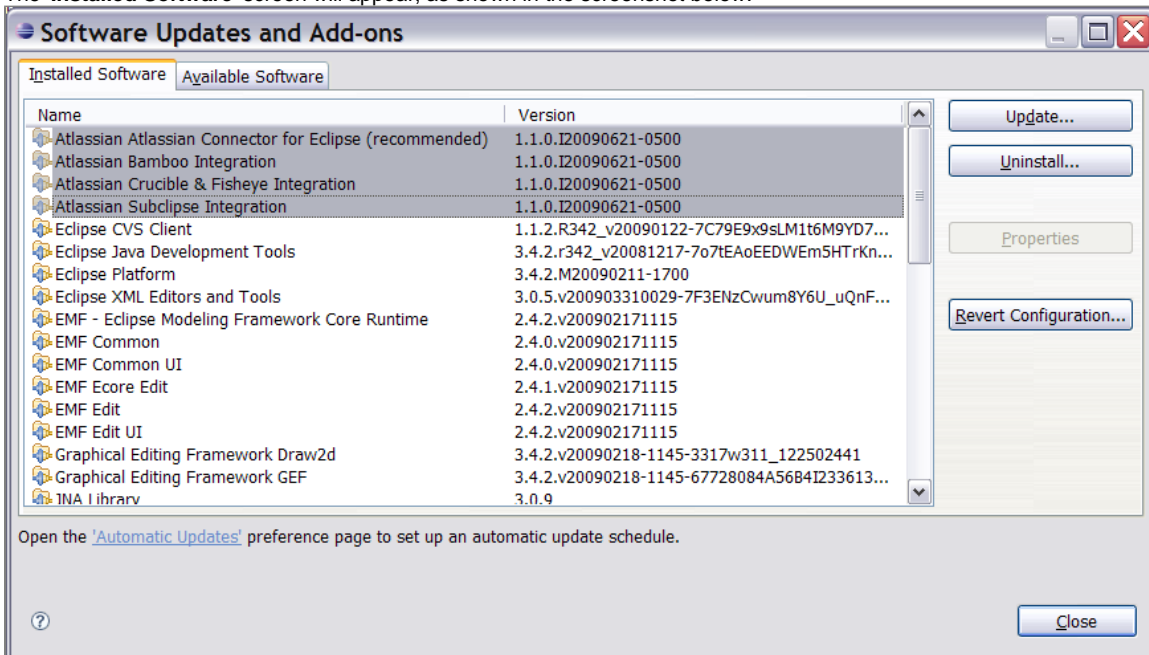
1. Uninstall the earlier version.
2. Then follow the [installation guide](#) to install the new Atlassian Connector for Eclipse.

Upgrading from Version 1.0 or Later

If you are upgrading from version 1.0 Beta, version 1.0 or later versions of the Atlassian Connector for Eclipse, you can follow the standard Eclipse software upgrade procedure.

These are the steps in the standard upgrade procedure for Eclipse 3.4 (Ganymede):

1. Open the Eclipse software updates manager via '**Help**', '**Software Updates**', '**Installed Software**'.
2. The '**Installed Software**' screen will appear, as shown in the screenshot below:



Select the required components of the '**Atlassian Connector for Eclipse**' and click the '**Update**' button.

3. When prompted, verify the installation details and click the '**Next**' button.
4. If prompted, accept the license agreement and click the '**Finish**' button.
5. When prompted, restart Eclipse.

RELATED TOPICS

- [Configuring the Eclipse Connector](#)
- [License and Copyright for the Eclipse Connector](#)
- [Installing the Eclipse Connector](#)
- [Upgrading the Eclipse Connector](#)
- [Atlassian Connector for Eclipse - Release Notes](#)

[IDE Connector Documentation](#)

Atlassian Connector for Eclipse - Release Notes



New name: Atlassian Connector for Eclipse

We are changing the connector's name, from 'Atlassian Eclipse Connector' to '**Atlassian Connector for Eclipse**'. The new name complies with the Eclipse Foundation's [Guidelines for Eclipse Logos & Trademarks](#).



Current released version:

Version **1.2** of the Atlassian Connector for Eclipse has now been released — see the [release notes](#).

- [Atlassian Connector for Eclipse - v1.2 Release Notes](#)
- [Atlassian Connector for Eclipse - v1.1 Release Notes](#)
- [Atlassian Eclipse Connector 1.0 Release Notes](#)
- [Atlassian Eclipse Connector 1.0 Beta Release Notes](#)
- [Atlassian IDE Plugin 0.4 for Eclipse - Release Notes](#)
- [Atlassian IDE Plugin 0.3 for Eclipse - Release Notes](#)
- [Atlassian IDE Plugin 0.2 for Eclipse - Release Notes](#)
- [Atlassian IDE Plugin 0.1 for Eclipse - Release Notes](#)

Atlassian Connector for Eclipse - v1.2 Release Notes

19 August 2009

With pleasure, Atlassian presents **version 1.2 of the Atlassian Connector for Eclipse**.

It's now easier than ever to install the Atlassian Connector for Eclipse. For those who cannot use our update site (for example, those who are working behind a firewall) or just prefer local installation, we now offer a zipped version of the entire update site. There is also a weekly update site available. Refer to our [installation guide](#) for details.

This release introduces the collection of usage statistics to help us better understand how people use the connector, so that we can work on improving the user experience. When you first open your IDE with the connector installed, we ask whether you agree to participate in the collection of usage statistics. You can also change your decision later via the configuration settings, as described in our [configuration guide](#).

Not sure how to use the Crucible, FishEye or Bamboo components of the connector, or want to see what they can do for you? We now have detailed user guides:

- [Working with Crucible Reviews in Eclipse](#)
- [Working with your FishEye Repository View in Eclipse](#)
- [Working with Bamboo Builds in Eclipse](#)

Don't have the Atlassian Connector for Eclipse yet?

You can install the connector directly from the Eclipse software updates manager, via the Mylyn Connector Discovery wizard or from a zipped archive. Follow our [installation guide](#).

This is an open source project. The source code is available in our [SVN repository](#).



Please log your issues, requests and votes. They help us decide what needs doing.

Complete List of Fixes in this Release

JIRA Issues (20 issues)			
Key	Summary	Priority	Status
PLE-598	Create Upload Server		 Resolved
PLE-582	Connector does not pick new revisions of the files under review		 Closed
PLE-637	Query Title should be automatically filled out for predefined queries in New Query dialog		 Resolved
PLE-632	two identical usage reports in New wizard		 Closed
PLE-630	Textual changes on Usage Tracking screens		 Closed
PLE-628	Fix help link on "Usage Data" preferences window		 Closed
PLE-626	Open the Task Repository view opens Task List instead		 Closed
PLE-617	ensure forked version of mylyn usage plugin does not conflict with the released mylyn usage monitor		 Resolved
PLE-613	Create a weekly update site		 Closed
PLE-590	Allow user to get to Eclipse from CRU, FE, and JIRA.		 Resolved
PLE-589	Error trying to open file from Crucible Code Review Editor		 Resolved
PLE-588	New Review wizard fails to get Crucible repos when Save Password is disabled, but correct password was provided when prompted		 Resolved
PLE-587	New Crucible Review wizard after Update Repository Data should not reset Application, and other combo boxes		 Closed
PLE-583	Implement basic usage tracking		 Resolved
PLE-576	associate sites fail to get enabled		 Closed
PLE-573	Endless loop in Crucible New Review after opening repository that didn't have any changesets		 Resolved
PLE-566	dialog for mapping crucible repository is too small		 Resolved
PLE-561	[Collection] JIRA Bugs for M6		 Closed
PLE-554	Bamboo validation prompts for password		 Resolved
PLE-83	integrate clover with PDE build		 Closed

Atlassian Connector for Eclipse - v1.1 Release Notes

24 June 2009

With pleasure, Atlassian presents **version 1.1 of the Atlassian Connector for Eclipse**.

The Atlassian Connector for Eclipse builds on [Mylyn](#)'s task-focused interface to integrate your Eclipse IDE closely with your [JIRA](#) issues, [Crucible](#) code reviews, [FishEye](#) repository views and [Bamboo](#) builds. For a full list of the connector's features, take a look at our [version 1.0 release notes](#).

With the release of **version 1.1**, the Atlassian Connector for Eclipse is now fully compatible with Eclipse 3.5 (Galileo) and [Mylyn 3.2](#).

This release introduces full support for [Subversive](#) where you are versioning your project under [SVN](#). This means that you can work with issues, code reviews, builds and FishEye repository inspection when using the Subversive plugin for Eclipse, as well as the already-supported [Subclipse](#) plugin. Are you using [CVS](#)? The connector's FishEye integration now provides full support for CVS.

You can now log work against a JIRA issue from within your Eclipse IDE. Combined with the connector's existing functionality, this means that you hardly ever need to move out of your IDE to your web browser when managing issues. We have also fixed a number of problems and concentrated on code refactoring.

 We are changing the connector's name, from 'Atlassian Eclipse Connector' to 'Atlassian Connector for Eclipse'. The new name complies with the Eclipse Foundation's [Guidelines for Eclipse Logos & Trademarks](#).

Highlights of this Release:

- Highlights of this Release
 - [Log Work](#) against a JIRA Issue
 - Full Support for Subversive Plugin
 - FishEye Support for CVS
 - Other Points Worth Mentioning
 - Complete List of Fixes in this Release

Don't have the Atlassian Connector for Eclipse yet?

You can install the connector directly from the Eclipse software updates manager or via the Mylyn Connector Discovery wizard. Follow our [installation guide](#).

This is an open source project. The source code is available in [our SVN repository](#).

 **We love your feedback!** 

Please log your [issues](#), [requests](#) and [votes](#). They help us decide what needs doing.

Note: Install either Subclipse or Subversive integration option, not both

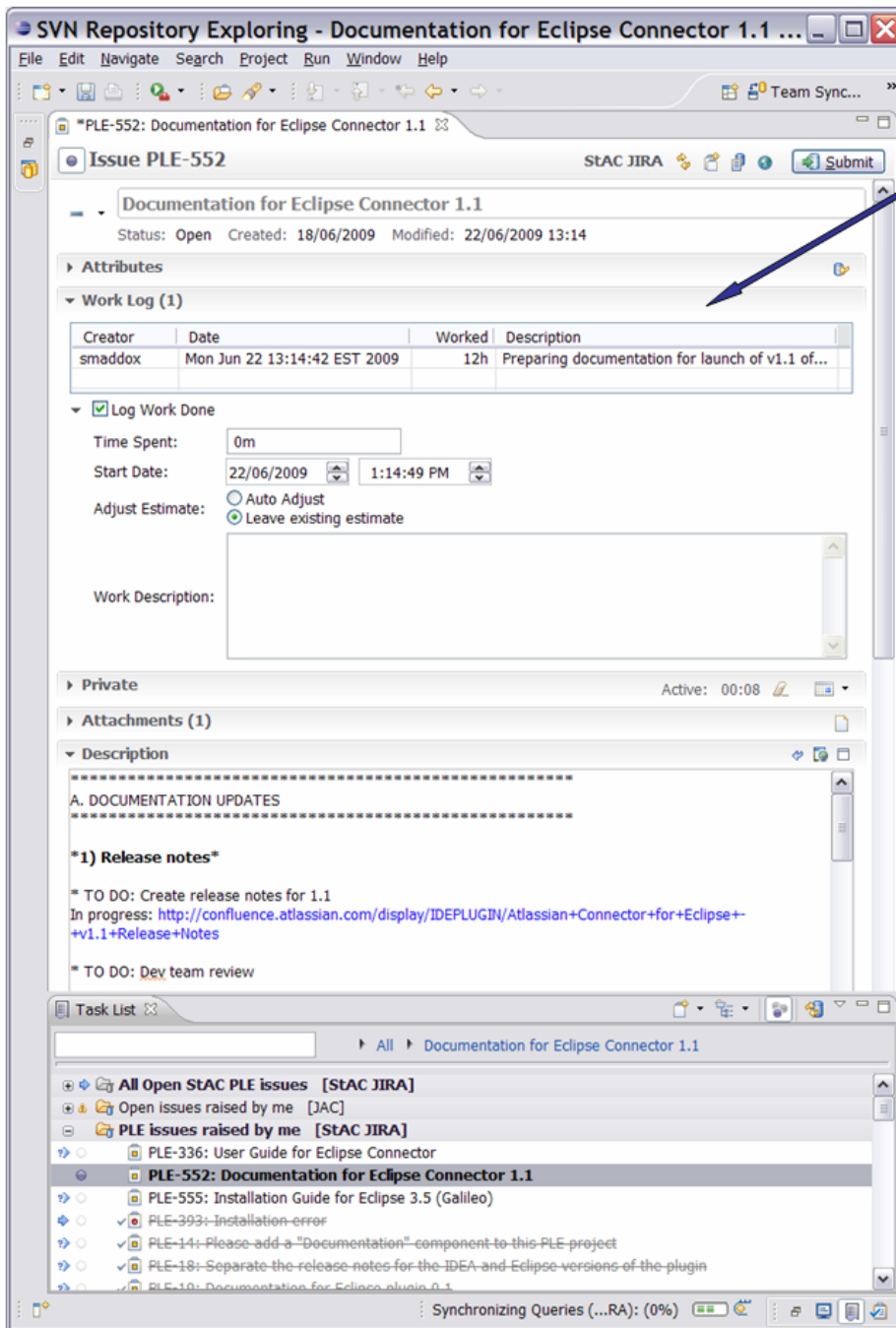
When installing components of the Atlassian Connector for Eclipse, please select either **Subclipse** or **Subversive**, not both. (Details are in our [installation guide](#).) Installing one or the other of these two integration options will allow you to make full use of the Atlassian Connector's integration with Crucible for code reviews and FishEye for source repository viewing, when versioning your project under SVN.

Highlights of this Release



Log Work against a JIRA Issue

You can now log work against a JIRA issue from within your Eclipse IDE. Combined with the connector's existing functionality, this means that you hardly ever need to move out of your IDE to your web browser when managing issues.



Log work on your JIRA issue from within Eclipse

2

Full Support for Subversive Plugin

The connector now provides full support for the [Subversive](#) plugin for Eclipse where you are versioning your project under [SVN](#). Earlier releases of the connector already support [Subclipse](#). This means that you can work with all the connector's Atlassian integrations from within your Eclipse IDE via Subversive or Subclipse.

- Manage JIRA issues.
- Conduct Crucible code reviews.
- Monitor Bamboo builds.
- Use FishEye's awesome repository inspection.

If you would like more details, please refer to our [SCM support guide](#).

3

FishEye Support for CVS

The connector's FishEye integration now provides full support for [CVS](#) within Eclipse.

- Open a source file in FishEye's web interface, directly from your Eclipse editor.
- Copy the FishEye link to your clipboard and share FishEye file references with others.

If you would like more details, please refer to our [SCM support guide](#).

4

Other Points Worth Mentioning


















































- When you are [configuring a FishEye SCM mapping](#), the connector now provides a default SCM path.
- The connector supports **Eclipse 3.5 (Galileo)**, which is due for GA (general availability) release on 24 June.
- The connector supports the new [Mylyn 3.2](#) release, including a more streamlined and flexible bug reporting feature.
- You can now install the connector via Mylyn's new 'Connector Discovery' wizard, as described in our [installation guide](#).
- In the Eclipse software installation dialogue, both Subversive and Subclipse are included in the 'optional' integration features rather than the 'recommended' features for the Atlassian connector.














Complete List of Fixes in this Release

Please take a look at the fixes in the connector's JIRA integration project on the [Eclipse bug list](#).

Below is a list of the fixes and improvements in the main project for the Atlassian Connector for Eclipse. The list below is drawn from our [JIRA issue tracker](#).

JIRA Issues (42 issues)			
Key	Summary	Priority	Status
PLE-567	[subversive] compare editor viewer order mixed up		 Closed
PLE-529	Eclipse Connector should not depend on Subversive and Subclipse		 Closed
PLE-560	SCM problems		 Closed
PLE-546	consume name change for connector		 Closed
PLE-537	FishEye Subversive integration connects to repository when opening popup menu		 Resolved
PLE-534	[e3.5] team.main contribution is invalid under e3.5		 Closed
PLE-526	NPE when opening FishEye link on CVS-managed project		 Closed
PLE-568	[subversive] create review from changeset fails		 Closed
PLE-564	help points to non-existing wiki page		 Closed
PLE-559	prepare better welcome screen messages		 Closed
PLE-556	Please review the release notes for the Eclipse Connector 1.1		 Closed

PLE-552	Documentation for Eclipse Connector 1.1		 Closed
PLE-549	Update error messages to include Subversive support too		 Resolved
PLE-547	Subclipse shouldn't be installed if I don't select it		 Closed
PLE-544	Add contributions to Welcome page		 Resolved
PLE-541	Add ability to create Crucible review for Subversive changeset		 Resolved
PLE-539	Add Eclipse Connector section to Welcome (Overview) page		 Closed
PLE-538	Auto populate SCM Path while configuring FishEye mapping for unmapped SCM		 Closed
PLE-536	cannot see reviewers in Crucible Editor		 Resolved
PLE-533	Bamboo feature doesn't show up in Eclipse 3.5 during initial installation (but it does show up after calculating requirements)		 Closed
PLE-532	Subversive support		 Closed
PLE-531	Handle null returns in TeamUiUtils and ITeamResourceConnector		 Closed
PLE-528	Empty implementations of CVS and Subversive Team Connectors break existing functionality		 Closed
PLE-527	Build PLE depending on a concrete ACC version		 Closed
PLE-524	Improved Subversive Support		 Closed
PLE-522	FishEye vs Subversive integration		 Closed
PLE-521	Open with FishEye for binary files should point to a FishEye history page (not raw file)		 Closed
PLE-519	Switch to Subclipse 1.6		 Closed
PLE-514	crucible validation does not work		 Closed
PLE-509	Support Open with FishEye for CVS projects		 Closed
PLE-507	Select SCM repository dialog comes up empty		 Resolved
PLE-503	"Get 10 more revisions" does not get 10 on Linux		 Closed
PLE-502	implement support extension		 Closed
PLE-473	[3.5] Issues with PLE and e3.5 Galileo		 Closed
PLE-465	When opening a non-Java file for a review and annotations cannot be displayed, show a popup to advise the user		 Closed
			

PLE-459	Provide ability to log work against JIRA task		Resolved
PLE-452	[Collection] JIRA Bugs for M4		 Closed
PLE-450	update build for Mylyn 3.2		 Closed
PLE-427	CrucibleHttpSessionCallback and BambooHttpSessionCallback look suprisingly similar (time to merge them?)		 Closed
PLE-305	[e3.5] annotation hovers are not displayed		 Closed
PLE-295	[e3.5] key-conflict error in log		 Closed
PLE-518	Add wizard banner to FishEye configuration mapping add/edit dialog		 Resolved

Atlassian Eclipse Connector 1.0 Release Notes

27 May 2009

With pleasure, Atlassian presents the **Atlassian Eclipse Connector 1.0**.

The Atlassian Eclipse Connector builds on [Mylyn's](#) task-focused interface to integrate your Eclipse IDE closely with your [JIRA](#) issues, [Crucible](#) code reviews, [FishEye](#) repository views and [Bamboo](#) builds. With the [earlier beta release](#) we announced the improved connector resulting from the collaboration between [Atlassian](#) and [Tasktop](#). Now we are delighted to present the production-ready 1.0 release.

What's new since the beta release?

- Use the new integration with [Atlassian FishEye](#) to open a source file in FishEye's web interface directly from Eclipse, or to copy a FishEye URL to your clipboard and share it with others.
- Conduct your [Crucible](#) code reviews when versioning your project under [SVN](#) with [Subversive](#) or under [CVS](#). Of course, the existing integration with [Subclipse](#) is also still available.
- Create a Crucible review from within Eclipse, for uncommitted code ('patch') or committed code ('changeset').
- Publish a draft review comment.
- Edit and remove existing comments.
- Assign reviewers without leaving Eclipse.
- Choose your favourite Bamboo builds as defined on the Bamboo server.

In the JIRA and Bamboo integrations we have focused on refactoring and bug-fixing, based on the results of testing the beta version.

Highlights of this Release:

- [FishEye Code Insight via Eclipse](#)
 - [Configure your FishEye Server as a Mylyn Task Repository](#)
 - [Open a File in FishEye directly from Eclipse](#)
- [Crucible Reviews in Eclipse](#)
 - [Manage your Code Reviews via Mylyn's Task-Focused Interface](#)
 - [Create a Crucible Review from within Eclipse](#)
 - [Conduct your Code Reviews in your IDE](#)
 - [See the Review Comments in Your Code](#)
 - [Work on the Review in the Diff View Too](#)
 - [Receive Notification of Review Updates](#)
- [Bamboo Builds in Eclipse](#)
 - [Configure your Bamboo Server as a Mylyn Task Repository](#)
 - [Manage your Bamboo Builds in your IDE](#)
 - [View the Build Logs](#)
 - [View the Changed Files](#)
 - [Monitor the Failed Tests](#)
 - [Receive Notification of Build changes](#)
- [JIRA Issues in Eclipse](#)
 - [Use the Awesome JIRA Mylyn Plugin as Part of the New Connector](#)
- [Complete List of Fixes in this Release](#)

Don't have the Atlassian Eclipse Connector yet?

You can install the connector directly from the Eclipse software updates manager. Follow our [installation guide](#).

This is an open source project. The source code is available in [our SVN repository](#).



Please log your [issues](#), [requests](#) and [votes](#). They help us decide what needs doing. It is easy to [add an issue](#). If you do not yet have an account, the system will prompt you to create one.

Upgrading from a previous version of the IDE Plugin for Eclipse

If you have previously installed version **0.4 or earlier** of the Atlassian IDE Plugin for Eclipse, please uninstall the earlier version. Then follow the [installation guide](#) to install the new Atlassian Eclipse Connector.

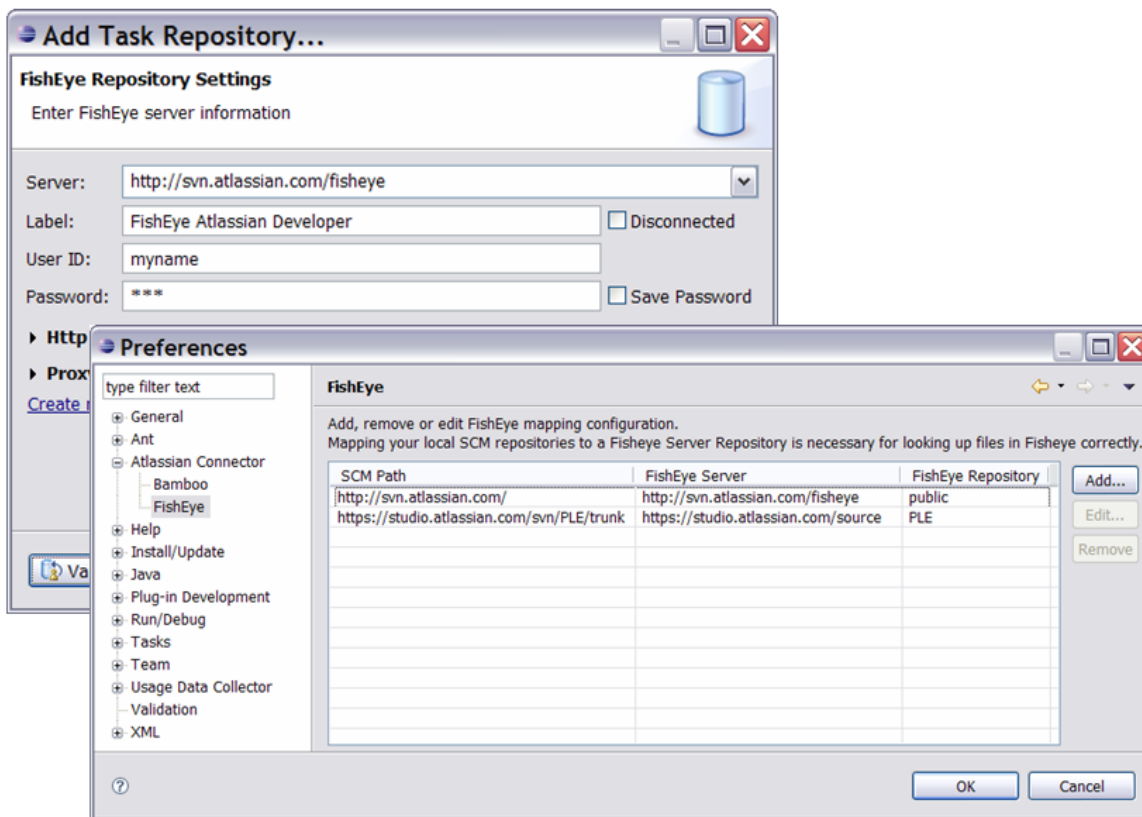
FishEye Code Insight via Eclipse

Even if you do not have Crucible, you can now hook up your [FishEye](#) server as an Eclipse task repository.

1

Configure your FishEye Server as a Mylyn Task Repository

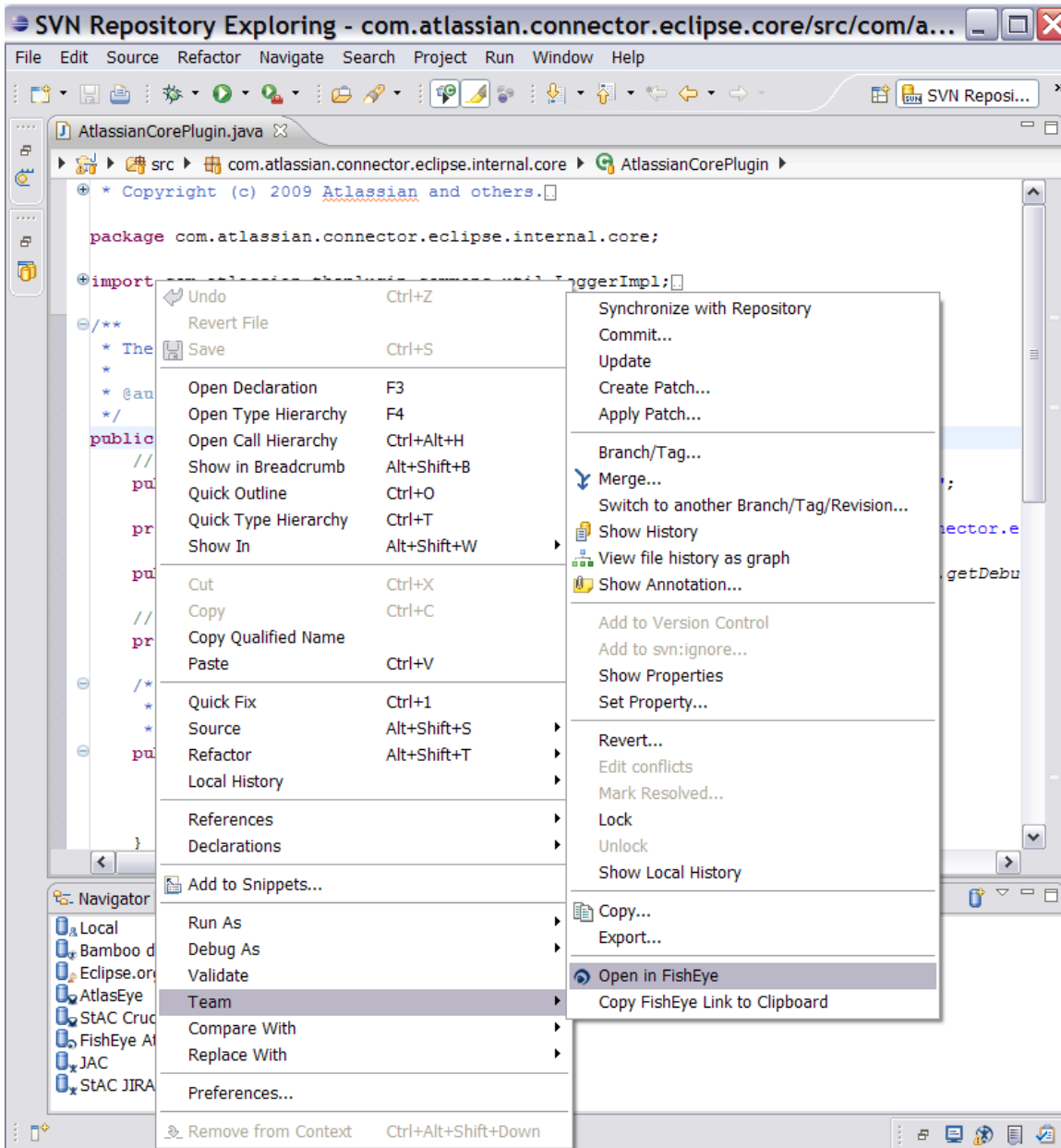
- Configure your FishEye server as a task repository.
- Map your source repository to your FishEye repository.



2

Open a File in FishEye directly from Eclipse

- Open a source file in FishEye's web interface, directly from your Eclipse editor.
- Copy the FishEye link to your clipboard and share FishEye file references with others.



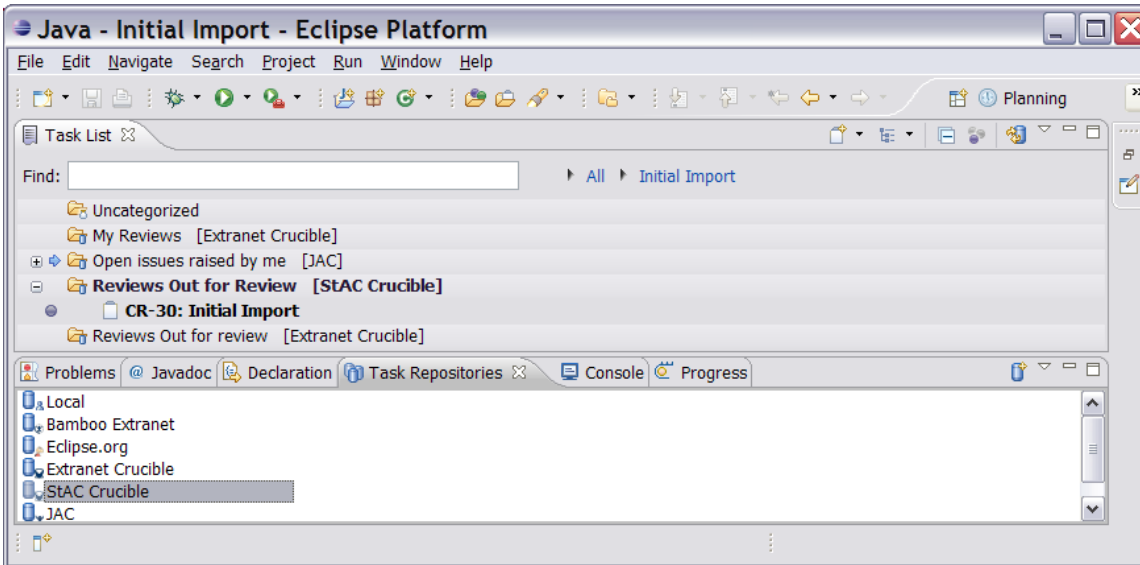
Crucible Reviews in Eclipse

Enjoy Crucible's lightweight workflow to conduct code reviews within your IDE. Using the connector, you can handle your [Crucible](#) reviews as Eclipse Mylyn tasks and take advantage of Mylyn's powerful context management features. In addition, you can perform much of the review workflow without leaving your IDE.



Manage your Code Reviews via Mylyn's Task-Focused Interface

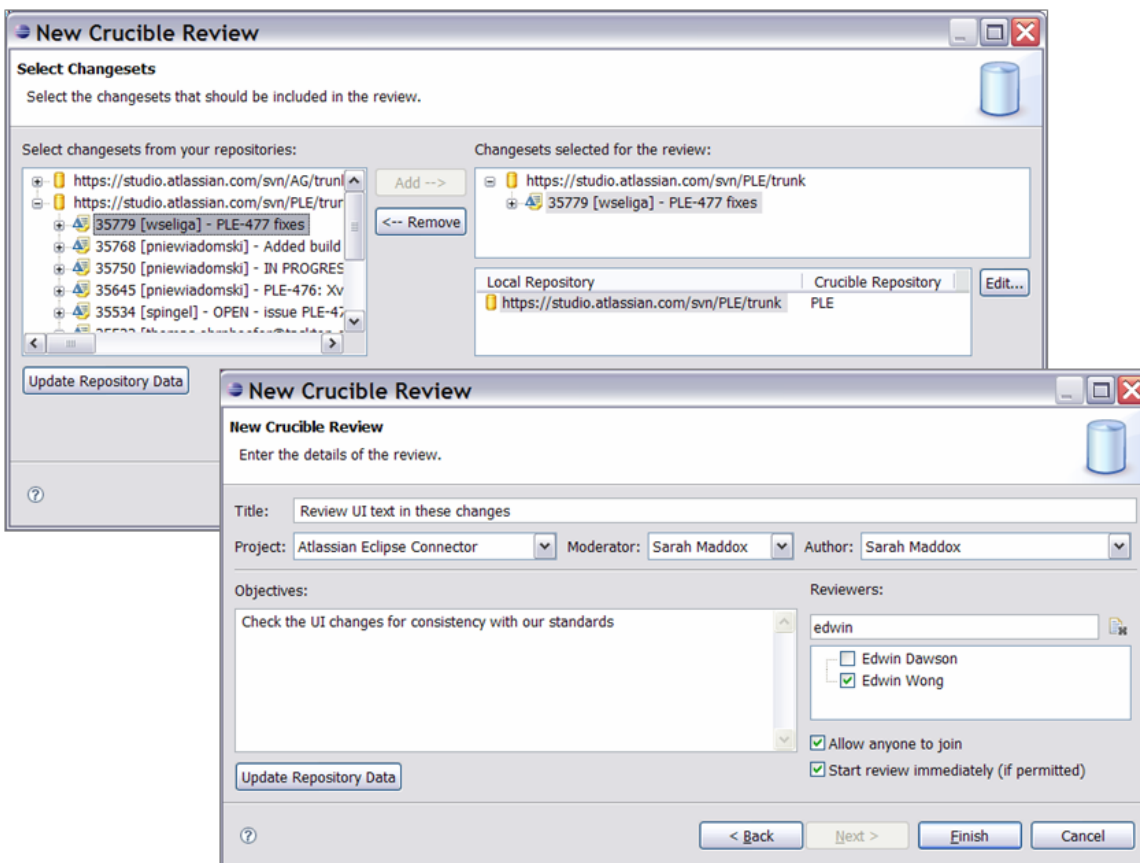
- Set up your Crucible server as a task repository.
- Build Mylyn queries to filter your reviews, based on the Crucible server filters or your own custom criteria.
- Manage the reviews in the Mylyn task list.
- Take advantage of Mylyn's context management to keep track of the files and other assets associated with your review.



4

Create a Crucible Review from within Eclipse

- Create a review for a committed changeset.
- Create a patch review for files not yet committed.
- Choose to start the review immediately or leave it in draft state.



Conduct your Code Reviews in your IDE

- See the review details, files and comments in an Eclipse view.
- Assign reviewers without leaving Eclipse.
- Create and reply to comments.
- Edit or remove comments.
- Open the source code file at the line that the comment refers to.
- Add changesets or a patch to an existing review.
- Summarise and close the review or perform other review workflow actions from within your IDE.

SVN Repository Exploring - Initial Import - Eclipse Platform

File Edit Navigate Search Project Run Window Help

CR-30: Initial Import CR-DEMO-2: Test review for technical writing

Review CR-30

Initial Import

State: Under Review Open Since: 16/07/2008 09:35

Participants

Author: Nicholas Muldoon

Moderator: Sarah Maddox

Reviewers: Andrew Lui

Statement of Objectives

14888: Initial Import

Review by technical writers

General Comments (1 comments)

Sarah Maddox 16/07/2008 09:41

Please add more javadoc elements.

Review Files (21 files, 1 comments)

trunk/IssueKeyComparator.java [Rev: N/A] Unknown Type

trunk/JIRAServer.java [Rev: N/A] Unknown Type

trunk/JIRAServerFacade.java [Rev: N/A] Unknown Type

trunk/JIRAServerFacadeImpl.java [Rev: N/A] Unknown Type

trunk/JIRAServerFactory.java [Rev: N/A] Unknown Type

trunk/api [Rev: N/A] Unknown Type

trunk/api/AbstractJIRAConstantBean.java [Rev: N/A] Unknown Type

trunk/api/JIRAComponent.java [Rev: N/A] Unknown Type

trunk/api/JIRAConstant.java [Rev: N/A] Unknown Type

trunk/api/JIRAException.java [Rev: N/A] Unknown Type

trunk/api/JIRAIssue.java [Rev: N/A] Unknown Type

trunk/api/JIRAIssueBean.java [Rev: N/A] Unknown Type

trunk/api/JIRAIssueTypeBean.java [Rev: N/A] Unknown Type

trunk/api/JIRALoginException.java [Rev: N/A] Unknown Type

trunk/api/JIRAProject.java [Rev: N/A] Unknown Type

Sarah Maddox 16/07/2008 09:42 [Line: 3]

Please add sjort javadoc summary of purpose of this interface.

Select Reviewer(s)

edwin

Edwin Dawson

Edwin Wong

OK Cancel

Console Bamboo Task Repositories Task List

Find: 30

Search repository for key or summary...

All my StAC reviews [StAC Crucible]

CR-30: Initial Import

All open Atlassian Gadgets issues [StAC JIRA]

All open StAC PLE issues [StAC JIRA]

6

See the Review Comments in Your Code

- See the comments annotated in your file editor.
- Add comments directly in the editor.

The screenshot shows a code editor window titled "BambooView.java 29892". The code is in Java and includes a diff view overlay. The diff view shows changes to the `showJUnitView` method. The changes are highlighted in yellow. The diff view also includes comments from Thomas Ehrnhoefer and Wojtek Seliga.

```
@Override
public void done(IJobChangeEvent event) {
    if (event.getResult() == Status.OK_STATUS) {
        File testResults = ((RetrieveTestResultsJob) event.getJob
        if (testResults != null) {
            showJUnitView(testResults, build.getBuildKey() + "-" + build.getId());
        }
    }
}

private void showJUnitView(final File testResults, final String buildKey) {
    getSite().getShell().getDisplay().asyncExec(new Runnable() {
        @SuppressWarnings("restriction")
        public void run() {
            if (!getSite().getShell().isDisposed()) {
                try {
                    getViewSite().getPage().showView(TestRunnerViewPart.NAME);
                    final IJavaProject[] javaProjectsTmp = JavaModelManager.getJavaModelManager().getJavaModel().getJavaProjects();
                    final Collection<IJavaProject> javaProjects = Arrays.asList(javaProjectsTmp);
                    final CompositeJavaProject compositeProject = new CompositeJavaProject(javaProjects);
                    final IJavaProject[] javaProjectsTmp = JavaCore.create(ResourcesPlugin.getWorkspace().getRoot()).getJavaProjects();
                    final TestRunSession trs = new TestRunSession("Bamboo build " + buildKey, trs.getLaunchConfiguration().getName(), name);
                } catch (Exception e) {
                    // ERROR, Bamboo build failed
                }
            }
        }
    });
}
```

▼ Thomas Ehrnhoefer Mar 9, 2009 6:10 PM [Lines: 472 - 471] [Open Review](#)

This seems to be unused?

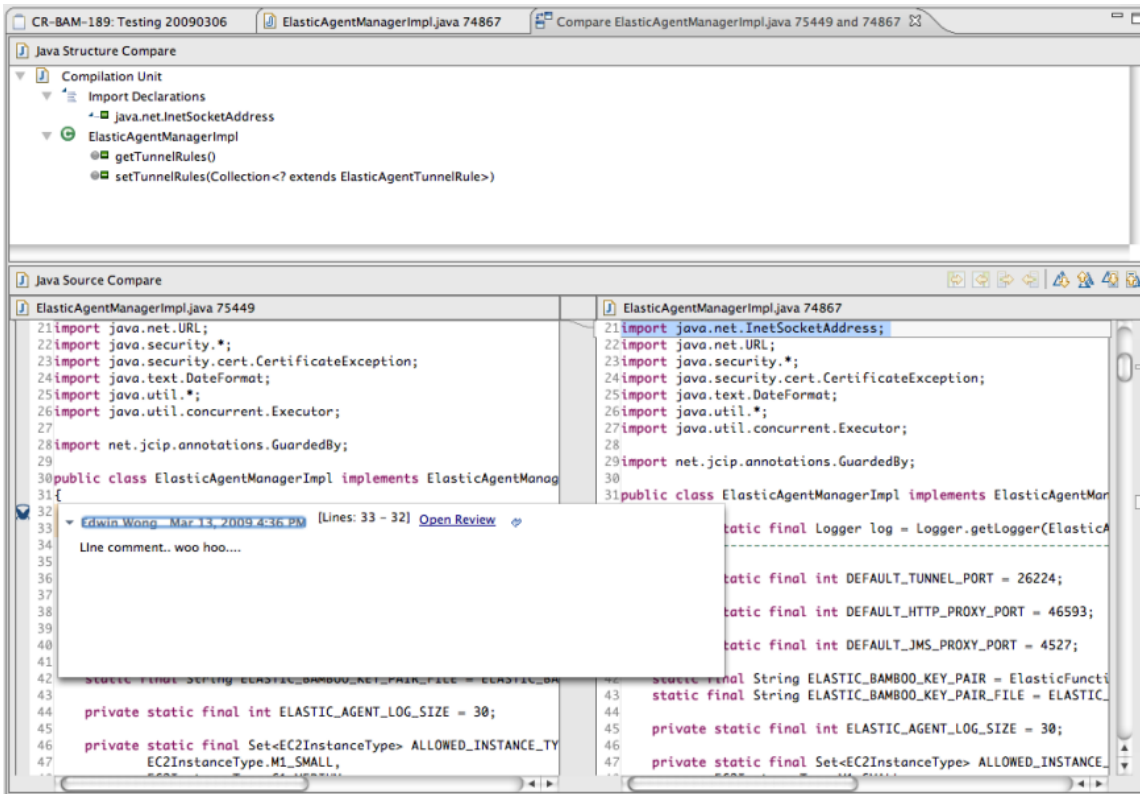
▼ Wojtek Seliga Mar 10, 2009 2:20 AM

yes. Forgot to remove it after recent refactoring and code cleanup. I will remove it

7

Work on the Review in the Diff View Too

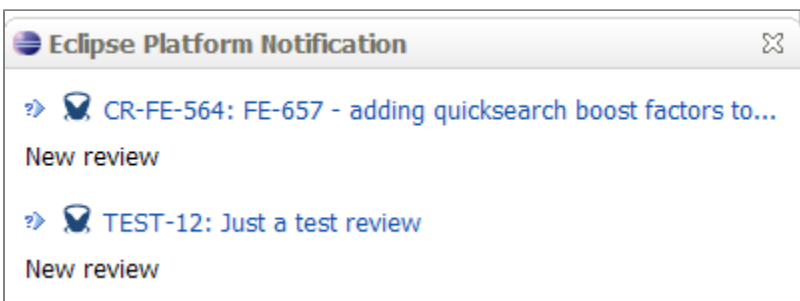
- Open a diff view to see the changes under review, with the comments highlighted.
- Add comments directly in the diff view too.



8

Receive Notification of Review Updates

- Receive notification of new reviews needing your attention.
- See what's happening in your reviews, via the Atlassian Eclipse Connector's integration with the standard Eclipse notification messages.



Bamboo Builds in Eclipse

Using the Atlassian Eclipse Connector, you can set up your [Bamboo](#) server as a Mylyn task repository. Now you can choose the builds you want to monitor and view the build results, build logs, changed files and failed tests inside your IDE.

9

Configure your Bamboo Server as a Mylyn Task Repository

- Configure your Bamboo server as a task repository.
- Choose the builds you want to monitor from within Eclipse.

Properties for Task Repository

Bamboo Repository Settings
Enter Bamboo server information

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

► **Http Authentication**

► **Proxy Server Configuration**

▼ **Build Plans**

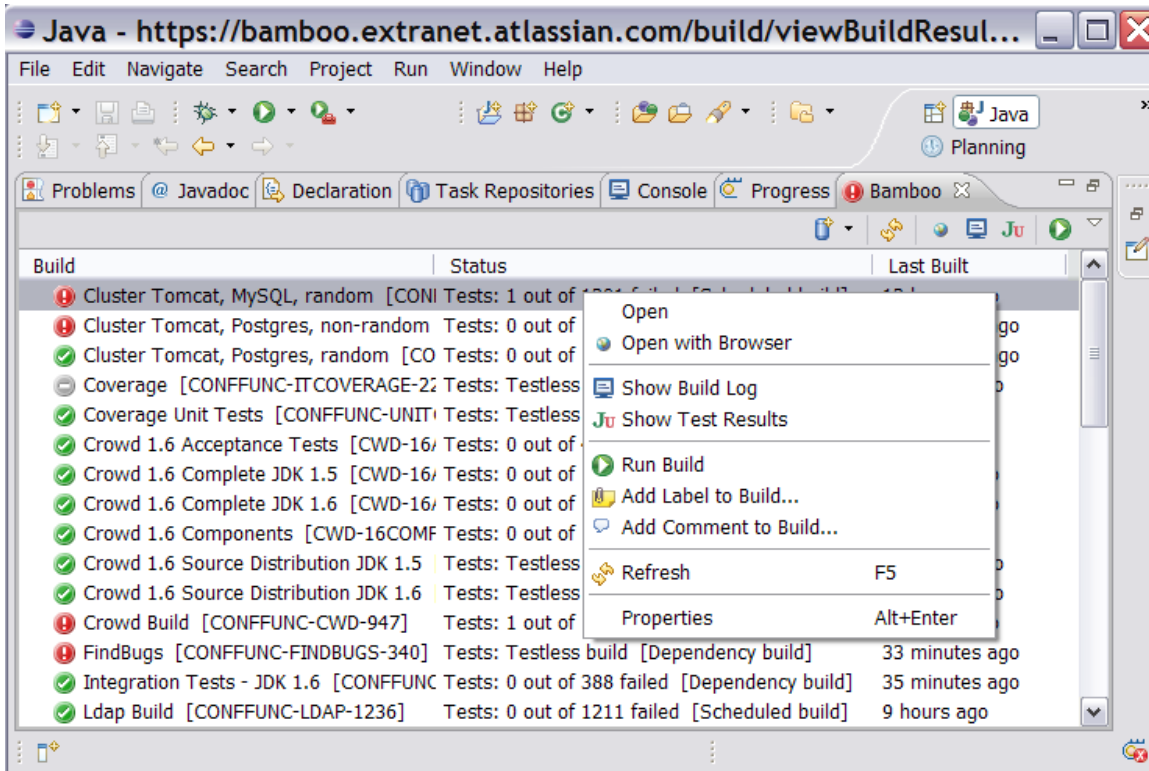
<input type="checkbox"/>	Atlassian Activity Stream Plugin - trunk - jdk 1.5 (STRM-DEF)
<input type="checkbox"/>	Atlassian AppLinks - Trunk (APL-TRUNK)
<input type="checkbox"/>	Atlassian Cache - trunk - jdk 1.5 (CACHE-TRUNKJDK15)
<input type="checkbox"/>	Atlassian Cache - trunk - jdk 1.6 (CACHE-TRUNKJDK16)
<input checked="" type="checkbox"/>	Atlassian Gadgets - 0.x Branch (AG-STABLE)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Atlassian UI - Dashboard Branch (AG-AUI)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Dashboard Hackathon branch (AG-DASHHACK)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Nightly Deployment (AG-DEPLOY)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Summit Only Branch (AG-SUMMITONLY)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Trunk - JDK 1.5 (AG-TRUNK)
<input checked="" type="checkbox"/>	Atlassian Gadgets - Trunk - JDK 1.6 (AG-TRUNKJDK16)

[Create new account](#) [Change account settings](#)

10

Manage your Bamboo Builds in your IDE

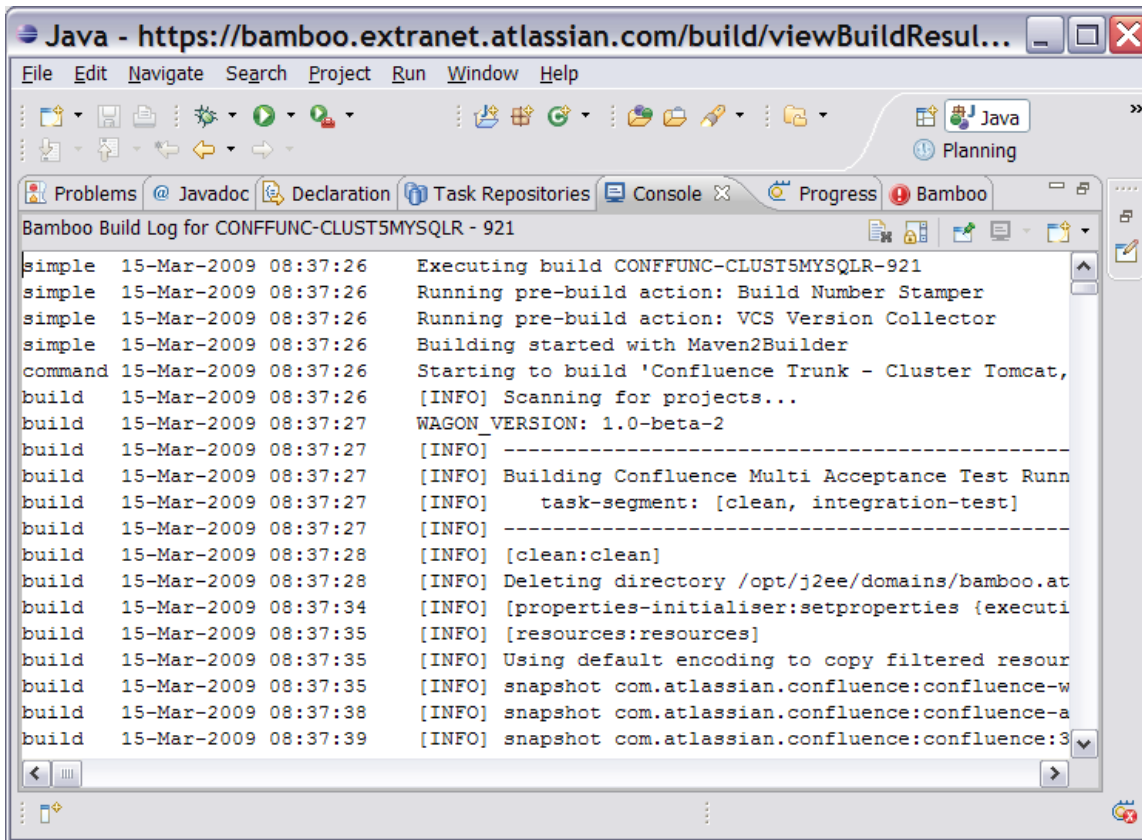
- View the status of your selected builds.
- Right-click a build line to act upon it.
- Re-run the build.
- Add a label or comment to the build.
- Double-click the build to open it in an Eclipse view.



11

View the Build Logs

- Open the Bamboo build log in your Eclipse Console view.



12

View the Changed Files

- See the changes that triggered the build.

ECL-TST-375

Build ECL-TST-375

Project: Atlassian Eclipse Connector Plan: Compile and Test Build: 375

Summary

Build ECL-TST-375 failed.

Build completed on Wed Mar 18 11:05:02 PDT 2009

Build took 18 minutes

Build reason: Manual build

Tests

Tests in total: 69 Failed: 3

Failed Tests:

VersionedCommentDateComparatorTest : Compare
CrucibleUiUtilTest : Get Crucible Task From Id
CrucibleUiUtilTest : Get Crucible Task Repository From String

[Show Test Results](#)

Code Changes

- [sminto] IN PROGRESS - issue PLE-77: add license to features
- [sminto] OPEN - issue PLE-318: Immediately show own comments in Rich Editor
 - /PLE/trunk/com.atlassian.connector.eclipse.crucible.tests/src/com/atlassian/connector/eclipse/internal/crucible/ui/Crucib

Build Log

The build generated 143 lines (8 error lines). See the [full build log](#) for details.

```

BUILD FAILED
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat

Total time: 1 minute 32 seconds

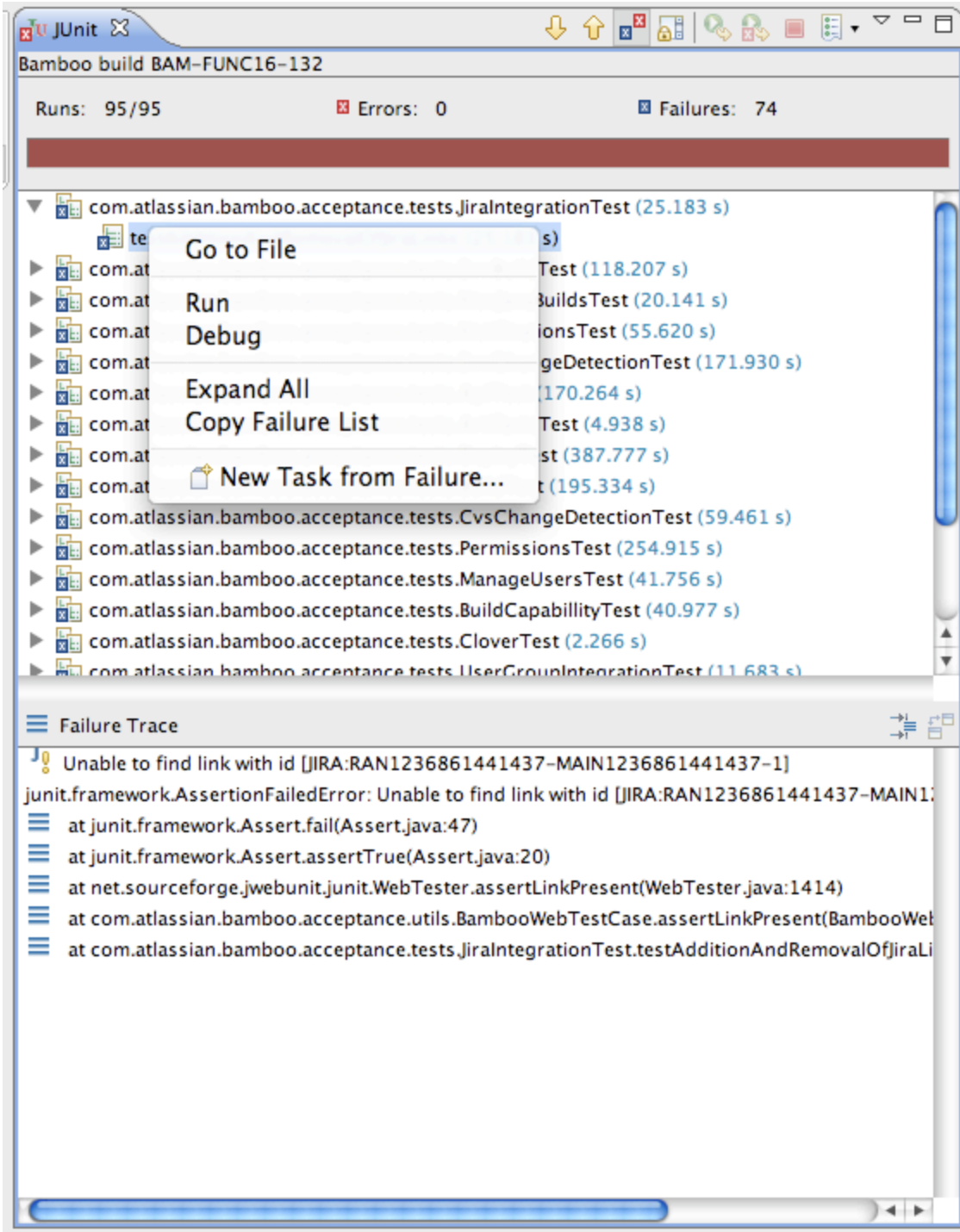
```

Bamboo

13

Monitor the Failed Tests

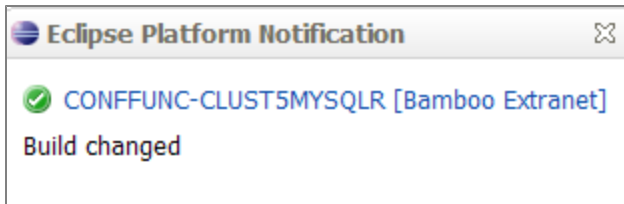
- See the test results for the selected build.
- Re-run the failed tests locally, to make sure your fixes work before breaking the build again.
- Create a new Mylyn task from the failure. This Mylyn task could be a JIRA issue.



14

Receive Notification of Build changes

- Receive build notifications via the Atlassian Eclipse Connector's integration with the standard Eclipse notification messages.



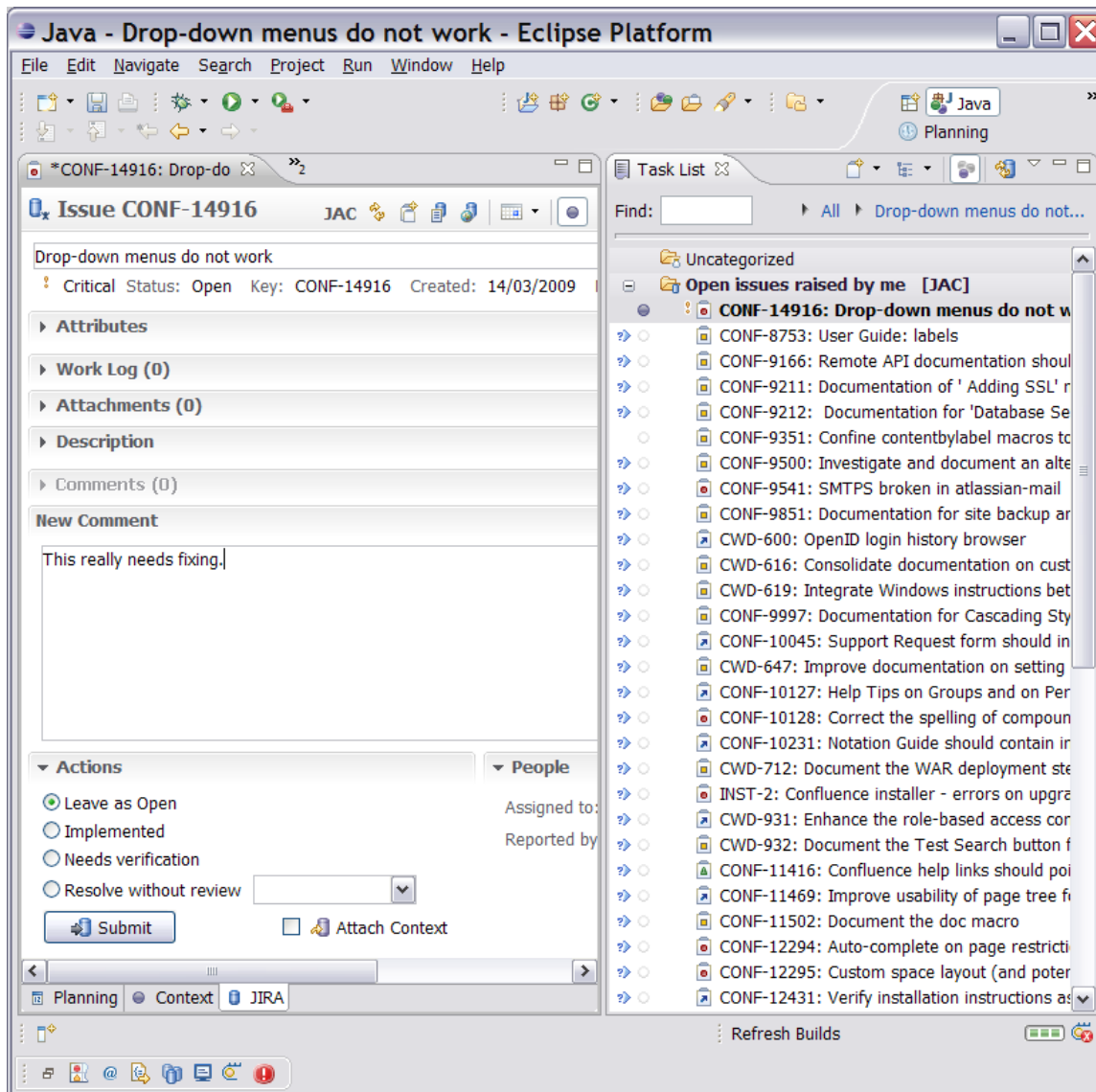
JIRA Issues in Eclipse

The new connector incorporates the [JIRA Mylyn plugin](#), which was previously available as a JIRA-only connector. Configure your [JIRA](#) server as a task repository. Use Mylyn's context management to keep track of the files and other assets associated with your issue.




















































Use the Awesome JIRA Mylyn Plugin as Part of the New Connector














































- Configure your [JIRA](#) server as a task repository.
- Build Mylyn queries to filter your issues, based on your JIRA server filters or on custom criteria.
- Manage the issues via the Mylyn task list.
- Take advantage of Mylyn's context management to keep track of the files and other assets associated with your issue.
- Create a new JIRA issue and edit existing issues.
- Add and reply to comments.
- Receive notification of updated issues.



Complete List of Fixes in this Release

JIRA Issues (54 issues)				
Key	Summary	Priority	Status	
PLE-477	Bamboo core/test not compiling due to commons changes		Closed	
PLE-466	remove beta tags from UI		Closed	
PLE-462	opening crucible editor blocks UI		Closed	
PLE-460	Copying something to clipboard in the last screen of the review creation wizard overwrote my patch		Closed	
PLE-505	Release Engineering		Closed	
PLE-499	Pre 1.0 nits		Closed	
PLE-489	Add support for Crucible servers being also FishEye servers		Closed	

PLE-454	change presentation of "reviewFiles" that originate from a "review from patch"		 Closed
PLE-437	test JIRA connector 3.1.1 release		 Closed
PLE-498	New review Wizard: minor nits		 Closed
PLE-496	Recent change (ACC-31) seems to have disabled validating a connection with a Crucible server behind http authentication		 Closed
PLE-495	Sort projects in Create Review wizard (project drop-down/combo)		 Closed
PLE-486	installation fails with unsatisfied dependency error		 Closed
PLE-484	Add Atlassian Preference (About) Page		 Closed
PLE-482	Create Bamboo preference page		 Closed
PLE-481	Please review Eclipse Connector 1.0 Configuration Guide		 Closed
PLE-480	Fix spelling of "Favorites" on Bamboo repository settings screen		 Closed
PLE-479	Please review Eclipse Connector 1.0 installation guide		 Closed
PLE-478	Please review Eclipse Connector 1.0 release notes		 Closed
PLE-475	Smaller Icons for review editor		 Closed
PLE-469	Fix compilation problems after recent refactoring made to ACC (Comment.getReplies)		 Closed
PLE-464	Nits discussed in the M3.5 meeting		 Closed
PLE-456	Incorporate recent changes around ServerCfgy -> ServerData in Atlassian Connector Commons		 Closed
PLE-453	Fix reflow / layout problems in CrucibleEditor		 Closed
PLE-449	change nightly builds to point to the Mylyn release update site		 Closed
PLE-447	Referenced identifier 'com.atlassian.connector.eclipse.crucible.ui.comment.annotation' in attribute 'annotationType' cannot be found		 Closed
PLE-443	Add option to immediately start review while creating it from Eclipse		 Closed
PLE-422	File lookup of Subclipse connector does not work for our internal (subversive) repository		 Closed
PLE-420	Hide inapplicable actions related to adding comments on a code review		 Resolved
PLE-412	Provide Wizard for creating Review		 Resolved
PLE-410	opening Crucible editor leaves focus on Task List		 Closed
			

PLE-400	Review editor does not get correctly updated when a user removes a reply to a comment		Closed
PLE-399	Create action for saving/submitting changes made in the ReviewEditor		 Resolved
PLE-398	Copy to clipboard a link to Fisheye from a file in Eclipse		 Closed
PLE-397	Click through to Fisheye Browser from a file in Eclipse		 Closed
PLE-395	The connector should support Subversive SVN Team provider		 Closed
PLE-390	Refactor Popup Dialogs and Error Handling in DefaultTeamProvider / TeamUiUtils		 Closed
PLE-389	editor location jumps on refresh		 Closed
PLE-388	Refactor Bamboo Actions		 Closed
PLE-386	refactor AddGeneralCommentToFileAction		 Closed
PLE-381	Clean up Exception handling in opening JUnit Action		 Closed
PLE-380	user access rights are not respected while offering them add general file comment functionality		 Closed
PLE-371	Add API nature to atlassian projects		 Closed
PLE-368	Could not open JUnit Test Results		 Closed
PLE-316	Post Drafts		 Resolved
PLE-315	Edit/Delete own comments (including drafts)		 Resolved
PLE-314	Edit Comments		 Resolved
PLE-299	Action: create Task from failed Build		 Closed
PLE-250	create a basic rich editor for new reviews		 Resolved
PLE-245	provide a listing of all of the changesets to be added to the new review in the rich editor		 Resolved
PLE-239	provide a way of adding reviewers to the review in the new review rich editor		 Resolved
PLE-238	add an action to a changeset in the subclipse history view		 Resolved
PLE-173	Create Crucible review with patch (uncommitted code)		 Resolved
PLE-172	Create Crucible review with committed files		 Resolved

Atlassian Eclipse Connector 1.0 Beta Release Notes

23 March 2009

With pleasure, Atlassian presents the **Atlassian Eclipse Connector 1.0 Beta**.

[Atlassian](#) and [Tasktop](#) have collaborated to build a new connector with a radically changed look and feel and a wealth of new features. The resulting connector makes use of [Mylyn's](#) task-focused interface to bring new level of integration between your Eclipse Mylyn IDE, your [Crucible](#) code reviews and your [Bamboo](#) builds. The new connector also includes the existing [JIRA Mylyn plugin](#).

With the Atlassian Eclipse Connector, you can set up your JIRA, Bamboo and Crucible servers as Eclipse Mylyn task repositories. Monitor your Bamboo builds and re-run failed tests locally. Enjoy Crucible's lightweight workflow to conduct code reviews within your IDE. The JIRA issues and Crucible reviews are seen as tasks, allowing you to make use of Mylyn's powerful task and context management features. Instead of dropping out to view your email, you will now receive your issue, review and build notifications via the standard Eclipse notification popups.

Highlights of this Release:

- [Crucible Reviews in Eclipse](#)
 - [Manage your Code Reviews via Mylyn's Task-Focused Interface](#)
 - [Conduct your Code Reviews in your IDE](#)
 - [See the Review Comments in Your Code](#)
 - [Work on the Review in the Diff View Too](#)
 - [Receive Notification of Review Updates](#)
- [Bamboo Builds in Eclipse](#)
 - [Configure your Bamboo Server as a Mylyn Task Repository](#)
 - [Manage your Bamboo Builds in your IDE](#)
 - [View the Build Logs](#)
 - [View the Changed Files](#)
 - [Monitor the Failed Tests](#)
 - [Receive Notification of Build changes](#)
- [JIRA Issues in Eclipse](#)
 - [Use the Awesome JIRA Mylyn Plugin as Part of the New Connector](#)

Don't have the Atlassian Eclipse Connector yet?

You can install the connector directly from the 'Software Updates' option in Eclipse. Follow our easy [installation guide](#).

This is an open source project. The source code is available in [our SVN repository](#).



Please log your [issues](#), [requests](#) and [votes](#). They help us decide what needs doing. It is easy to [create an issue](#). (If you do not yet have an account, the system will prompt you to create one.)

Upgrading from a previous version of the IDE Plugin for Eclipse

If you have previously installed an earlier version of the **Atlassian IDE Plugin for Eclipse**, please uninstall the earlier version. Then follow the [installation guide](#) to install the new **Atlassian Eclipse Connector**.

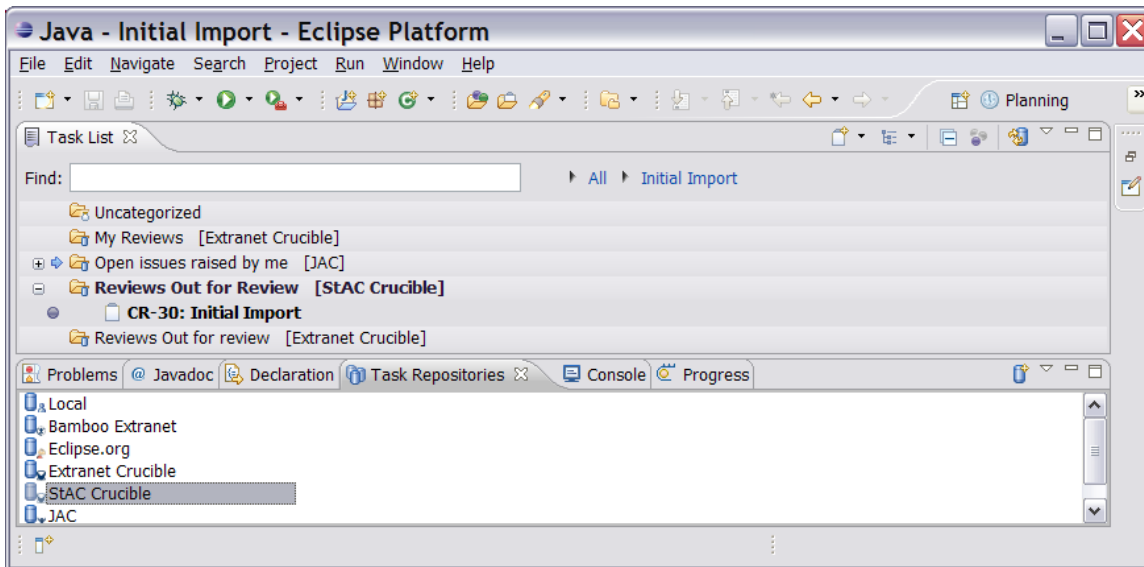
Crucible Reviews in Eclipse

Enjoy Crucible's lightweight workflow to conduct code reviews within your IDE. Using the connector, you can handle your [Crucible](#) reviews as Eclipse Mylyn tasks and take advantage of Mylyn's powerful context management features. In addition, you can perform much of the review workflow without leaving your IDE.



Manage your Code Reviews via Mylyn's Task-Focused Interface

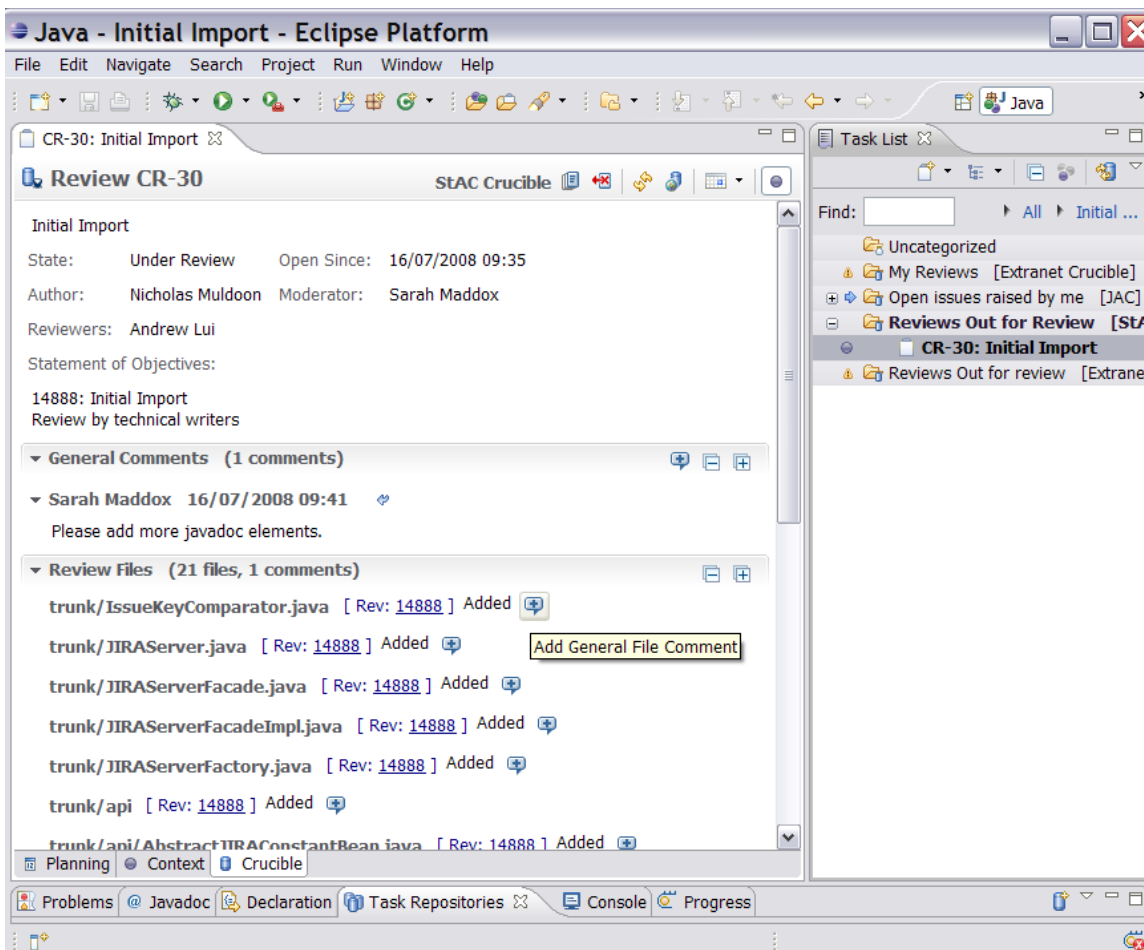
- Set up your Crucible server as a task repository.
- Build Mylyn queries to filter your reviews, based on the Crucible server filters or your own custom criteria.
- Manage the reviews in the Mylyn task list.
- Take advantage of Mylyn's context management to keep track of the files and other assets associated with your review.



2

Conduct your Code Reviews in your IDE

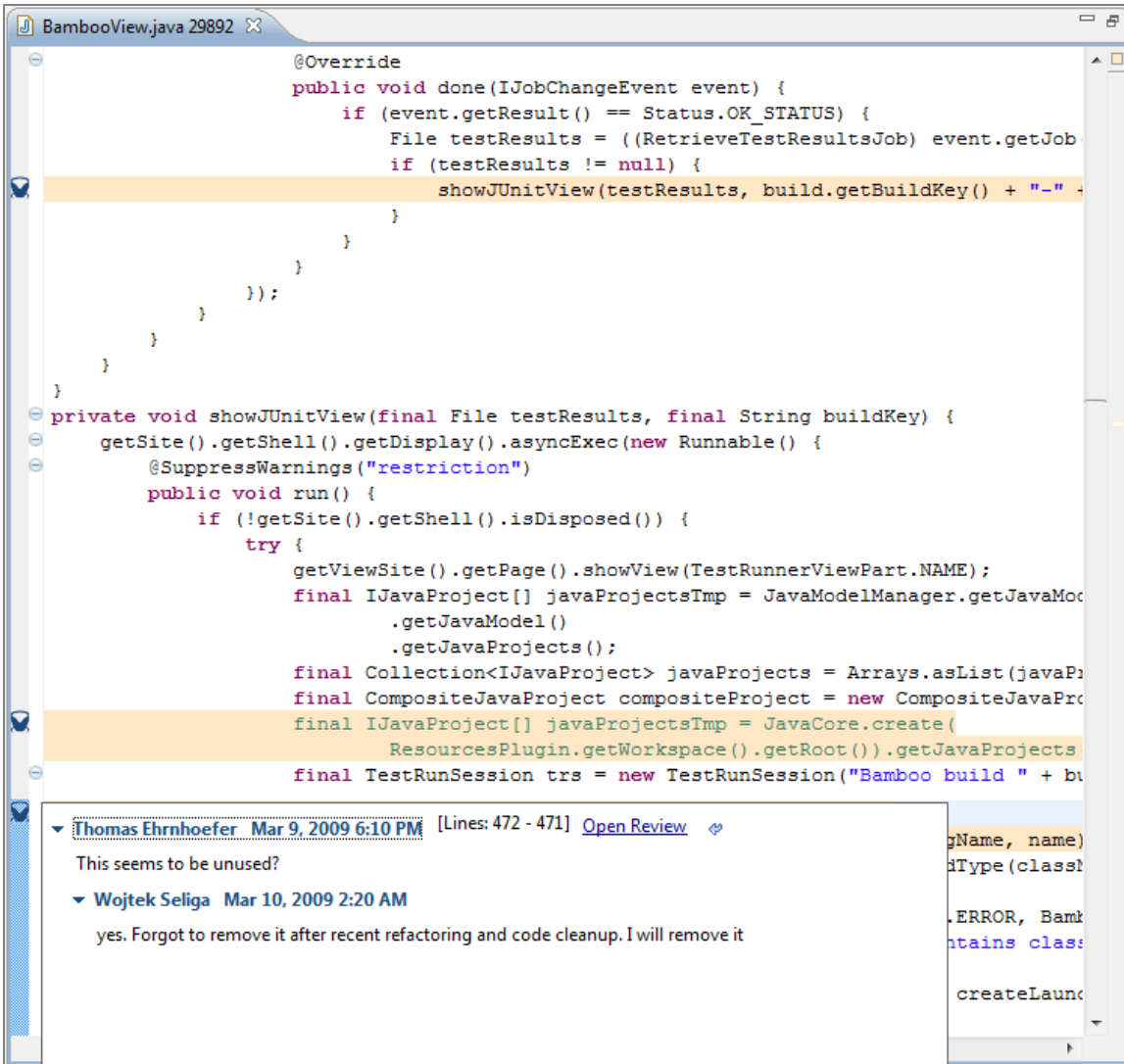
- See the review details, files and comments in an Eclipse view.
- Create and reply to comments.
- Summarise and close the review or perform other review workflow actions from within your IDE.



3

See the Review Comments in Your Code

- See the comments annotated in your file editor.
- Add comments directly in the editor.



The screenshot shows an IDE window titled "BambooView.java 29892". The code is a Java class with a method `done(IJobChangeEvent event)` that calls `showJUnitView`. A comment from Thomas Ehrnhoefer (Mar 9, 2009 6:10 PM) is visible at the bottom, stating "This seems to be unused?". Another comment from Wojtek Seliga (Mar 10, 2009 2:20 AM) responds, "yes. Forgot to remove it after recent refactoring and code cleanup. I will remove it".

```

@Override
public void done(IJobChangeEvent event) {
    if (event.getResult() == Status.OK_STATUS) {
        File testResults = ((RetrieveTestResultsJob) event.getJob()
            .getTestResults()).getTestResults();
        if (testResults != null) {
            showJUnitView(testResults, build.getBuildKey() + "-" + build.getId());
        }
    }
}

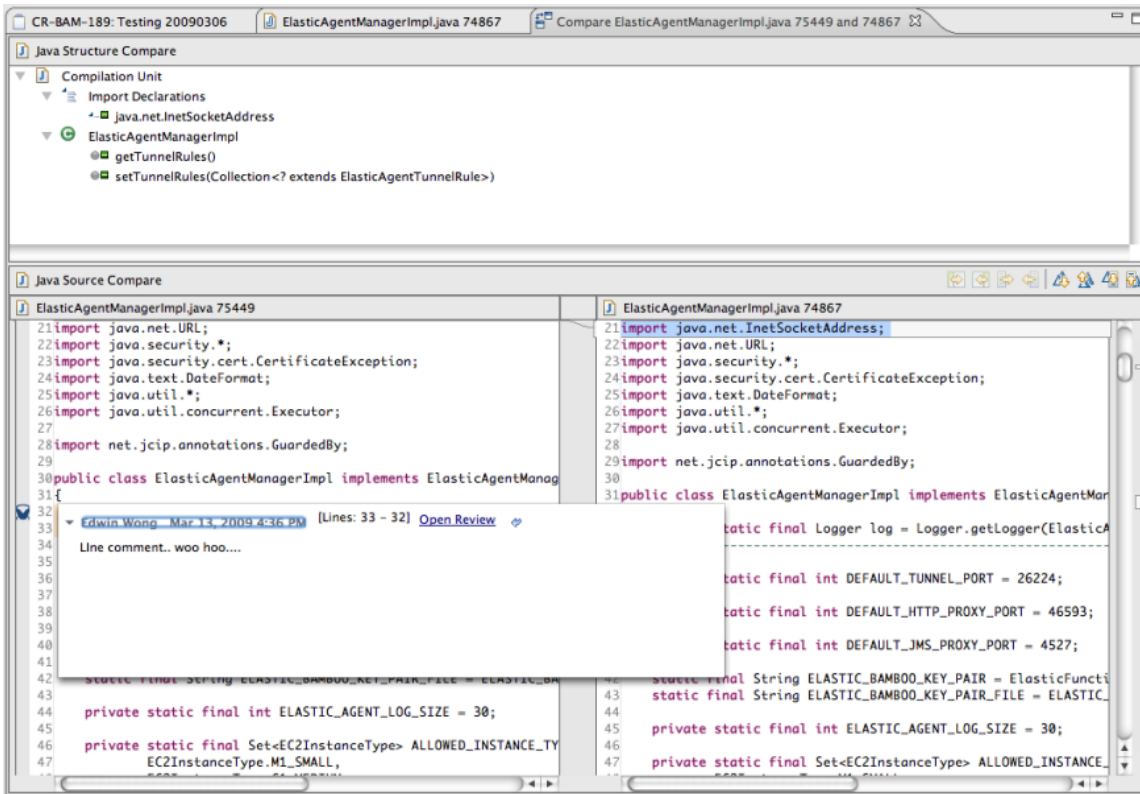
private void showJUnitView(final File testResults, final String buildKey) {
    getSite().getShell().getDisplay().asyncExec(new Runnable() {
        @SuppressWarnings("restriction")
        public void run() {
            if (!getSite().getShell().isDisposed()) {
                try {
                    getViewSite().getPage().showView(TestRunnerViewPart.NAME);
                    final IJavaProject[] javaProjectsTmp = JavaModelManager.getJavaModelManager()
                        .getJavaModel().getJavaProjects();
                    final Collection<IJavaProject> javaProjects = Arrays.asList(javaProjectsTmp);
                    final CompositeJavaProject compositeProject = new CompositeJavaProject(javaProjects);
                    final IJavaProject[] javaProjectsTmp = JavaCore.create(
                        ResourcesPlugin.getWorkspace().getRoot().getJavaProjects().toArray(new IJavaProject[0]));
                    final TestRunSession trs = new TestRunSession("Bamboo build " + buildKey,

```

4

Work on the Review in the Diff View Too

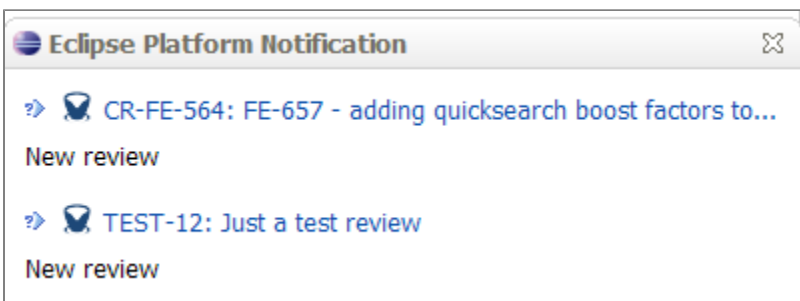
- Open a diff view to see the changes under review, with the comments highlighted.
- Add comments directly in the diff view too.



5

Receive Notification of Review Updates

- Receive notification of new reviews needing your attention.
- See what's happening in your reviews, via the Atlassian Eclipse Connector's integration with the standard Eclipse notification messages.



Bamboo Builds in Eclipse

Using the Atlassian Eclipse Connector, you can set up your [Bamboo](#) server as a Mylyn task repository. Now you can choose the builds you want to monitor and view the build results, build logs, changed files and failed tests inside your IDE.

6

Configure your Bamboo Server as a Mylyn Task Repository

- Configure your Bamboo server as a task repository.
- Choose the builds you want to monitor from within Eclipse.

Properties for Task Repository

Bamboo Repository Settings

Enter Bamboo server information

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

► **Http Authentication**

► **Proxy Server Configuration**

▼ **Build Plans**

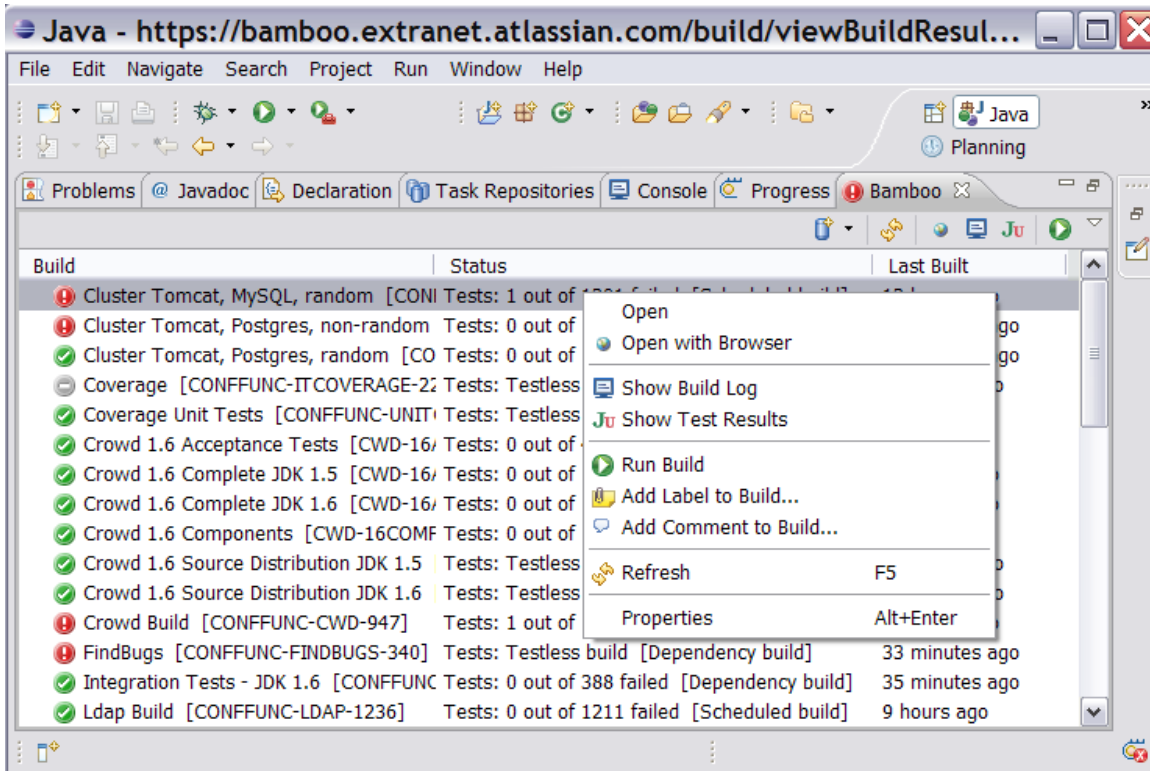
- ☒ Atlassian Javascript - Selenium, IE7 (AJS-SELENIUMIE7)
- ☒ Atlassian Javascript - Trunk (AJS-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.5 (LICENSE-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.6 (LICENSE-TRUNKJDK16)
- ☐ Clover + Sonar - Sonar - Bamboo - Unit Test (QA-BAM)
- ☐ Clover + Sonar - Sonar - Clover (QA-CLOVER)
- ☐ Clover + Sonar - Sonar - Common - Bamboo Maven Plugin (QA-BAM)
- ☐ Clover + Sonar - Sonar - Common - Bandana (QA-BANDANA)
- ☐ Clover + Sonar - Sonar - Common - Base Archetype (QA-BARCHTYPE)
- ☐ Clover + Sonar - Sonar - Common - Bonnie (QA-BONNIE)
- ☐ Clover + Sonar - Sonar - Common - Bucket (QA-BUCKET)

[Create new account](#) [Change account settings](#)

7

Manage your Bamboo Builds in your IDE

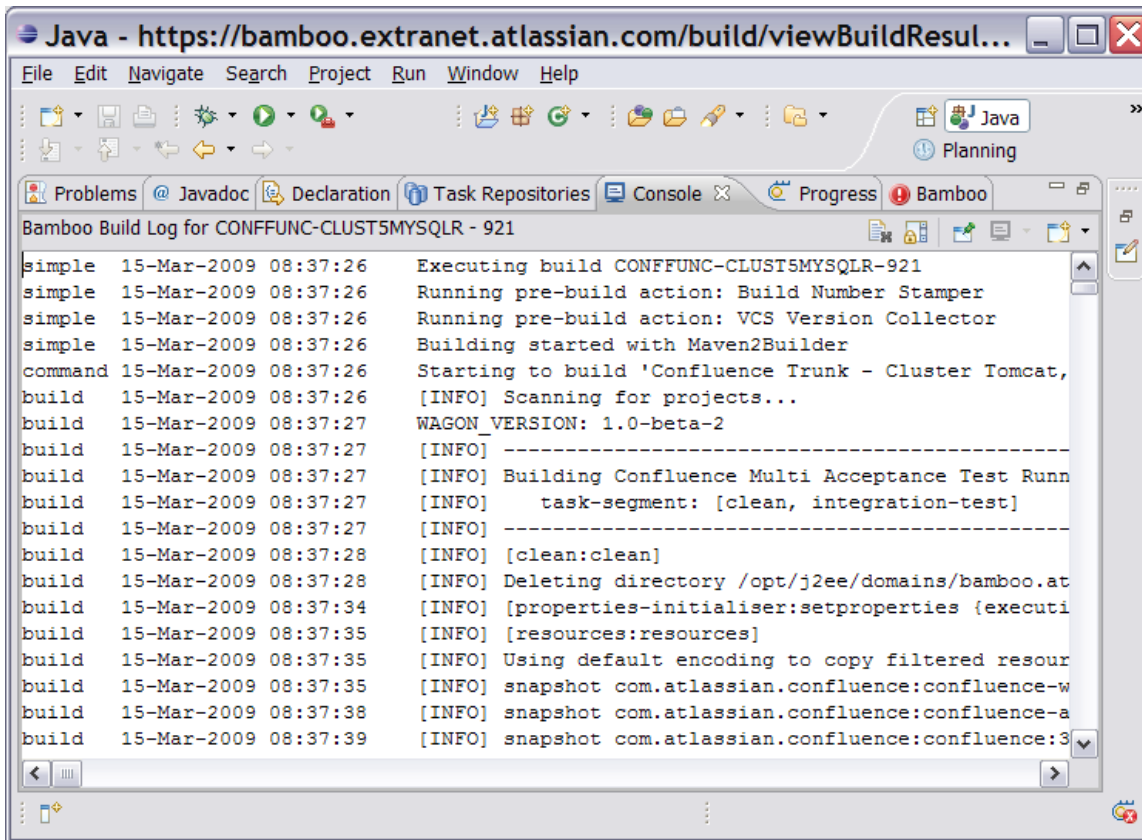
- View the status of your selected builds.
- Right-click a build line to act upon it.
- Re-run the build.
- Add a label or comment to the build.
- Double-click the build to open it in an Eclipse view.



8

View the Build Logs

- Open the Bamboo build log in your Eclipse Console view.



9

View the Changed Files

- See the changes that triggered the build.

ECL-TST-375

Build ECL-TST-375

Project: Atlassian Eclipse Connector Plan: Compile and Test Build: 375

Summary

Build ECL-TST-375 failed.

Build completed on Wed Mar 18 11:05:02 PDT 2009

Build took 18 minutes

Build reason: Manual build

Tests

Tests in total: 69 Failed: 3

Failed Tests:

VersionedCommentDateComparatorTest : Compare
CrucibleUiUtilTest : Get Crucible Task From Id
CrucibleUiUtilTest : Get Crucible Task Repository From String

[Show Test Results](#)

Code Changes

- [sminto] IN PROGRESS - issue PLE-77: add license to features
- [sminto] OPEN - issue PLE-318: Immediately show own comments in Rich Editor
 - /PLE/trunk/com.atlassian.connector.eclipse.crucible.tests/src/com/atlassian/connector/eclipse/internal/crucible/ui/Crucib

Build Log

The build generated 143 lines (8 error lines). See the [full build log](#) for details.

```

BUILD FAILED
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat
/opt/j2ee/domains/bamboo2.atlassian.com/gdansk/webapps/data/bamboo-2.0.2-attempt/xml-dat

Total time: 1 minute 32 seconds

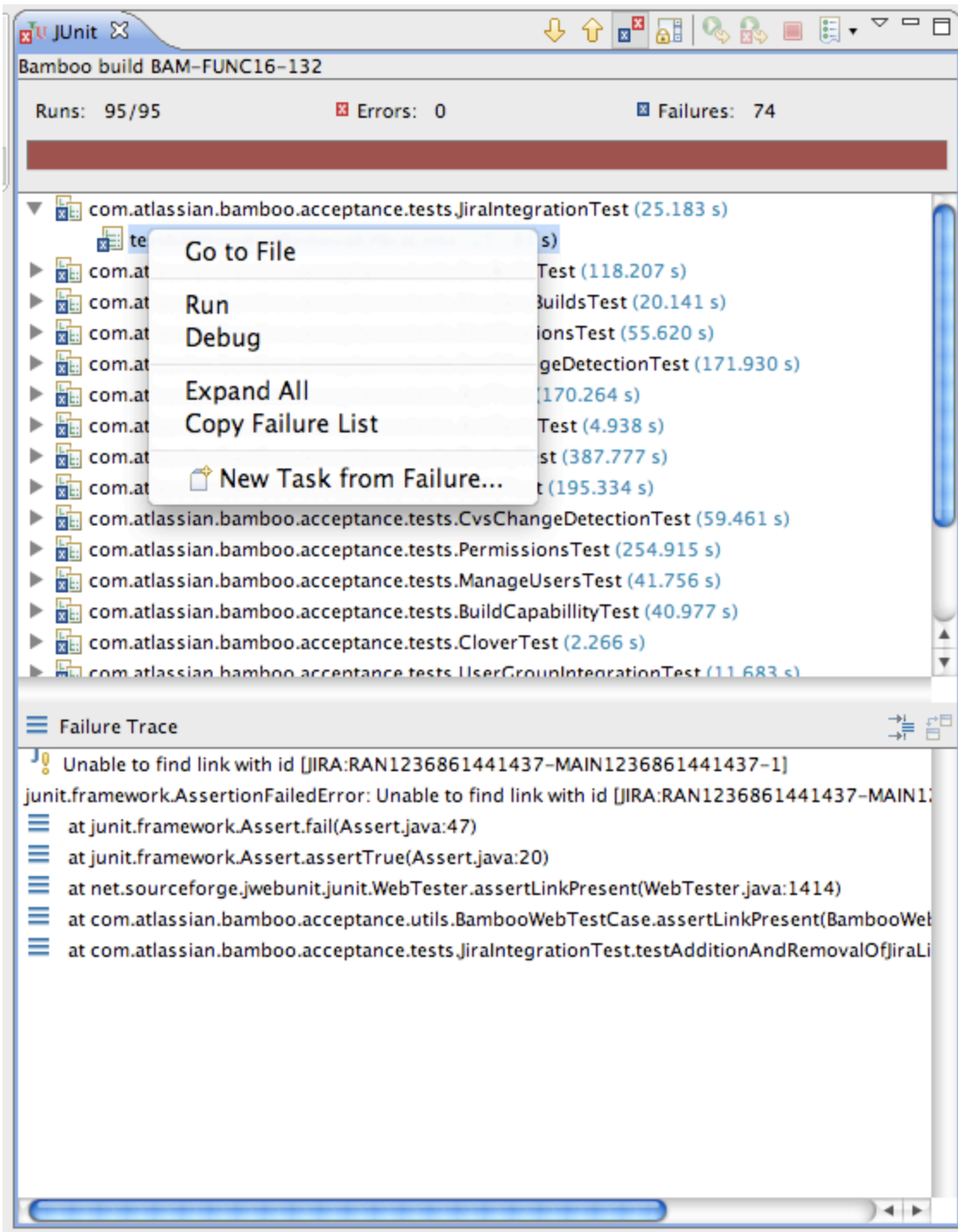
```

Bamboo

10

Monitor the Failed Tests

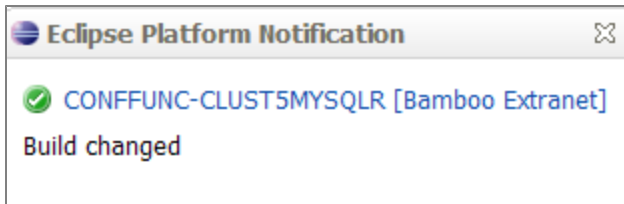
- See the test results for the selected build.
- Re-run the failed tests locally, to make sure your fixes work before breaking the build again.
- Create a new Mylyn task from the failure. This Mylyn task could be a JIRA issue.



11

Receive Notification of Build changes

- Receive build notifications via the Atlassian Eclipse Connector's integration with the standard Eclipse notification messages.



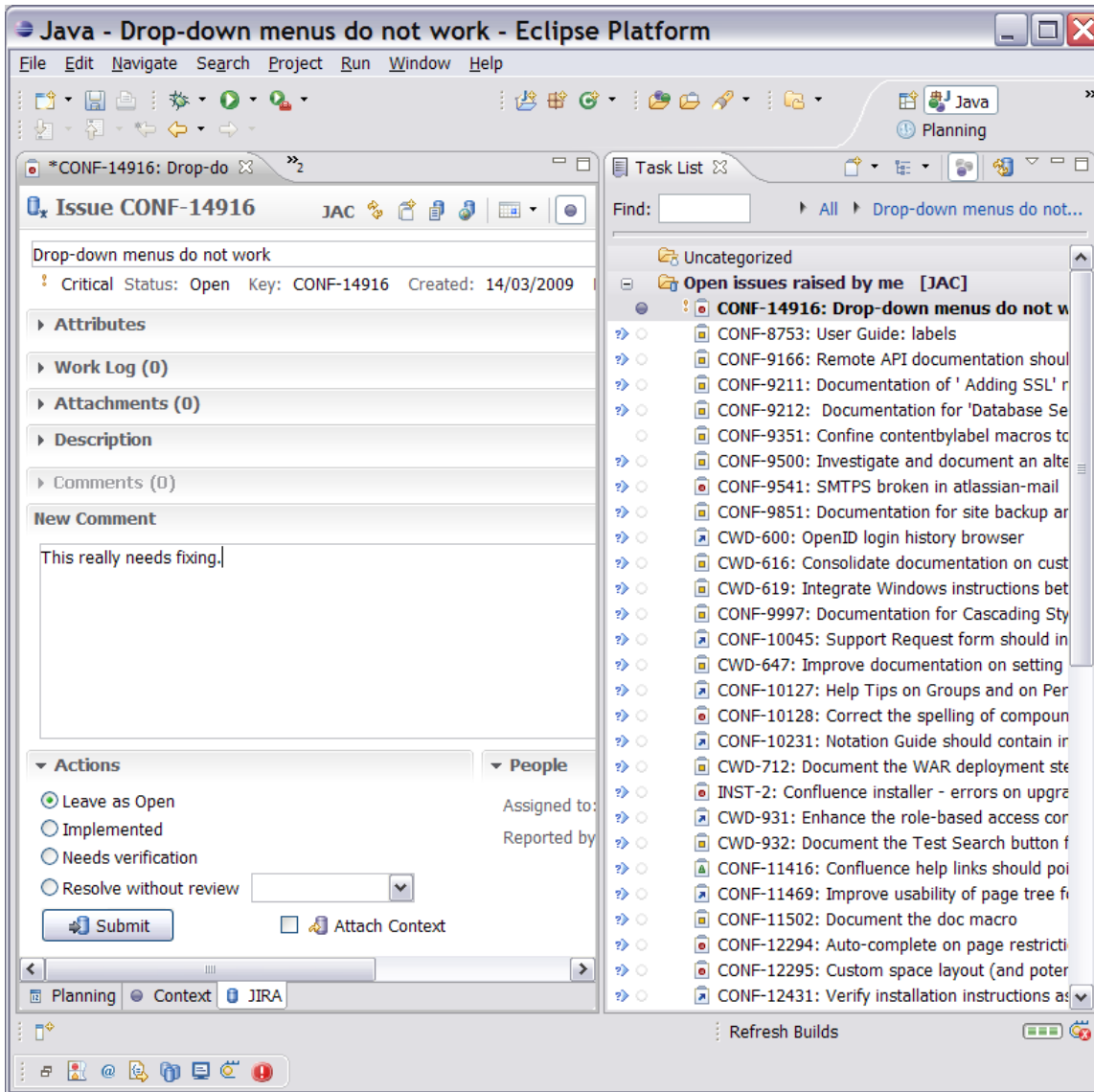
JIRA Issues in Eclipse

The new connector incorporates the [JIRA Mylyn plugin](#), which was previously available as a JIRA-only connector. Configure your [JIRA](#) server as a task repository. Use Mylyn's context management to keep track of the files and other assets associated with your issue.



Use the Awesome JIRA Mylyn Plugin as Part of the New Connector

- Configure your [JIRA](#) server as a task repository.
- Build Mylyn queries to filter your issues, based on your JIRA server filters or on custom criteria.
- Manage the issues via the Mylyn task list.
- Take advantage of Mylyn's context management to keep track of the files and other assets associated with your issue.
- Create a new JIRA issue and edit existing issues.
- Add and reply to comments.
- Receive notification of updated issues.



Atlassian IDE Plugin 0.4 for Eclipse - Release Notes

1 August 2008

With pleasure, Atlassian presents **version 0.4** of the Atlassian IDE Plugin for Eclipse.

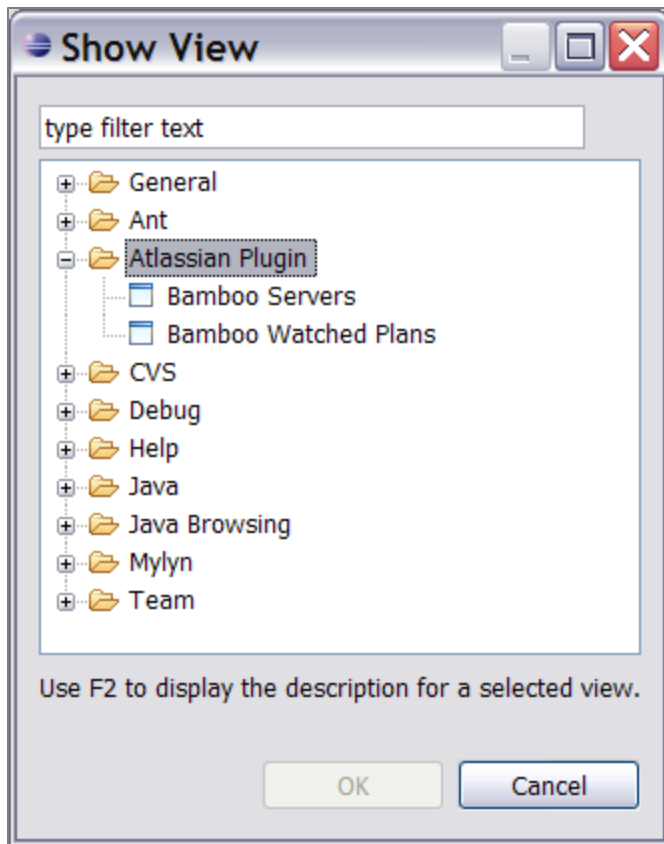
This release brings further enhancements to the Bamboo support in the plugin. You can now connect to multiple Bamboo servers instead of just one. For each of your build plans, you can now show the full build log within the Eclipse IDE.

Highlights of this Release

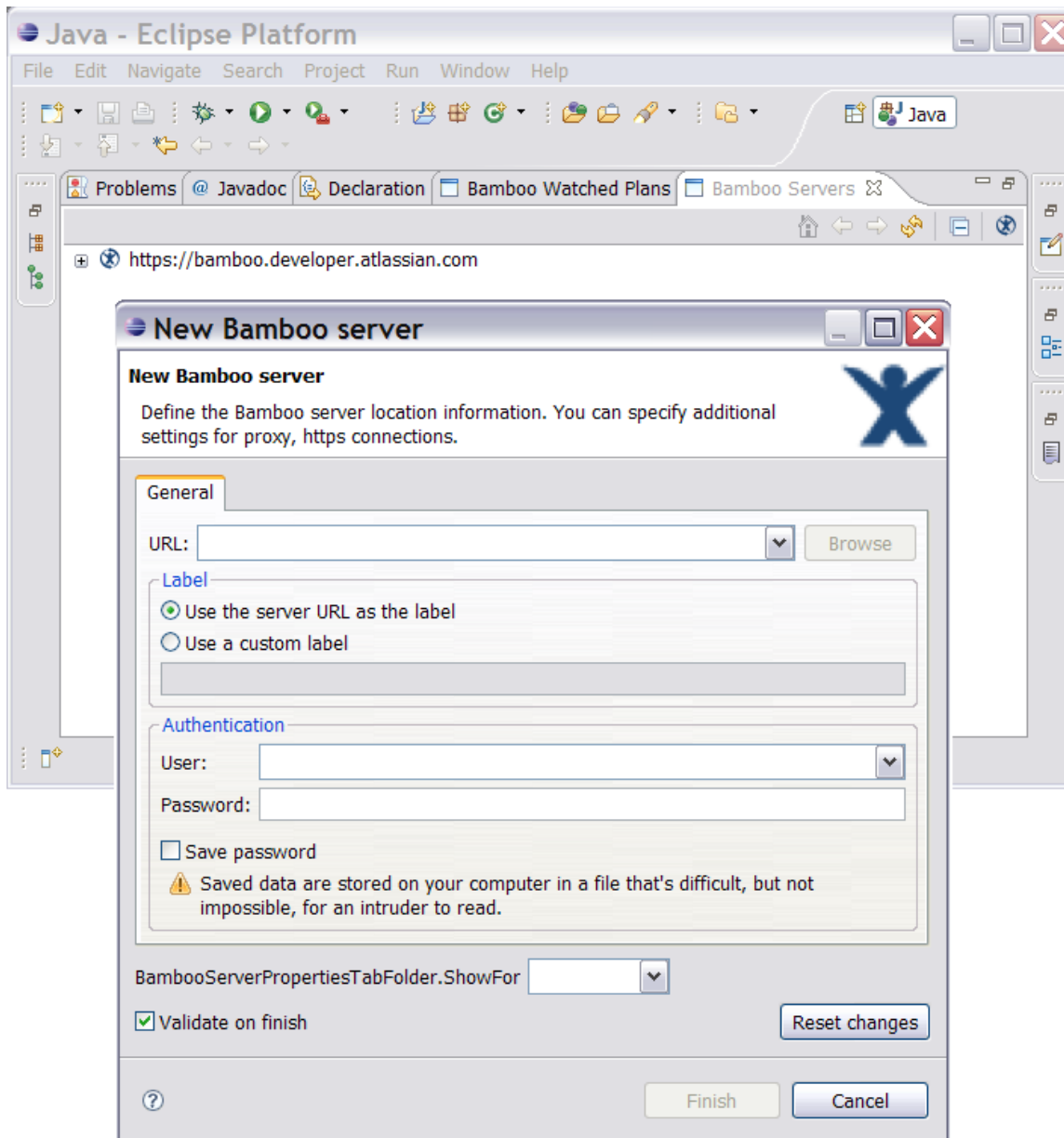
1

Bamboo Servers View

- There are now two Bamboo views: 'Bamboo Servers' and 'Bamboo Watched Plans'.



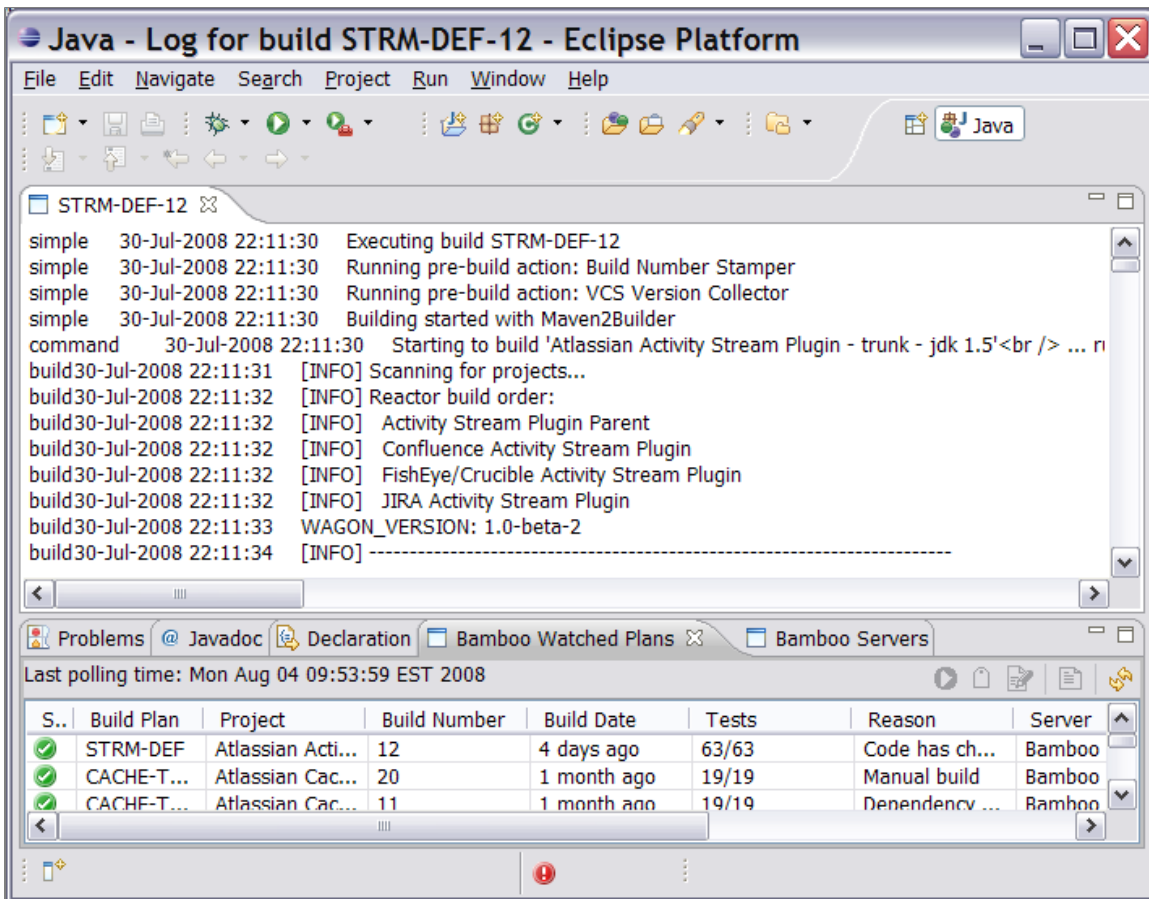
- The 'Bamboo Watched Plans' view shows the information for your selected build plans, as in previous versions of the plugin.
- The new 'Bamboo Servers' view allows you to add Bamboo servers and view the plans defined on each server.
- You can now connect to multiple Bamboo servers instead of just one.



2

Full Build Log

- You can now show the full build log within the Eclipse IDE. Just select the build plan, then click the 'Show Build Log' icon in the toolbar.



Complete List of Fixes in the Atlassian IDE Plugin 0.4 for Eclipse

JIRA Issues (4 issues)			
Key	Summary	Priority	Status
PLE-51	Add new view to show defined bamboo servers	↑	Resolved
PLE-49	Release notes for Eclipse plugin 0.4	↑	Closed
PLE-45	Show full log for build	↑	Resolved
PLE-40	Several icons in status bar	↑	Resolved

Atlassian IDE Plugin 0.3 for Eclipse - Release Notes



Plugin version 0.3 requires Eclipse 3.4

Update on 25/7/2008: From version 0.3 of the Atlassian IDE Plugin, you will need Eclipse version 3.4.0 or later. For more information, see [PLE-46](#).

18 July 2008

With pleasure, Atlassian presents **version 0.3** of the Atlassian IDE Plugin for Eclipse. Here are the highlights of this release:

- A new Bamboo icon in the Eclipse status bar shows the summary status of builds: red means at least one build is currently failing; green means all builds are passing.
 - The configuration screen allows you to select your favourite build plans as defined on the Bamboo server.
 - You can now add labels and comments to your build plans (Bamboo 2 only).
- Below is a list of all the fixes and improvements in this release.

The Atlassian IDE Plugin is available for IDEA too

Take a look at the [release notes](#) for the IDEA version of the plugin.

Don't have the Atlassian IDE Plugin for Eclipse yet?









Follow our easy [installation instructions](#).

This is an open source project. You can get the source code from [our SVN repository](#).

The plugin's blog

From time to time, the plugin development team posts some information on the plugin's [blog](#).

Complete List of Fixes in the Atlassian IDE Plugin 0.3 for Eclipse

JIRA Issues (8 issues)			
Key	Summary	Priority	Status
PLE-39	Add status bar notification icon	↑	 Resolved
PLE-38	Documentation for Eclipse plugin 0.3	↑	 Resolved
PLE-23	"favourite plans" handling in the config dialog	↑	 Resolved
PLE-22	Please update teh IDE plugin web page (WAC/software/ideplugin) to reflect the availability of an Eclipse version	↑	 Closed
PLE-16	Show license text when installing plug-in	↑	 Resolved
PLE-9	Enable label and comment features for bamboo2	↑	 Resolved
PLE-34	License text incomplete during installation on Eclipse 3.4	↓	 Resolved
PLE-32	Bamboo tab does not display project name and server name	↓	 Resolved

Atlassian IDE Plugin 0.2 for Eclipse - Release Notes

3 July 2008

With pleasure, Atlassian presents **version 0.2** of the Atlassian IDE Plugin for Eclipse.

Here are the highlights of this release:

- The configuration screen allows you to select your Bamboo build plans from a list, rather than typing them into a text box.
- The new 'Test Connection' button lets you verify the connection data you have entered, by connecting to the Bamboo server.
- The new 'Refresh' button lets you retrieve the list of build plans, using the most recent data entered into the form.
- A notification in the Eclipse status bar lets you know the status of background plugin activities, such as retrieving the list of plans or the build status.

Below is a list of all the fixes and improvements in this release.

The Atlassian IDE Plugin is available for IDEA too

Take a look at the [release notes](#) for the IDEA version of the plugin.

Don't have the Atlassian IDE Plugin for Eclipse yet?






Follow our easy [installation instructions](#).

This is an open source project. You can get the source code from [our SVN repository](#).

The plugin's blog

From time to time, the plugin development team posts some information on the plugin's [blog](#).

Complete List of Fixes in the Atlassian IDE Plugin 0.2 for Eclipse

JIRA Issues (5 issues)				
Key	Summary	Priority	Status	
PLE-30	Documentation for Eclipse plugin 0.2	↑	 Resolved	
PLE-21	a list of bamboo plans in Eclipse config dialog	↑	 Resolved	
PLE-19	Documentation for Eclipse plugin 0.1	↑	 Closed	
PLE-7	Add some notification (animation) that builds' list is reloaded for bamboo tab	↑	 Resolved	
PLE-3	Add test connectio button to the config window	↑	 Resolved	

Atlassian IDE Plugin 0.2 for Eclipse - Upgrade Notes

Below are some essential notes on upgrading to version 0.2 of the Atlassian IDE Plugin for Eclipse. For details of the new features and improvements in this release, please read the [Release Notes](#).

Upgrade Instructions

1. Please upgrade as usual. Instructions are available in the [Upgrade Guide](#).
2. After upgrading, please change the name of your existing configuration file from
`atlassian_eclipse_plugin.prefs`
to
`atlassian-eclipse-plugin.prefs`
(replacing the underscores with hyphens).

Background Information

For the first time in this release, we built the release automatically using Maven with an Ant script. A side effect is that your plugin configuration data will be empty after upgrade. To retrieve the configuration data, change the file name as described above.

RELATED TOPICS

[Atlassian IDE Plugin 0.2 for Eclipse - Release Notes](#)

Atlassian IDE Plugin 0.1 for Eclipse - Release Notes

23 June 2008

With pleasure, Atlassian presents **version 0.1** of the Atlassian IDE Plugin for Eclipse. Highlights of this release include:

- You can configure the polling interval that the plugin will use to monitor the Bamboo build plans.
- Bamboo will now notify you of build failures and successes via a popup. The popup's behaviour is configurable via the plugin's Preferences window.
- You can change the order of the columns in the Bamboo view. The plugin will remember the column order the next time you start Eclipse.

The Atlassian IDE Plugin is available for IDEA too

Take a look at the [release notes](#) for the IDEA version of the plugin.

Don't have the Atlassian IDE Plugin for Eclipse yet?





Follow our easy [installation instructions](#).

This is an open source project. You can get the source code from our [SVN repository](#).

The plugin's blog

The plugin's team from time to time posts some info on the plugin's [blog](#)

Complete List of Fixes in the Atlassian IDE Plugin 0.1 for Eclipse

JIRA Issues (5 issues)			
Key	Summary	Priority	Status
PLE-18	Separate the release notes for the IDEA and Eclipse versions of the plugin	↑	 Closed
PLE-17	Add setting bamboo polling time to the configuration	↑	 Closed
PLE-5	Add feature to remember bamboo tab columns width and order between Eclipse sessions	↑	 Closed
PLE-2	Eclipse: bamboo notification "toaster" popup	↑	 Closed
PLE-1	Allow to change columns order for bamboo tab	↑	 Closed

Configuring the Eclipse Connector

This page tells you how to set up your [JIRA](#), [Bamboo](#), [Crucible](#) and [FishEye](#) servers as **task repositories** in [Eclipse](#).

On this page:

- [Prerequisites](#)
- [Setting Up your Task Repositories](#)
 - [Setting Up a Bamboo Task Repository](#)
 - [Setting Up a Crucible Task Repository](#)
 - [Setting Up a FishEye Task Repository](#)
 - [Setting Up a JIRA Task Repository](#)

Prerequisites

Before you can set up your task repositories for the Atlassian Connector for Eclipse, you will need to do the following:

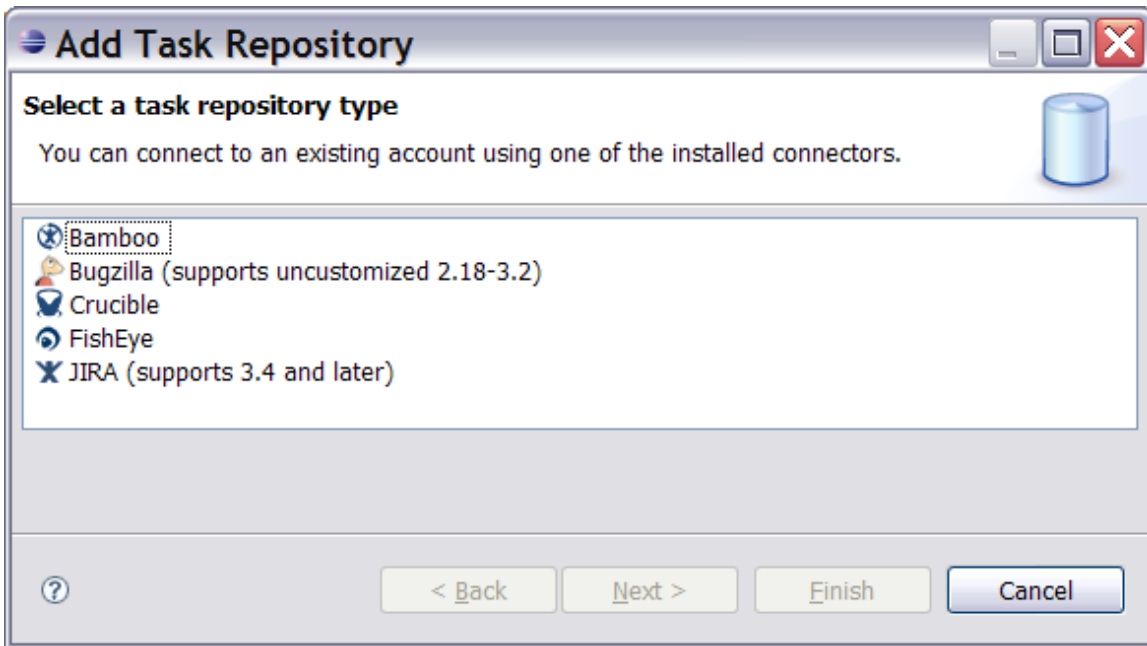
- Install the Atlassian Connector for Eclipse as described in the [installation guide](#).
- Check your version of Mylyn.

The Atlassian Connector for Eclipse makes use of [Mylyn](#)'s task-focused interface. Mylyn is shipped with the more recent versions of Eclipse. The Atlassian Connector for Eclipse requires **Mylyn 3.2.x**.

- If you do not already have Mylyn, it will be installed automatically when you install one of the Atlassian Connector's features.
- If you already have Mylyn installed, please check your version of Mylyn. If your version of Mylyn is earlier than 3.2, please upgrade to [Mylyn 3.2.x](#).
- The Atlassian Connector for Eclipse now includes the [JIRA Mylyn connector](#), which was previously available as a JIRA-only connector.
- The Mylyn documentation includes a [quick start guide](#).

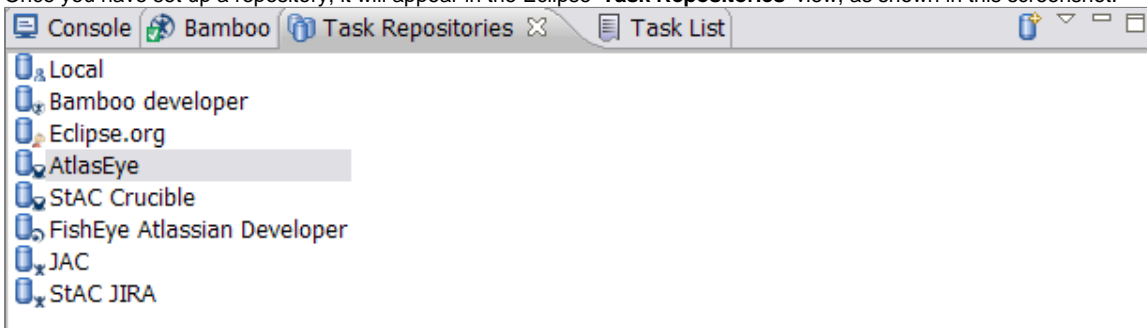
Setting Up your Task Repositories

1. In Eclipse Mylyn, open the '**Task Repositories**' view:
 - Click '**Window**', '**Show View**', '**Other**'.
 - Open the '**Tasks**' category and select the '**Task Repositories**' view.
2. Click the '**Add Task Repository**' icon.
3. The 'Add Task Repository' screen appears, as shown below:



4. Now you can select and configure one or more of the task repositories and configure them to match your server(s), as described below.

Once you have set up a repository, it will appear in the Eclipse **'Task Repositories'** view, as shown in this screenshot:



Setting Up a Bamboo Task Repository

1. On the Eclipse 'Add Task Repository' screen, select the **'Bamboo'** task repository type and click **'Next'**.
2. The 'Bamboo Repository Settings' screen appears, as shown below:

Add Task Repository...

Bamboo Repository Settings

Authentication credentials are valid.

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

► **Http Authentication**

► **Proxy Server Configuration**

▼ **Build Plans**

- ☐ Atlassian Javascript - Selenium, IE7 (AJS-SELENIUMIE7)
- ☐ Atlassian Javascript - Trunk (AJS-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.5 (LICENSE-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.6 (LICENSE-TRUNKJDK16)
- ☐ Clover + Sonar - Sonar - Bamboo - Unit Test (QA-BAM)
- ☐ Clover + Sonar - Sonar - Clover (QA-CLOVER)
- ☐ Clover + Sonar - Sonar - Common - Bamboo Maven Plugin (QA-BAM)
- ☐ Clover + Sonar - Sonar - Common - Bandana (QA-BANDANA)
- ☐ Clover + Sonar - Sonar - Common - Base Archetype (QA-BARCHTYPE)
- ☐ Clover + Sonar - Sonar - Common - Bonnie (QA-BONNIE)
- ☐ Clover + Sonar - Sonar - Common - Bucket (QA-BUCKET)

[Create new account](#) [Change account settings](#)

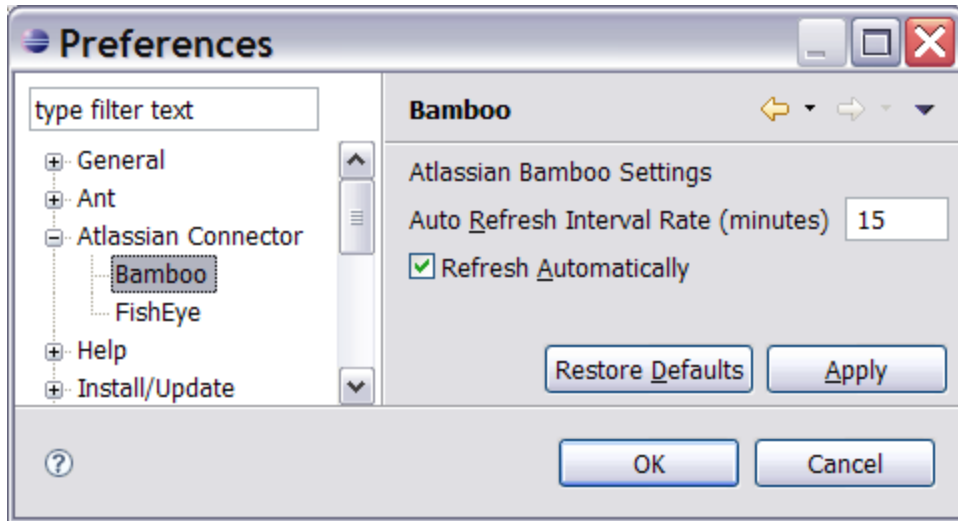
Validate Settings < Back Next > Finish Cancel

3. Enter the following information:


- **Server** — The location (URL) of your Bamboo server.
- **Label** — A descriptive name for your Bamboo server, e.g. 'Bamboo Extranet'.
- **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
- **User ID** — The username you use to connect to your Bamboo server.
- **Password** — Your password on the Bamboo server, matching the above username.
- **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.


If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

4. Click the '**Refresh**' button to verify the information you have entered. The connector will attempt to connect to the Bamboo server. It will also retrieve the latest list of build plans from the Bamboo server.
5. Select the plans that you want the connector to watch. You can either select the plans individually or you can click the '**Favourites**' button to choose your favourite plans as defined on the Bamboo server.
6. Click '**Finish**' to save the changes.
7. The new task repository appears in your Eclipse '**Task Repositories**' view.
8. If necessary, you can now you can adjust the Bamboo settings.
 - Select '**Window**', '**Preferences**' then open the Bamboo settings panel as shown below:




- Enter the '**Auto Refresh Interval Rate**'. This is the number of minutes that the connector will wait between calls to fetch new information from your Bamboo server, when the connector is set to refresh automatically.
 - Tick or untick the '**Refresh Automatically**' checkbox. Untick this option if you do not want the connector to poll the server automatically. In that case, you will need to refresh you Bamboo information manually by synchronising the Bamboo task repository.
9. Next step: To see your Bamboo builds, open the '**Bamboo**' view. (Select '**Window**', '**Show View**', '**Other**', then open the '**Atlassian**' folder.)

 You can configure one or more Bamboo repositories, i.e. you can connect to more than one Bamboo server.

 **Bamboo builds are shown in a Bamboo-specific view**
Bamboo has its own '**Bamboo**' view. This is different from JIRA and Crucible, where issues and reviews are included into your 'Task List' view. Next, take a look at [the Bamboo view](#).

Setting Up a Crucible Task Repository

 **Single server for both Crucible and FishEye = single task repository**
If you have a single server for both FishEye and Crucible, you should set up the Crucible task repository and then select the '**Crucible Server Contains FishEye Instance**' option as described below. Due to Mylyn requirements, you cannot set up two separate task repositories for the same URL.

1. On the Eclipse 'Add Task Repository' screen, select the '**Crucible**' task repository type and click '**Next**'.
2. The 'Crucible Repository Settings' screen appears, as shown below:

Properties for Task Repository

Crucible Repository Settings
Enter Crucible server information

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

► **Http Authentication**

► **Proxy Server Configuration**

▼ **Review Activation**

Activate Review when opening a file from within the Review Editor

☐ Always ☐ Never ☒ Prompt


▼ **FishEye**

☒ Crucible Server Contains FishEye Instance

[Create new account](#) [Change account settings](#)

3. Enter the following information:
 - **Server** — The location (URL) of your Crucible server.
 - **Label** — A descriptive name for your Crucible server, e.g. 'Extranet Crucible'.
 - **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
 - **User ID** — The username you use to connect to your Crucible server.
 - **Password** — Your password on the Crucible server, matching the above username.
 - **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.

If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.
4. Click the **'Validate Settings'** button to verify the information you have entered.
5. Choose a **'Review Activation'** setting. These settings determine what happens when you open a source code file from within the Crucible review editor in Eclipse. For example, you may open a Crucible review within Eclipse. The review will open in an Eclipse editor view, and its 'Review Files' section will show the files in the review. You can then click a file to open it in an Eclipse editor. The options are as follows:
 - **'Always'** — The Crucible review will be automatically activated if you open a file from the Crucible review editor.
 - **'Never'** — The Crucible review will never be automatically activated.
 - **'Prompt'** (default) — When you open a file from the Crucible review editor, the connector will ask you whether you want to activate the Crucible review. This is the default setting.
6. If your Crucible server is linked to a FishEye server:
 - Put a tick in the checkbox labelled **'Crucible Server Contains FishEye Instance'**.
 - Set up your FishEye mappings, as described in the FishEye section of this documentation.
7. Click **'Finish'** to save the changes.
8. Eclipse Mylyn will prompt you to add a new query for the new Crucible repository. This is where you will choose your Crucible filter, to determine which reviews appear in your task list. You can do this now or skip this step and do it later. You can also add more queries later.
9. The new task repository appears in your Eclipse **'Task Repositories'** view. Your Crucible reviews will appear in your 'Task List' view.

 You can configure one or more Crucible repositories, i.e. you can connect to more than one Crucible server.

Setting Up a FishEye Task Repository

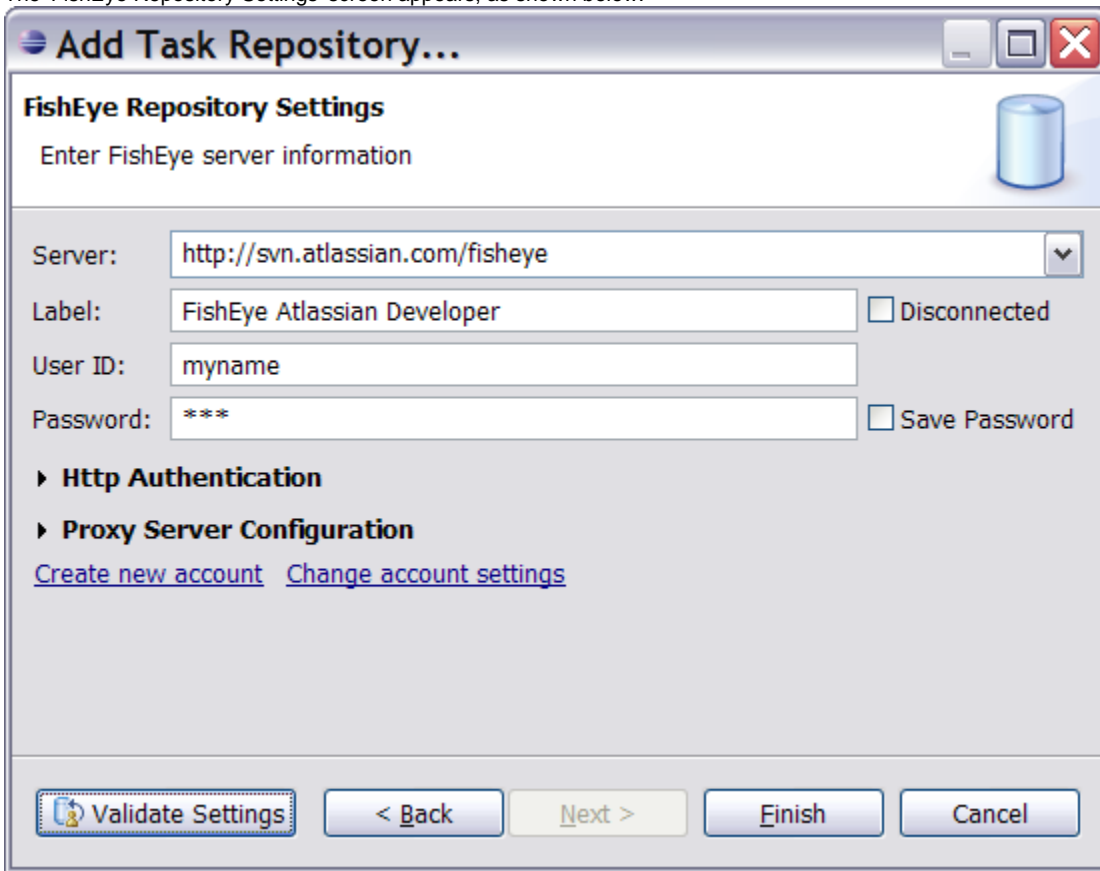


Single server for both Crucible and FishEye = single task repository


If you have a single server for both FishEye and Crucible, you should set up the Crucible task repository and then select the '**Crucible Server Contains FishEye Instance**' option as described [above](#). Due to Mylyn requirements, you cannot set up two separate task repositories for the same URL.

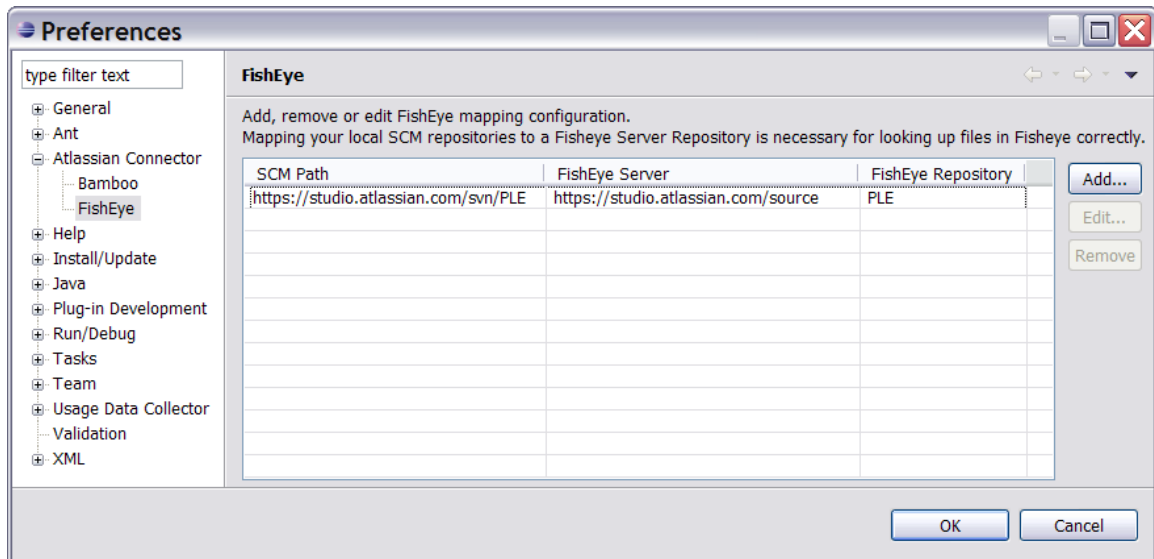
If you do not have a Crucible server, you can hook up your FishEye server as an Eclipse task repository without the additional Crucible functionality.

1. On the Eclipse 'Add Task Repository' screen, select the '**FishEye**' task repository type and click '**Next**'.
2. The 'FishEye Repository Settings' screen appears, as shown below:

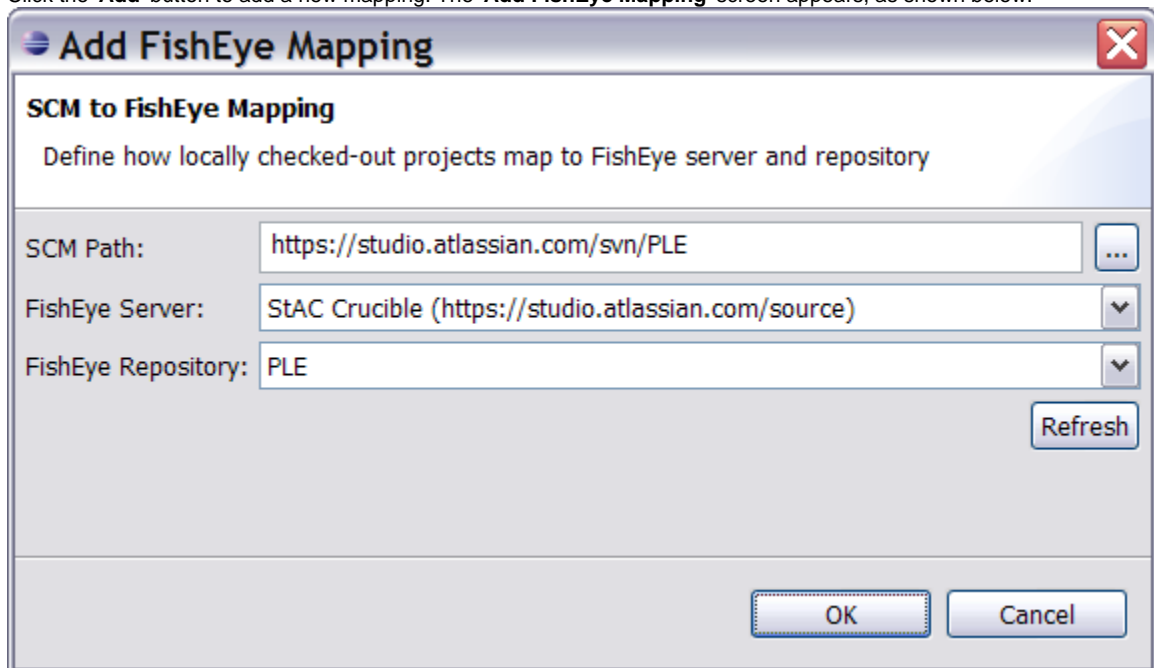


3. Enter the following information:
 - **Server** — The location (URL) of your FishEye server.
 - **Label** — A descriptive name for your FishEye server, e.g. 'FishEye Atlassian Developer'.
 - **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
 - **User ID** — The username you use to connect to your FishEye server.
 - **Password** — Your password on the FishEye server, matching the above username.
 - **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.


 If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.
4. Click the '**Validate Settings**' button to verify the information you have entered.
5. Click '**Finish**' to save the changes.
6. The new task repository appears in your Eclipse '**Task Repositories**' view.
7. Now you need to map your source code to your FishEye repository and FishEye project.
 - Select '**Window**', '**Preferences**' then open the FishEye settings panel as shown below:



- Click the 'Add' button to add a new mapping. The 'Add FishEye Mapping' screen appears, as shown below:



- Enter the '**SCM Path**' — Supply the path to your source code as used locally by the project.
- Select the '**FishEye Server**' as configured in your task repositories. (See instructions above.)
- The connector will retrieve a list of FishEye repositories from the server you have supplied. Select the appropriate '**FishEye Repository**' to map your local project to the FishEye repository.

 Some notes about the FishEye integration:


- You can configure one or more FishEye repositories, i.e. you can connect to more than one FishEye server.
- We have chosen to configure FishEye servers as standard Mylyn task repositories. They are not really regular repositories like JIRA or Crucible, because there are no tasks associated with a FishEye task repository. We are using the task repository concept to take advantage of various infrastructural facilities available in Mylyn and to make user experience more consistent with the rest of the Atlassian products. See [More about Configuring FishEye Repositories in Eclipse](#).

Setting Up a JIRA Task Repository

1. On the Eclipse 'Add Task Repository' screen, select the '**JIRA**' task repository type and click '**Next**'.
2. The 'JIRA Repository Settings' screen appears, as shown below:

Add Task Repository...

JIRA Repository Settings

 Authentication credentials are valid.

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

Additional Settings

Compression: ☐ Enabled

Refresh configuration: ☐ Automatically

Completed tasks: ☐ Based on resolution

Subtasks: ☐ Show linked tasks

Follow redirects: ☐ Enabled

Time tracking: working days per week

working hours per day

Search results: ☒ Limit

► Advanced Configuration

Character encoding: ☒ Default (UTF-8)

☐ Other:

► **Http Authentication**

► **Proxy Server Configuration**

Task Editor Settings

☐ Plain Text



☒ Confluence (default)

☐ MediaWiki

☐ TWiki


☐ Textile

☐ TracWiki


  Validate Settings

3. Enter the following information:

- **Server** — The location (URL) of your JIRA server.
- **Label** — A descriptive name for your JIRA server, e.g. 'JAC'.
- **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
- **User ID** — The username you use to connect to your JIRA server.
- **Password** — Your password on the JIRA server, matching the above username.
- **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.

 If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

4. If necessary, you can change the additional settings. If in doubt, leave these settings as set by default.
 - The **'Task Editor Settings'** section is used to configure the markup language used by the task editor when Mylyn's [WikiText](#) extension is installed. By default, the Atlassian Connector for Eclipse will set the markup language to **'Confluence'**, so that WikiText recognises Confluence wiki markup as used by JIRA. See the [Mylyn WikiText User Guide](#) for more information.
5. Click **'Finish'** to save the changes.
6. Eclipse Mylyn will prompt you to add a new query for the new JIRA repository. This is where you will choose your JIRA filter, to determine which issues appear in your task list. You can do this now or skip this step and do it later. You can also add more queries later.
7. The new task repository appears in your Eclipse **'Task Repositories'** view. Your JIRA issues will appear in your 'Task List' view.

 You can configure one or more JIRA repositories, i.e. you can connect to more than one JIRA server.

RELATED TOPICS

[Installing the Eclipse Connector](#)
[IDE Connector Documentation](#)

Collecting Usage Statistics for the Eclipse Connector

If you indicate your agreement, the Atlassian Connector for Eclipse will collect information on the usage of the connector and send the information to Atlassian.

On this page:

- [Reasons for Collecting Usage Data](#)
- [Enabling or Disabling the Collection of Information](#)
- [Submitting the Usage Data Manually](#)
- [Information Collected](#)
- [Mechanism Used to Send Information](#)
- [Another Way to Send Feedback](#)

Reasons for Collecting Usage Data

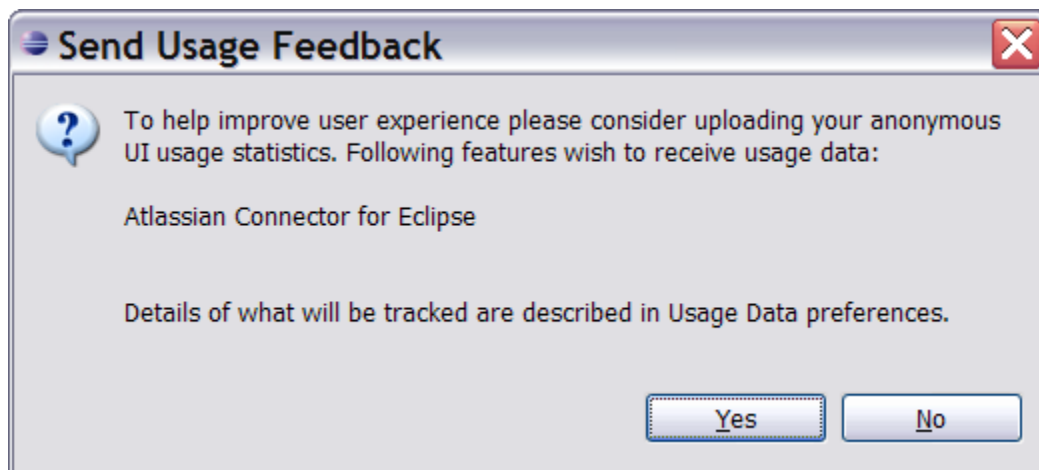
We are trying to better understand the usage of the connector, so that we can continue to develop a better product that meets your needs.

Enabling or Disabling the Collection of Information

We understand, of course, that some people do not wish to have any information collected from them. That's fine too.

When you first open your IDE with the connector installed, we ask you to decide whether you agree to participate in the collection of usage data. You can also change your decision later via the configuration settings, as described [below](#). The collection of data is disabled by default.

Screenshot 1: The connector asks your permission to collect anonymous data

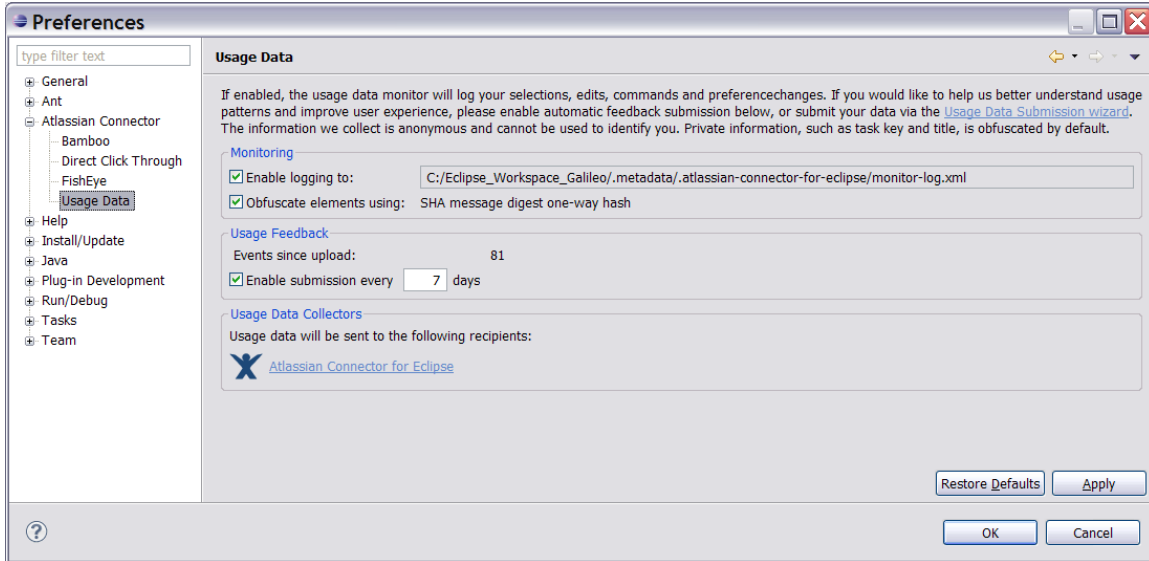


In addition, you can choose to enable or disable the collection of information at any time, and set other preferences.

[To set your usage data preferences,](#)

1. In Eclipse, open the **'Window'** menu and select **'Preferences'**.
2. Open the **'Atlassian Connector'** folder.
3. Select **'Usage Data'**.

Screenshot 2: Configuration preferences for usage data collection



Submitting the Usage Data Manually

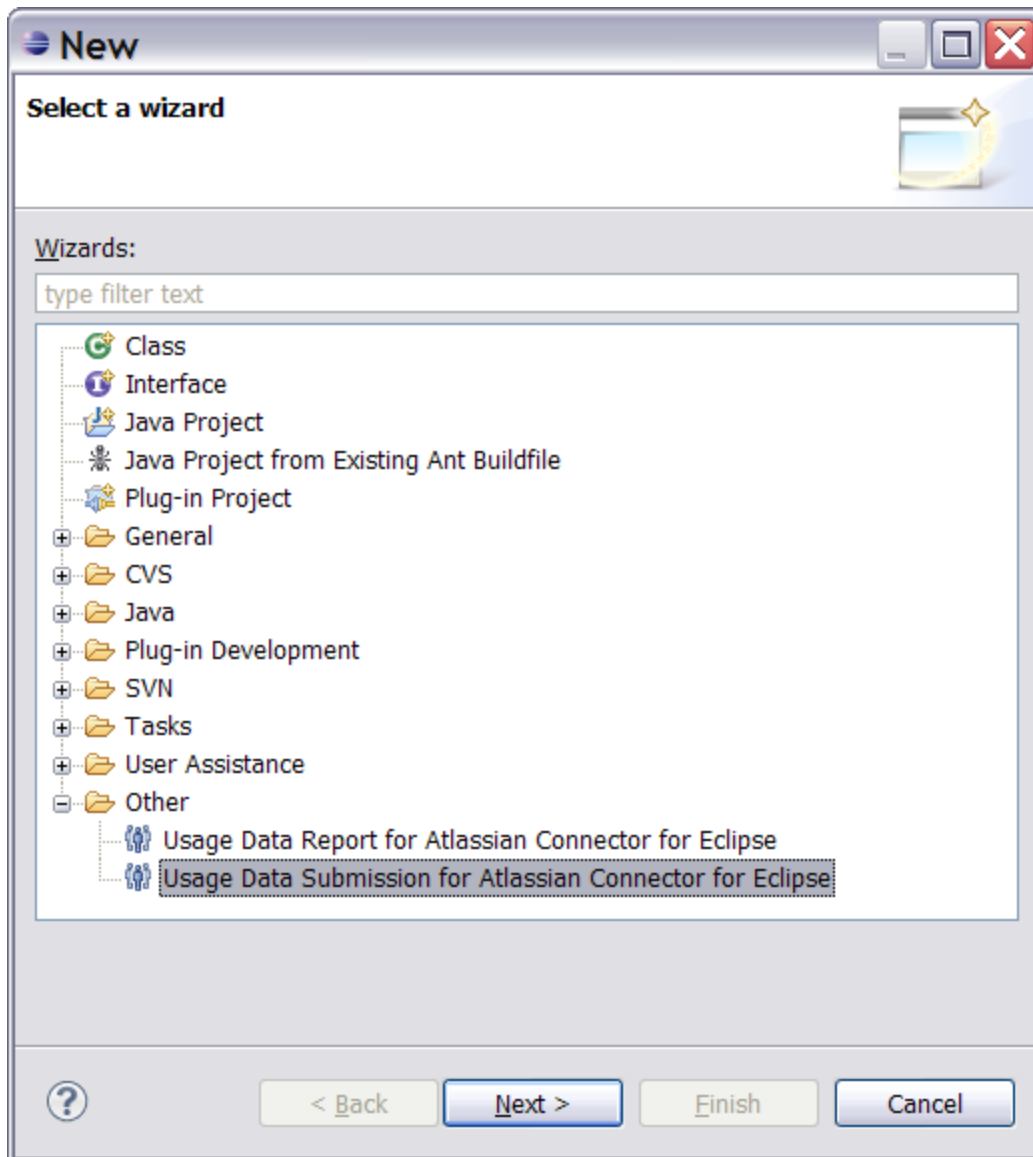
You can choose to submit your usage data manually, rather than allowing automatic submission. There are two ways to start the Usage Data Submission wizard:

- Either click the **'Usage Data Submission wizard'** link on the preferences screen, shown [above](#).
- Or start the wizard via the Eclipse menus, as described below.

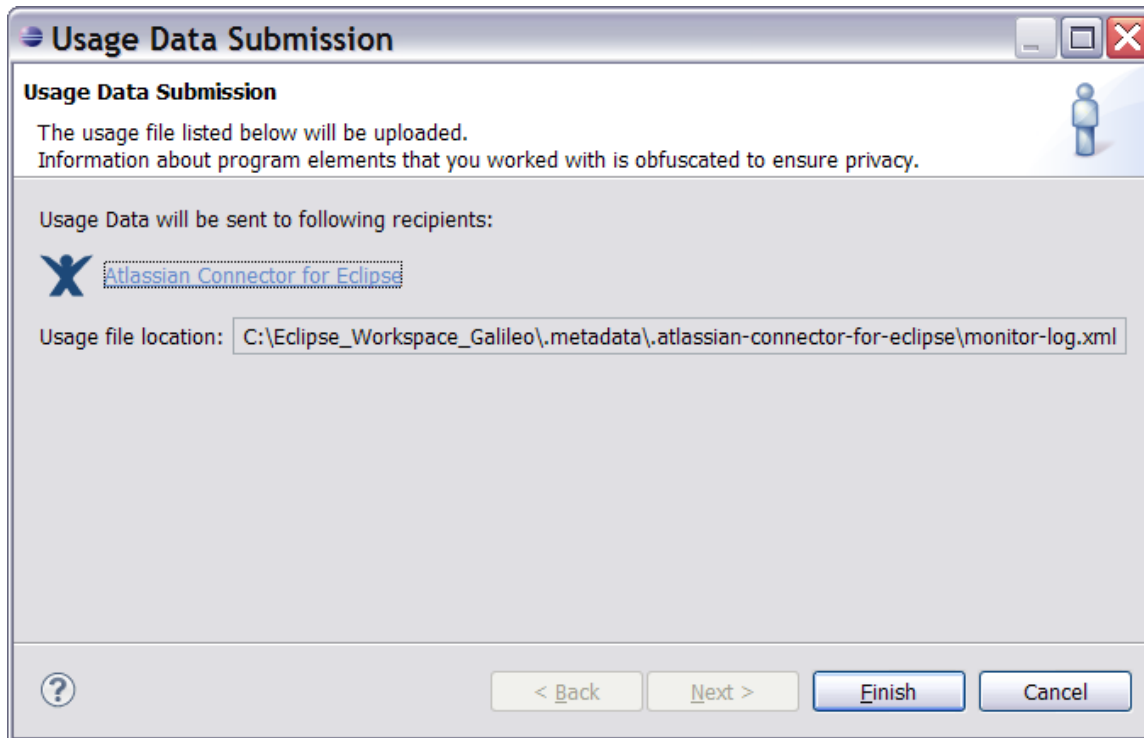
To submit usage data manually,

1. In Eclipse, open the **'File'** menu and select **'New', 'Other'**.
2. The 'Select a Wizard' screen appears, as shown in [screenshot 3](#) below. Open the **'Other'** folder.
3. Select **'Usage Data Submission for Atlassian Connector for Eclipse'**.
4. The 'Usage Data Submission' screen appears, as shown in [screenshot 4](#) below. Click the **'Finish'** button to send the data to Atlassian.

Screenshot 3: Starting the Usage Data Submission wizard



Screenshot 4: Submitting the usage data manually



Information Collected

We track general events in Eclipse, such as when you open a task, open a Bamboo build, open a Crucible review, get the Bamboo build details, get a FishEye resource link, get Bamboo build output, and so on.

We store things like:

- Event start and end dates.
- Event originator, such as the workbench window, command, toolbar, etc.
- Event name — usually the class name of the handler.
- Task type (jira, crucible, etc), for tasks opened by the user.

The information we collect is anonymous and cannot be used to identify you. Private information, such as task key and title, is obfuscated by default. If you choose not to obfuscate data, the information collected will include things like task key, task title, and in some cases a partial path name.

Mechanism Used to Send Information

The connector collects the information in a `monitor-log.xml` file, located in your Eclipse workspace directory as specified in your configuration settings (see [above](#)). The connector sends the data to Atlassian as a zipped XML file, periodically at the time interval specified in your configuration settings (see [above](#)).

Another Way to Send Feedback

Even if you decide not to send usage data through to us, we would still appreciate any feedback, comments, or suggestions you may have via our [JIRA issue tracker](#) or our [forums](#).

RELATED TOPICS

[Configuring the Eclipse Connector](#)

Using Bamboo in the Eclipse Connector

The Atlassian Connector for Eclipse gives you [Bamboo](#) build information right there in your integrated development environment.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

- [Configuring your Bamboo Options in Eclipse](#)
- [Working with Bamboo Builds in Eclipse](#)
 - [Monitoring Build Statuses in Eclipse](#)
 - [Viewing Build Details in Eclipse](#)
 - [Running a Build from within Eclipse](#)
 - [Viewing Test Results in Eclipse](#)
 - [Viewing a Build Log in Eclipse](#)
 - [Adding a Task from a Failed Build in Eclipse](#)

RELATED TOPICS

[Installation and Upgrade Guide for the Eclipse Connector](#)

Refer to the [Bamboo documentation](#) for more information about Bamboo builds.

Configuring your Bamboo Options in Eclipse

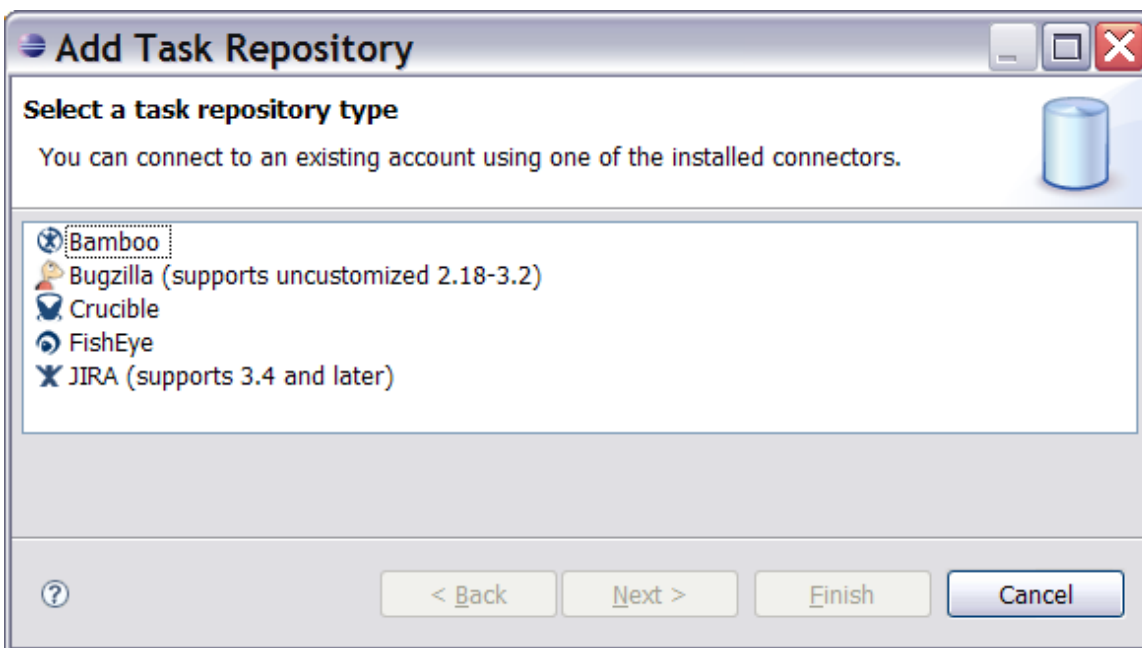
Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

On this page:

- [Adding a Task Repository](#)
- [Setting Up a Bamboo Task Repository](#)

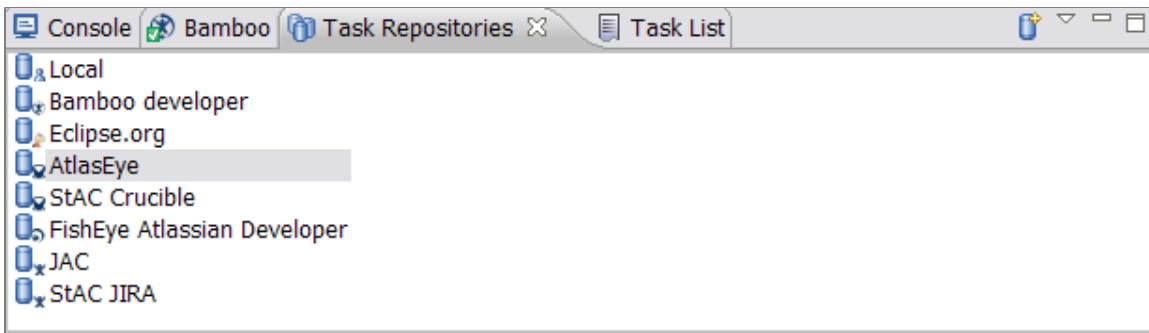
Adding a Task Repository

1. In Eclipse Mylyn, open the '**Task Repositories**' view:
 - Click '**Window**', '**Show View**', '**Other**'.
 - Open the '**Tasks**' category and select the '**Task Repositories**' view.
2. Click the '**Add Task Repository**' icon.
3. The 'Add Task Repository' screen appears, as shown below:



4. Now you can select and configure one or more of the task repositories and configure them to match your server(s), as described below.

Once you have set up a repository, it will appear in the Eclipse '**Task Repositories**' view, as shown in this screenshot:



Setting Up a Bamboo Task Repository

1. On the Eclipse 'Add Task Repository' screen, select the **'Bamboo'** task repository type and click **'Next'**.
2. The 'Bamboo Repository Settings' screen appears, as shown below:

Add Task Repository...

Bamboo Repository Settings

i Authentication credentials are valid.

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

► **Http Authentication**

► **Proxy Server Configuration**

▼ **Build Plans**


- ☐ Atlassian Javascript - Selenium, IE7 (AJS-SELENIUMIE7)
- ☐ Atlassian Javascript - Trunk (AJS-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.5 (LICENSE-TRUNK)
- ☒ Atlassian License - trunk - jdk 1.6 (LICENSE-TRUNKJDK16)
- ☐ Clover + Sonar - Sonar - Bamboo - Unit Test (QA-BAM)
- ☐ Clover + Sonar - Sonar - Clover (QA-CLOVER)
- ☐ Clover + Sonar - Sonar - Common - Bamboo Maven Plugin (QA-BAM)
- ☐ Clover + Sonar - Sonar - Common - Bandana (QA-BANDANA)
- ☐ Clover + Sonar - Sonar - Common - Base Archetype (QA-BARCHTY)
- ☐ Clover + Sonar - Sonar - Common - Bonnie (QA-BONNIE)
- ☐ Clover + Sonar - Sonar - Common - Bucket (QA-BUCKET)

[Create new account](#) [Change account settings](#)

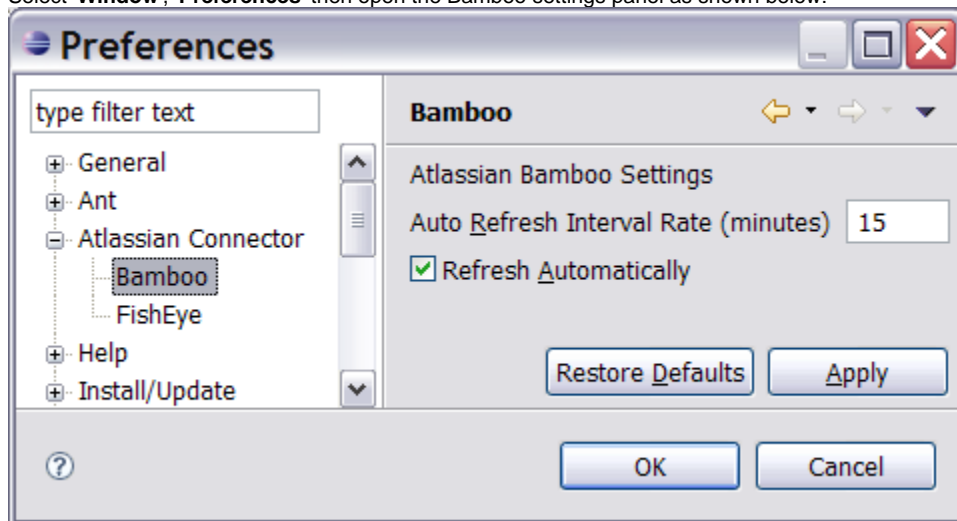
Validate Settings

3. Enter the following information:


- **Server** — The location (URL) of your Bamboo server.
- **Label** — A descriptive name for your Bamboo server, e.g. 'Bamboo Extranet'.
- **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
- **User ID** — The username you use to connect to your Bamboo server.
- **Password** — Your password on the Bamboo server, matching the above username.
- **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.


 If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

4. Click the **'Refresh'** button to verify the information you have entered. The connector will attempt to connect to the Bamboo server. It will also retrieve the latest list of build plans from the Bamboo server.
5. Select the plans that you want the connector to watch. You can either select the plans individually or you can click the **'Favourites'** button to choose your favourite plans as defined on the Bamboo server.
6. Click **'Finish'** to save the changes.
7. The new task repository appears in your Eclipse **'Task Repositories'** view.
8. If necessary, you can now you can adjust the Bamboo settings.
 - Select **'Window', 'Preferences'** then open the Bamboo settings panel as shown below:



- Enter the **'Auto Refresh Interval Rate'**. This is the number of minutes that the connector will wait between calls to fetch new information from your Bamboo server, when the connector is set to refresh automatically.
 - Tick or untick the **'Refresh Automatically'** checkbox. Untick this option if you do not want the connector to poll the server automatically. In that case, you will need to refresh you Bamboo information manually by synchronising the Bamboo task repository.
9. Next step: To see your Bamboo builds, open the **'Bamboo'** view. (Select **'Window', 'Show View', 'Other'**, then open the **'Atlassian'** folder.)

 You can configure one or more Bamboo repositories, i.e. you can connect to more than one Bamboo server.

 **Bamboo builds are shown in a Bamboo-specific view**
 Bamboo has its own **'Bamboo'** view. This is different from JIRA and Crucible, where issues and reviews are included into your 'Task List' view. Next, take a look at [the Bamboo view](#).

RELATED TOPICS

[Working with Bamboo Builds in Eclipse](#)
[Installation and Upgrade Guide for the Eclipse Connector](#)

Working with Bamboo Builds in Eclipse

With the Atlassian Connector for Eclipse, you can monitor and act upon your [Bamboo](#) builds from within your [Eclipse](#) IDE.

Prerequisites

Please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#), and defined at least one Bamboo server, as described in the [Configuration Guide](#).

Summary of What You Can Do

Below are the highlights of viewing and acting upon Bamboo builds within Eclipse. Follow the links to the relevant sections of the user guide.

- View a list of the builds you are monitoring, in the [Bamboo view in Eclipse](#)
- Receive [notification](#) of failed builds and other build changes
- Open the Bamboo [build details](#) in an Eclipse editor
- Open the Bamboo build details in your web browser, displaying the Bamboo web interface
- [Run](#) a build on the Bamboo server
- View a Bamboo [build log](#)
- View [test results](#)
- View [changed files](#) in the build
- [Comment](#) on a Bamboo build
- [Label](#) a Bamboo build
- Add a [new task based on a failed build](#)

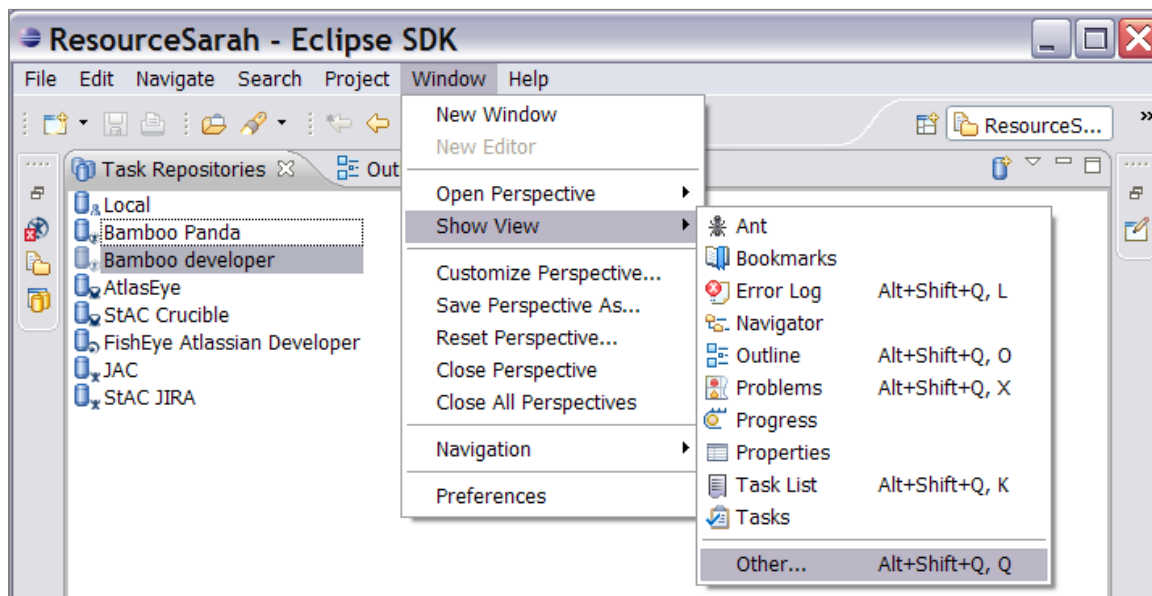
Opening the Bamboo View in Eclipse

Bamboo has its own '**Bamboo**' view in Eclipse. This is different from JIRA and Crucible, where issues and reviews are included into your 'Task List' view.

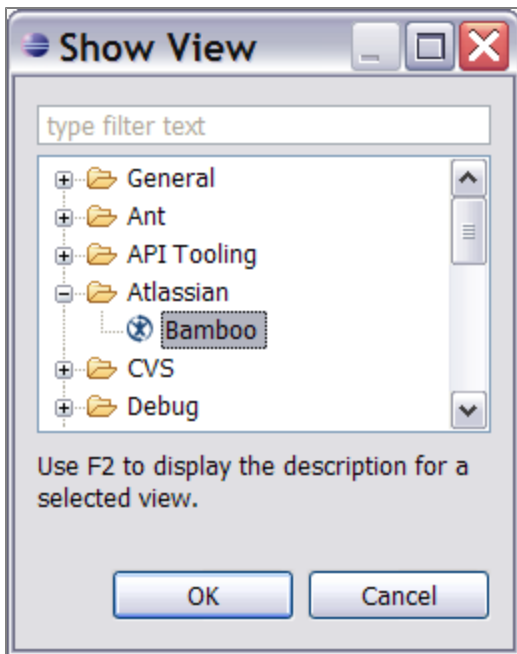
To open the '**Bamboo**' view,

1. In the Eclipse toolbar, select '**Window**', '**Show View**', '**Other**'. See [screenshot 1](#).
2. The 'Show View' popup window appears, as shown in [screenshot 2](#). Open the '**Atlassian**' folder and click '**Bamboo**'.
3. The Bamboo view appears, as shown in [screenshot 3](#).

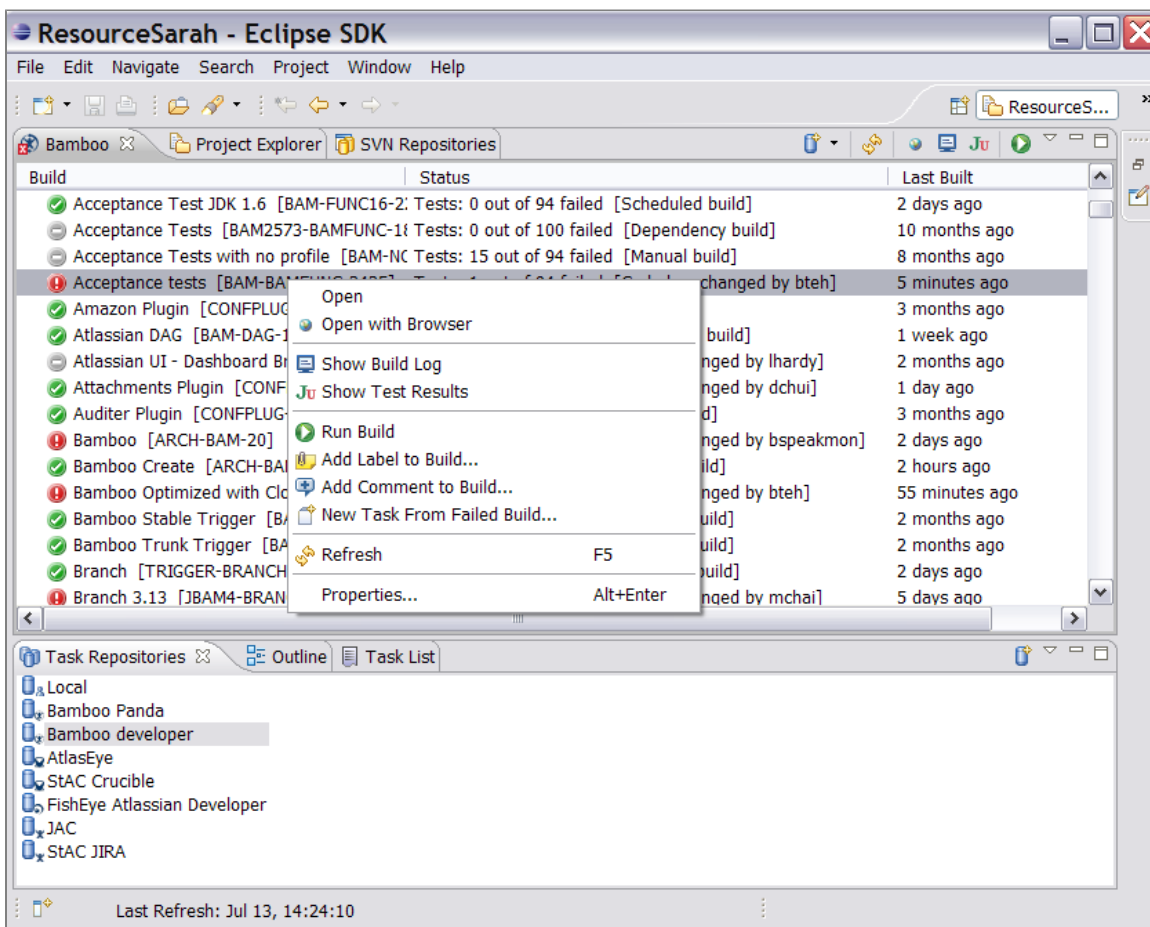
Screenshot 1: The Eclipse 'Show View' menu



Screenshot 2: The Eclipse 'Show View' popup



Screenshot 3: The Bamboo view




Next, take a look at the Bamboo [build details](#), as displayed in an Eclipse editor.

RELATED TOPICS

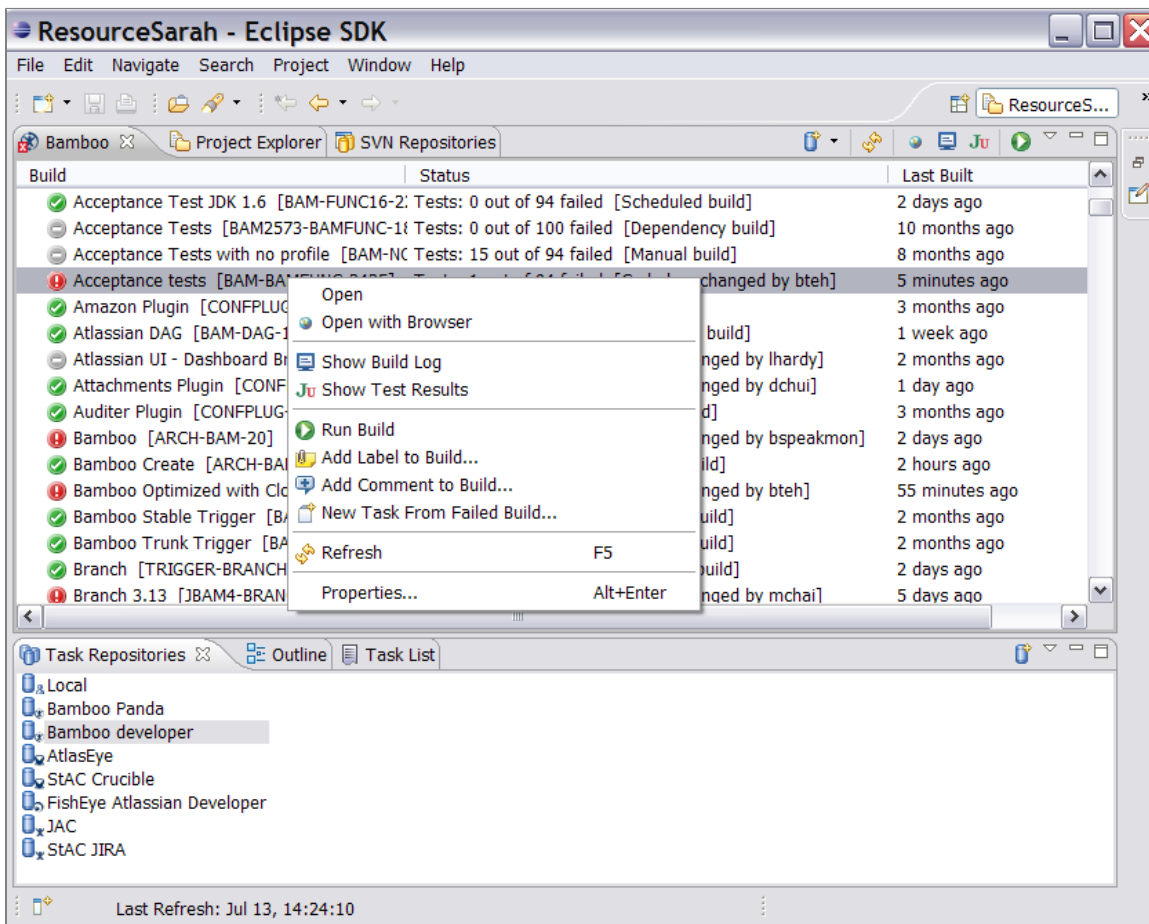
- [Installation and Upgrade Guide for the Eclipse Connector](#)
- [Configuring the Eclipse Connector](#)
- [Using Bamboo in the Eclipse Connector](#)
- [Using Crucible in the Eclipse Connector](#)
- [Using FishEye in the Eclipse Connector](#)
- [Using JIRA in the Eclipse Connector](#)
- [Eclipse Connector FAQ](#)

Refer to the [Bamboo documentation](#) for more information about Bamboo builds.

Monitoring Build Statuses in Eclipse

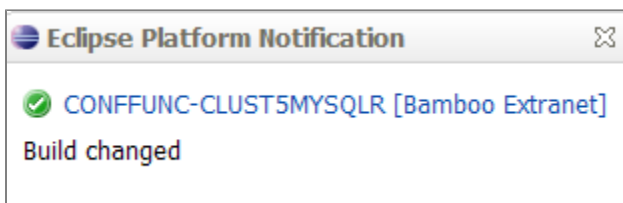
Once you have opened the [Bamboo view in Eclipse](#), the Bamboo icon at the top of the view shows the summary status of your builds. If at least one build is currently failing, the icon has a red cross: 

Screenshot: Bamboo status icon at the top of the Bamboo view



In addition, when the build status changes (e.g. a build fails or some other change occurs), an Eclipse notification popup window appears.

Screenshot: Bamboo build notification



Next, take a look at the Bamboo [build details](#), as displayed in an Eclipse editor.

RELATED TOPICS

Working with Bamboo Builds in Eclipse

- Monitoring Build Statuses in Eclipse
- Viewing Build Details in Eclipse
- Running a Build from within Eclipse
- Viewing Test Results in Eclipse
- Viewing a Build Log in Eclipse
- Adding a Task from a Failed Build in Eclipse

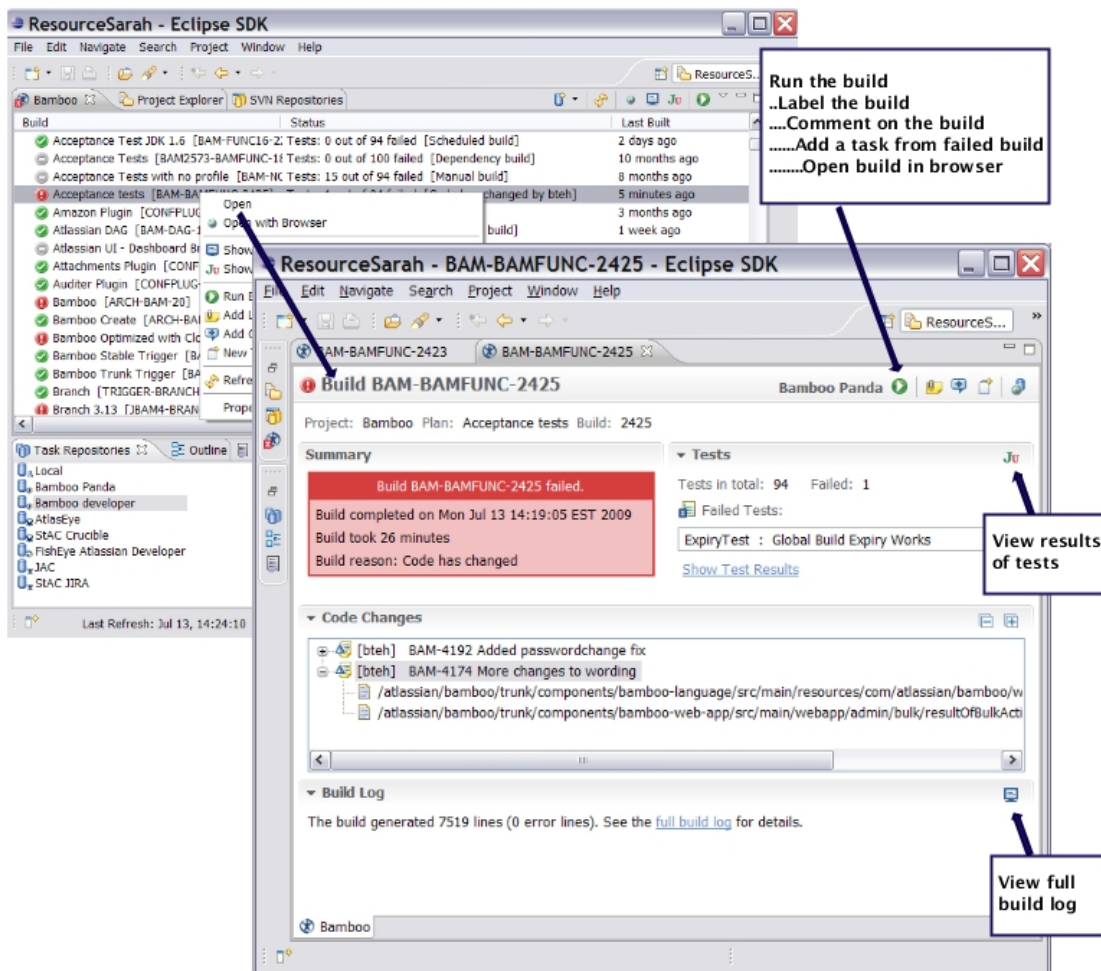
Viewing Build Details in Eclipse

Once you have opened the [Bamboo view in Eclipse](#), you can see the details of a specific build.

To view the details of a Bamboo build,

1. Open the Bamboo view:
 - In the Eclipse toolbar, select 'Window', 'Show View', 'Other'.
 - The 'Show View' popup window appears. Open the 'Atlassian' folder and click 'Bamboo'. (If you have trouble finding the options, try the [full instructions with screenshots](#).)
2. Double-click the line showing the build you want, or right-click the line and select 'Open'.
3. The build details appear in an Eclipse editor, as shown in the screenshot below.

Screenshot: Viewing Bamboo build details



The Bamboo build details screen opens in an Eclipse editor. It displays the following:

- The build summary.
- The number of tests in the build, with details of the failed tests. See [Viewing Test Results in Eclipse](#).
- The code changes included in the build.
- A summarised build log, with the option to see the full log. See [Viewing a Build Log in Eclipse](#).
- Further options to:
 - Run the build on the Bamboo server. See [Running a Build from within Eclipse](#).
 - Add a label to the build. See the [diagram above](#).
 - Add a comment to the build. See the [diagram above](#).
 - Add a new task based on the failed build. See [Adding a Task from a Failed Build in Eclipse](#).
 - Open the build in your browser, displaying Bamboo's web interface. See the [diagram above](#).

Next, take a look at [viewing test results](#).

RELATED TOPICS



Working with Bamboo Builds in Eclipse

- [Monitoring Build Statuses in Eclipse](#)
- [Viewing Build Details in Eclipse](#)
- [Running a Build from within Eclipse](#)
- [Viewing Test Results in Eclipse](#)
- [Viewing a Build Log in Eclipse](#)
- [Adding a Task from a Failed Build in Eclipse](#)

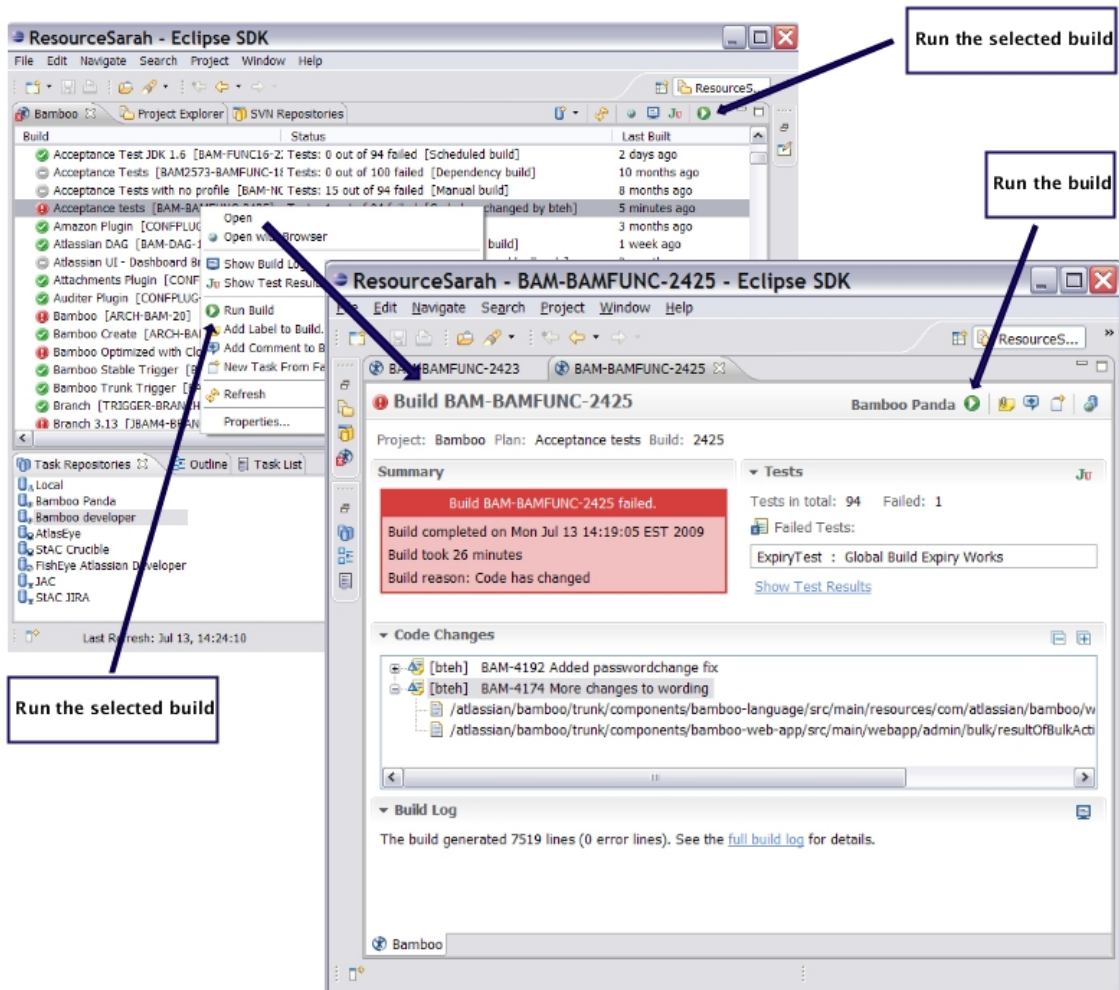
Running a Build from within Eclipse

Once you have opened the [Bamboo view in Eclipse](#), you can re-run a specific build. This will kick off the build on the Bamboo server.

To run a Bamboo build,

1. Open the Bamboo view:
 - In the Eclipse toolbar, select **'Window', 'Show View', 'Other'**.
 - The 'Show View' popup window appears. Open the **'Atlassian'** folder and click **'Bamboo'**. (If you have trouble finding the options, try the [full instructions with screenshots](#).)
2. Choose one of the following ways to run the build:
 - From the Bamboo view:
 - a. Right-click the line showing the build you want, and select **'Run Build'**.
 - Or, also from the Bamboo view:
 - a. Click the **'Run Build on Server'** icon  in the view's toolbar.
 - Or open the Bamboo build details screen:
 - a. Double-click or right-click the line showing the build you want, on the Bamboo view, to open the Bamboo build details.
 - b. The Bamboo build details are displayed in an Eclipse editor. Click the **'Run Build on Server'** icon  in the toolbar.

Screenshot: Various ways to run a Bamboo build



Next, take a look at viewing test results.

RELATED TOPICS

Working with Bamboo Builds in Eclipse

- Monitoring Build Statuses in Eclipse
- Viewing Build Details in Eclipse
- Running a Build from within Eclipse
- Viewing Test Results in Eclipse
- Viewing a Build Log in Eclipse
- Adding a Task from a Failed Build in Eclipse

Viewing Test Results in Eclipse

Once you have opened the Bamboo view in Eclipse, you can view the results of the tests in a specific build.

To view test results,



1. Open the Bamboo view:
 - In the Eclipse toolbar, select 'Window', 'Show View', 'Other'.
 - The 'Show View' popup window appears. Open the 'Atlassian' folder and click 'Bamboo'. (If you have trouble finding the options, try the [full instructions with screenshots](#).)
2. Choose one of the following ways to view the test results — all methods are illustrated in [diagram 1](#) below:
 - From the Bamboo view:
 - a. Right-click the line showing the build you want, and select 'Show Test Results'.
 - Or, also from the Bamboo view:
 - a. Select the line showing the build you want.
 - b. Click the JUnit icon  in the view's toolbar.
 - Or open the Bamboo build details screen:
 - a. Double-click or right-click the line showing the build you want, on the Bamboo view, to open the Bamboo build details.
 - b. The Bamboo build details are displayed in an Eclipse editor. Click the JUnit icon  in the toolbar.
3. The JUnit view opens, as shown in [diagram 2](#) below. Now you can use the JUnit functionality to:
 - Select a file and open it.
 - Re-run a test.
 - Debug the code.
 - Copy the failure trace information to your clipboard.

Diagram 1: Various ways to view the results of tests in a Bamboo build

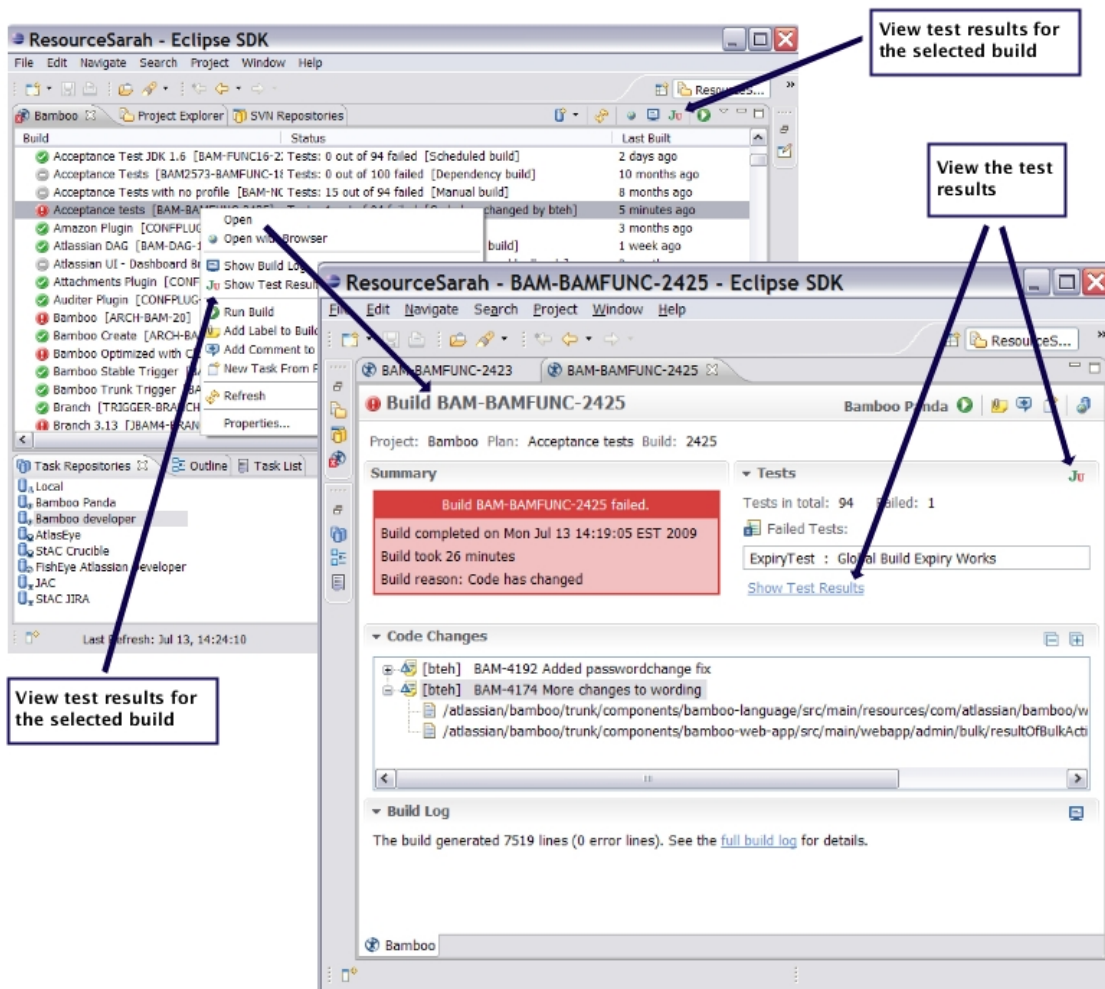
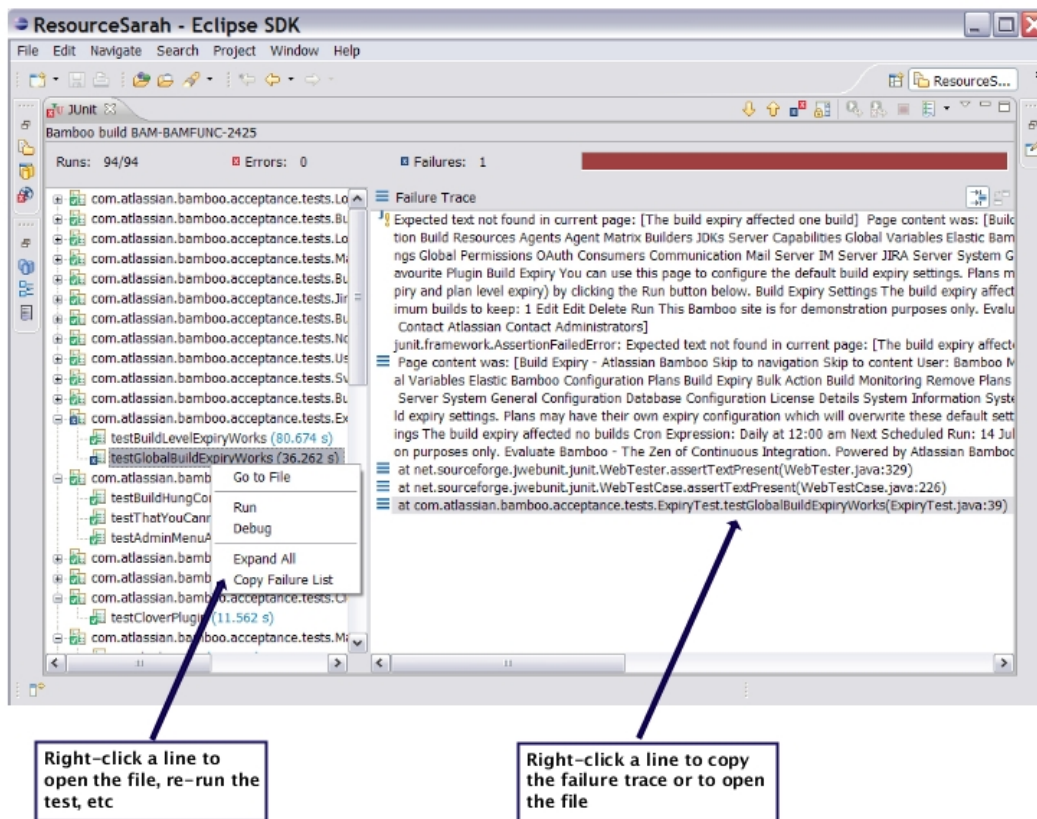


Diagram 2: Test results displayed in JUnit



Next, take a look at viewing a build log.

RELATED TOPICS

Working with Bamboo Builds in Eclipse

- Monitoring Build Statuses in Eclipse
- Viewing Build Details in Eclipse
- Running a Build from within Eclipse
- Viewing Test Results in Eclipse
- Viewing a Build Log in Eclipse
- Adding a Task from a Failed Build in Eclipse

Viewing a Build Log in Eclipse

Once you have opened the Bamboo view in Eclipse, you can view the full log for a specific build.

To view a full build log,



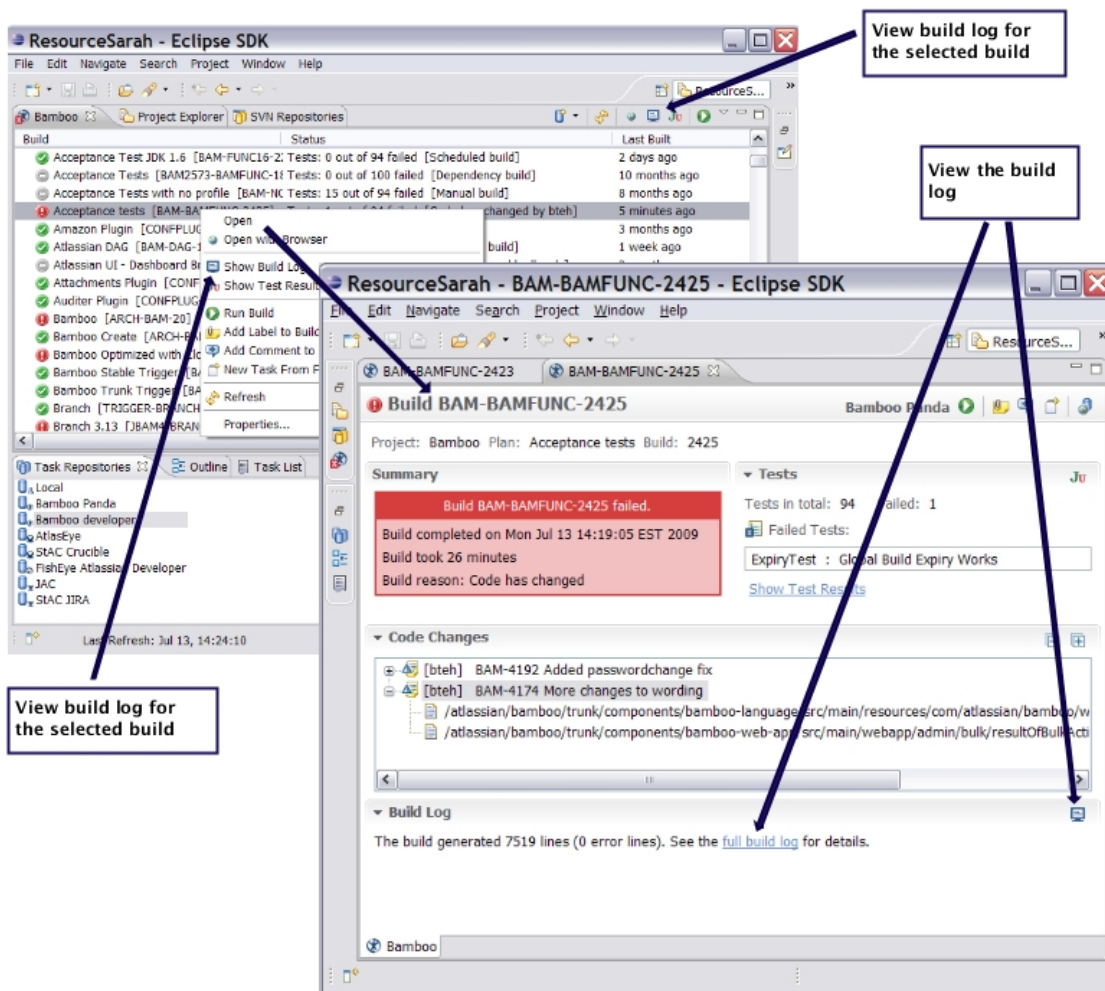
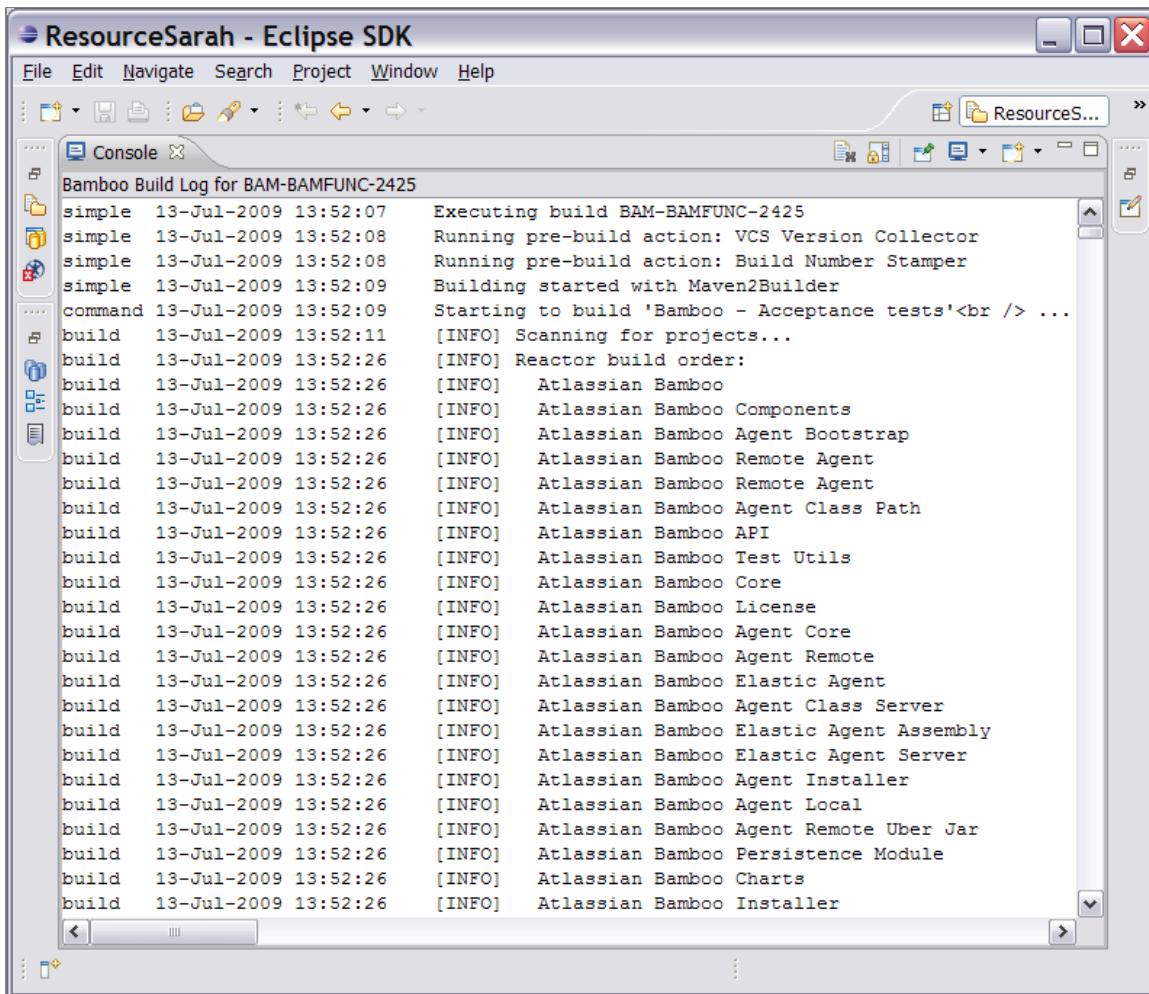
1. Open the Bamboo view:
 - In the Eclipse toolbar, select **'Window', 'Show View', 'Other'**.
 - The 'Show View' popup window appears. Open the **'Atlassian'** folder and click **'Bamboo'**. (If you have trouble finding the options, try the [full instructions with screenshots](#).)
2. Choose one of the following ways to view the build log — all methods are illustrated in [diagram 1](#) below:
 - From the Bamboo view:
 - a. Right-click the line showing the build you want, and select **'Show Build Log'**.
 - Or, also from the Bamboo view:
 - a. Select the line showing the build you want.
 - b. Click the **Show Build Log** icon  in the view's toolbar.
 - Or open the Bamboo build details screen:
 - a. Double-click or right-click the line showing the build you want, on the Bamboo view, to open the Bamboo build details screen.
 - b. The Bamboo build details are displayed in an Eclipse editor. Click the **Show Build Log** icon  or the text **'full build log'**.
3. The build log opens in the Eclipse console, as shown in [screenshot 2](#) below.

Diagram 1: Various ways to open a build log



Screenshot 2: Build log displayed in Eclipse console



Next, take a look at adding a task based on a failed build.

RELATED TOPICS

Working with Bamboo Builds in Eclipse

- [Monitoring Build Statuses in Eclipse](#)
- [Viewing Build Details in Eclipse](#)
- [Running a Build from within Eclipse](#)
- [Viewing Test Results in Eclipse](#)
- [Viewing a Build Log in Eclipse](#)
- [Adding a Task from a Failed Build in Eclipse](#)

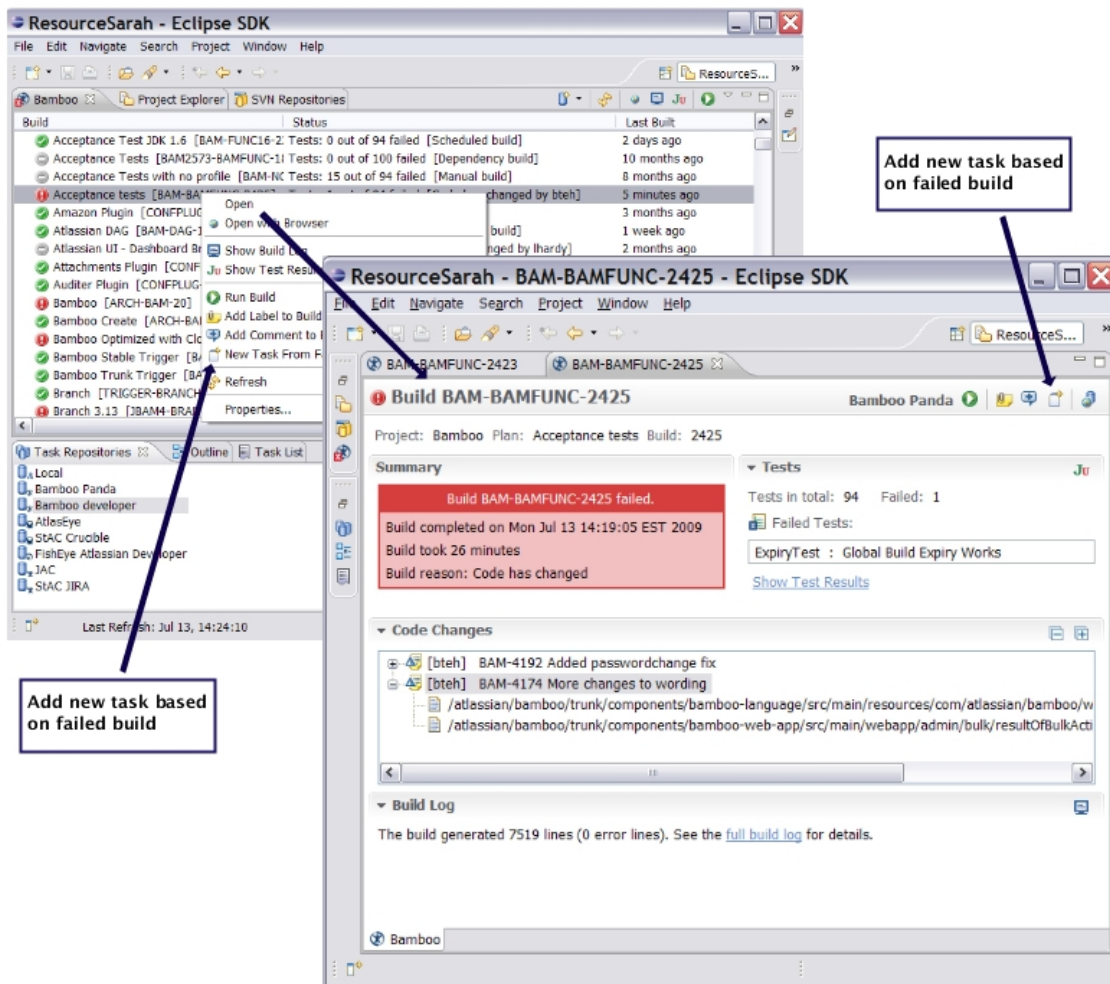
Adding a Task from a Failed Build in Eclipse

Once you have opened the [Bamboo view in Eclipse](#), you can create a new Mylyn task based on the failure of a specific build. This task could be an JIRA issue, for example.

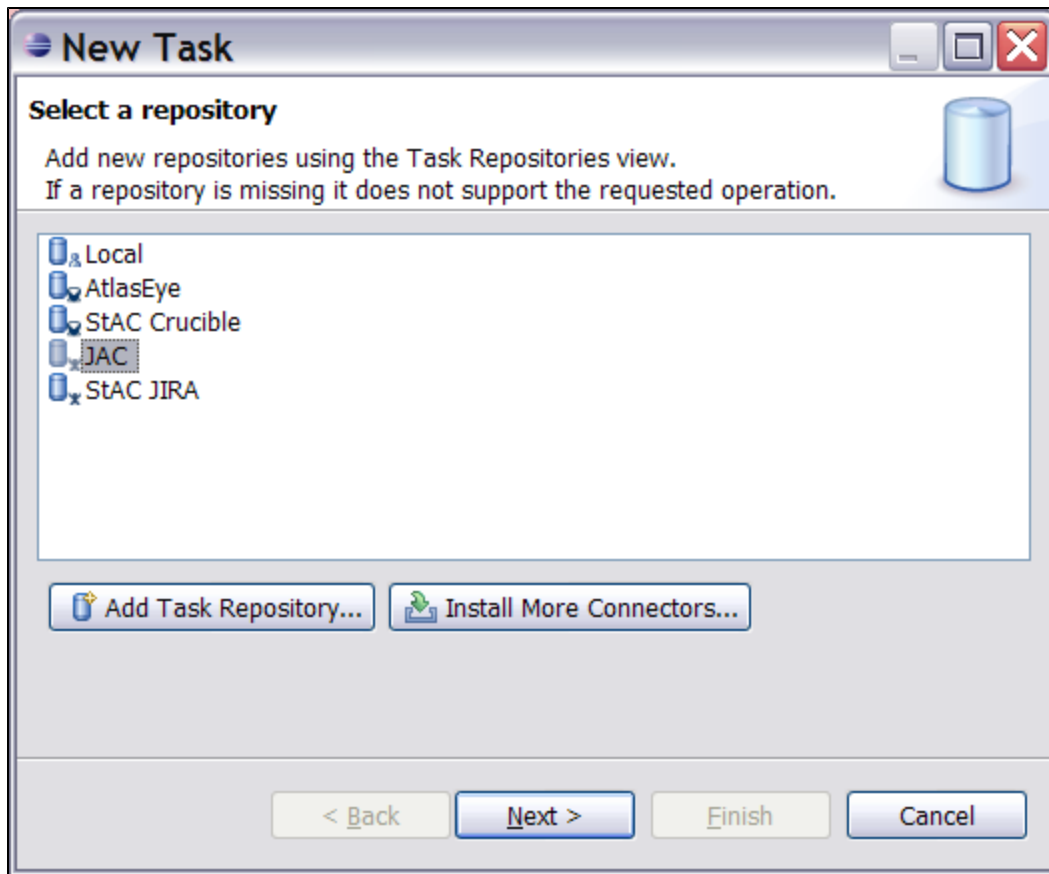
To add a task from a failed build,

1. Open the Bamboo view:
 - In the Eclipse toolbar, select **'Window', 'Show View', 'Other'**.
 - The 'Show View' popup window appears. Open the **'Atlassian'** folder and click **'Bamboo'**. (If you have trouble finding the options, try the [full instructions with screenshots](#).)
2. Choose one of the following ways to add a task — all methods are illustrated in [diagram 1](#) below:
 - From the Bamboo view:
 - a. Right-click the line showing the build you want, and select **'New Task From Failed Build'**.
 - Or open the Bamboo build details screen:
 - a. Double-click or right-click the line showing the build you want, on the Bamboo view, to open the Bamboo build details screen.
 - b. The Bamboo build details are displayed in an Eclipse editor. Click the **New Task From Failed Build** icon in the toolbar.
3. The **'New Task'** dialogue opens. Choose a task repository, to determine the task type. This task can be a JIRA issue, for example, as illustrated below. We have chosen a JIRA repository, as shown in [screenshot 2](#).
4. Click **'Next'**.
5. The **'New JIRA Task'** dialogue opens, as shown in [screenshot 3](#) below. Select the project for your task/issue.
6. Click **'Finish'**.
7. The JIRA issue opens in the Eclipse editor, as shown in [screenshot 4](#) below. Many of the issue details are already present, based on the details of the failed build. Enter or update any further details as required.
8. Click **'Submit'**.

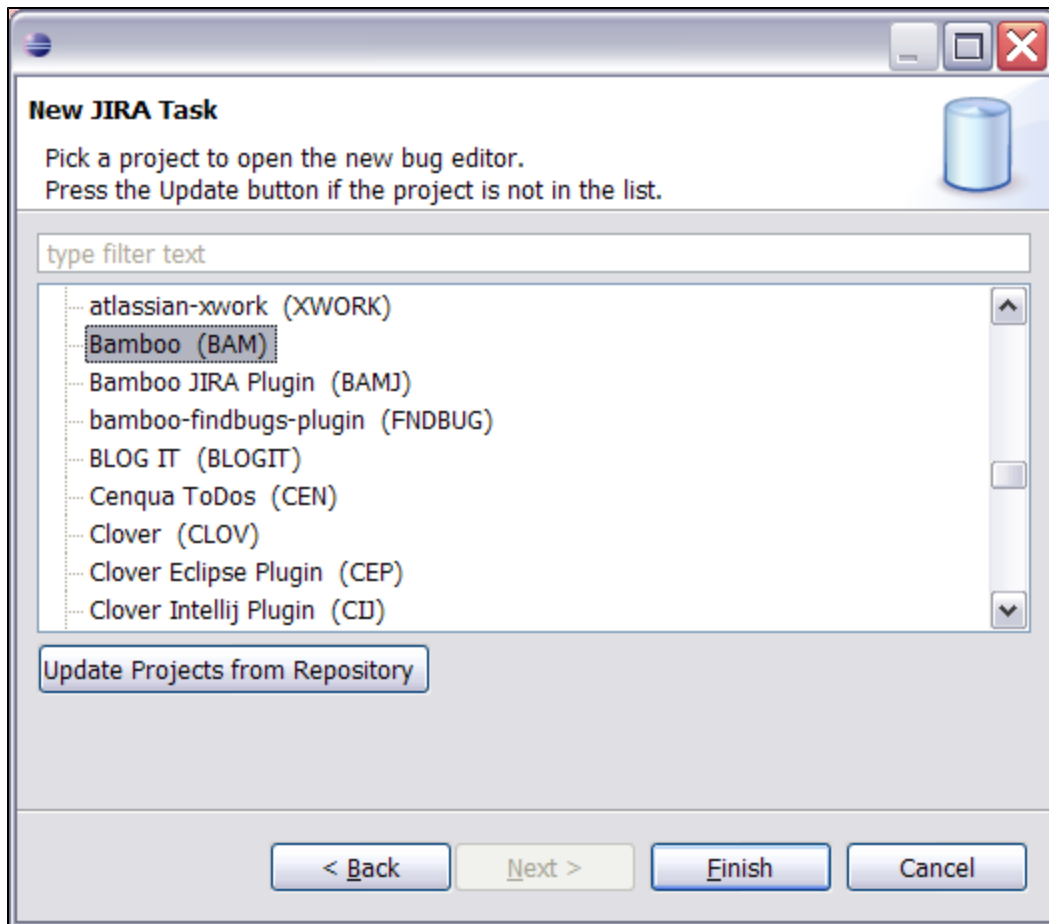
Diagram 1: Various ways to add a task from a failed Bamboo build



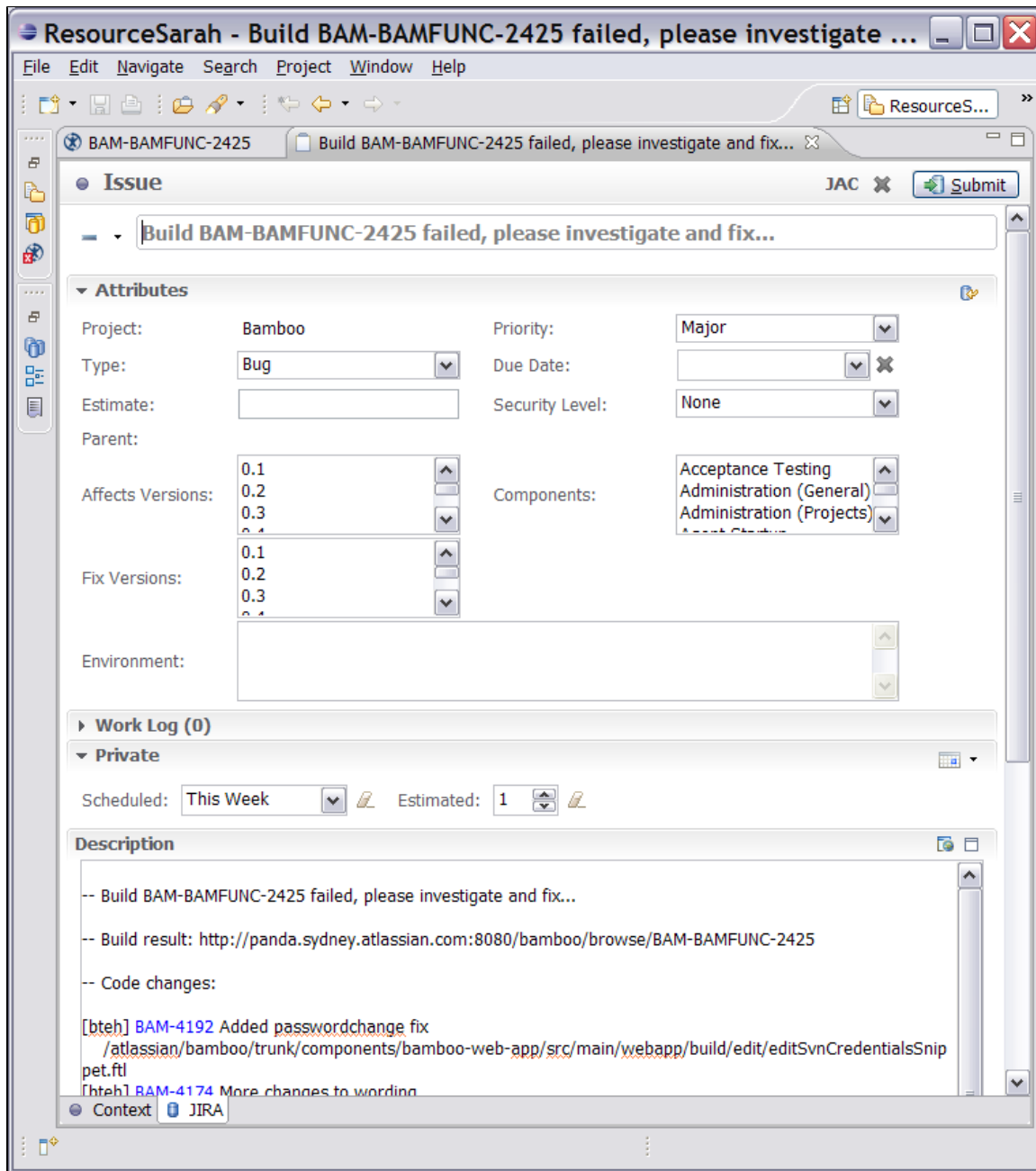
Screenshot 2: Selecting a task repository for the new task



Screenshot 3: Selecting a project for the new task



Screenshot 4: Adding issue details for the new task



RELATED TOPICS

Working with Bamboo Builds in Eclipse

- [Monitoring Build Statuses in Eclipse](#)
- [Viewing Build Details in Eclipse](#)
- [Running a Build from within Eclipse](#)
- [Viewing Test Results in Eclipse](#)
- [Viewing a Build Log in Eclipse](#)
- [Adding a Task from a Failed Build in Eclipse](#)

Using Crucible in the Eclipse Connector

The Atlassian Connector for Eclipse allows you to conduct [Crucible](#) reviews right there in your integrated development environment.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

- [Configuring your Crucible Options in Eclipse](#)
- [Working with Crucible Reviews in Eclipse](#)
 - [Monitoring Review Updates in Eclipse](#)
 - [Viewing a Review in Eclipse](#)
 - [Managing Review Comments in Eclipse](#)
 - [Working on a Review in the Diff View in Eclipse](#)
 - [Creating a Post-Commit Review from within Eclipse](#)
 - [Creating a Patch Review from within Eclipse](#)
 - [Assigning Reviewers in Eclipse](#)
 - [Adding Changesets or Patches to Review in Eclipse](#)

RELATED TOPICS

[Installation and Upgrade Guide for the Eclipse Connector](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Configuring your Crucible Options in Eclipse

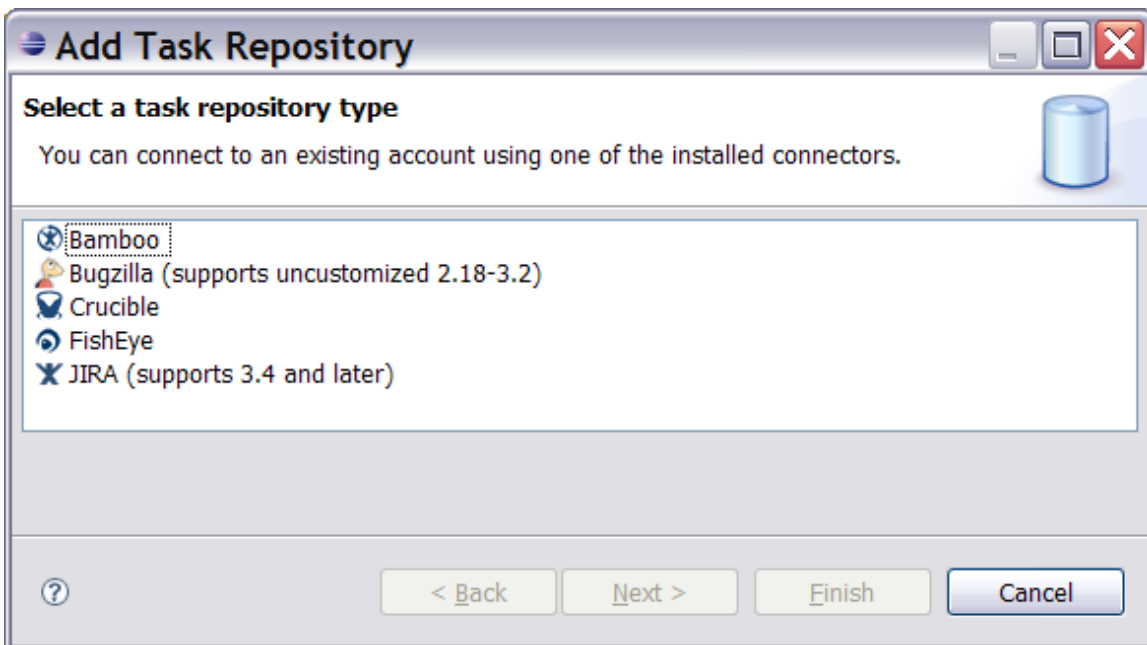
Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

On this page:

- [Adding a Task Repository](#)
- [Setting Up a Crucible Task Repository](#)

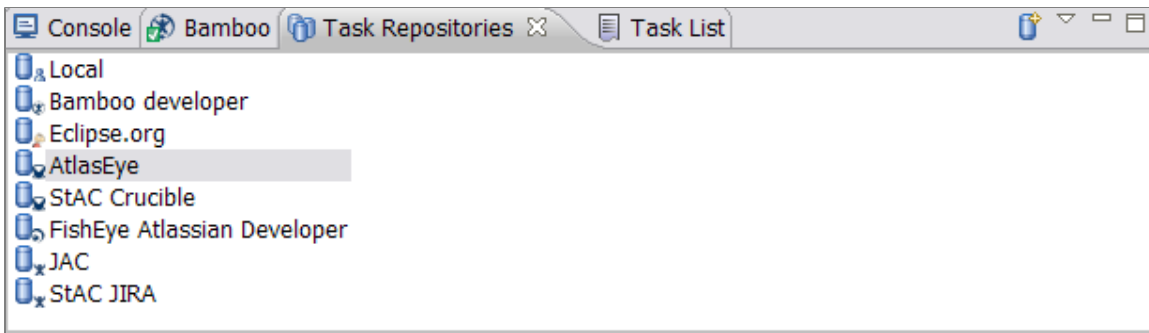
Adding a Task Repository

1. In Eclipse Mylyn, open the **'Task Repositories'** view:
 - Click **'Window', 'Show View', 'Other'**.
 - Open the **'Tasks'** category and select the **'Task Repositories'** view.
2. Click the **'Add Task Repository'** icon.
3. The 'Add Task Repository' screen appears, as shown below:



4. Now you can select and configure one or more of the task repositories and configure them to match your server(s), as described below.

Once you have set up a repository, it will appear in the Eclipse **'Task Repositories'** view, as shown in this screenshot:



Setting Up a Crucible Task Repository





Single server for both Crucible and FishEye = single task repository

If you have a single server for both FishEye and Crucible, you should set up the Crucible task repository and then select the '**Crucible Server Contains FishEye Instance**' option as described below. Due to Mylyn requirements, you cannot set up two separate task repositories for the same URL.

1. On the Eclipse 'Add Task Repository' screen, select the '**Crucible**' task repository type and click '**Next**'.
2. The 'Crucible Repository Settings' screen appears, as shown below:

3. Enter the following information:
 - **Server** — The location (URL) of your Crucible server.
 - **Label** — A descriptive name for your Crucible server, e.g. 'Extranet Crucible'.
 - **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your

- servers are behind a firewall and you do not have access to them at a particular point in time.
- **User ID** — The username you use to connect to your Crucible server.
 - **Password** — Your password on the Crucible server, matching the above username.
 - **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.
-  If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.
4. Click the **'Validate Settings'** button to verify the information you have entered.
 5. Choose a **'Review Activation'** setting. These settings determine what happens when you open a source code file from within the Crucible review editor in Eclipse. For example, you may open a Crucible review within Eclipse. The review will open in an Eclipse editor view, and its 'Review Files' section will show the files in the review. You can then click a file to open it in an Eclipse editor. The options are as follows:
 - **'Always'** — The Crucible review will be automatically activated if you open a file from the Crucible review editor.
 - **'Never'** — The Crucible review will never be automatically activated.
 - **'Prompt'** (default) — When you open a file from the Crucible review editor, the connector will ask you whether you want to activate the Crucible review. This is the default setting.
 6. If your Crucible server is linked to a FishEye server:
 - Put a tick in the checkbox labelled **'Crucible Server Contains FishEye Instance'**.
 - Set up your FishEye mappings, as described in the FishEye section of this documentation.
 7. Click **'Finish'** to save the changes.
 8. Eclipse Mylyn will prompt you to add a new query for the new Crucible repository. This is where you will choose your Crucible filter, to determine which reviews appear in your task list. You can do this now or skip this step and do it later. You can also add more queries later.
 9. The new task repository appears in your Eclipse **'Task Repositories'** view. Your Crucible reviews will appear in your 'Task List' view.

 You can configure one or more Crucible repositories, i.e. you can connect to more than one Crucible server.

RELATED TOPICS

[Working with Crucible Reviews in Eclipse](#)
[Installation and Upgrade Guide for the Eclipse Connector](#)

Working with Crucible Reviews in Eclipse

The Atlassian Connector for Eclipse allows you to conduct [Crucible](#) reviews right there in your integrated development environment. Your Crucible reviews will appear in your Eclipse 'Task List' view.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [installation guide](#), and defined a task repository for at least one Crucible server, as described in the [configuration guide](#).

Summary of What You Can Do

Below are the highlights of viewing and acting upon Crucible reviews within Eclipse. Follow the links to the relevant sections of the user guide.

- Add a query to [retrieve and filter your reviews](#) in Eclipse
- View a list of your reviews in the Eclipse [task list](#)
- Receive [notification](#) of new and updated reviews
- View the [review details](#) in an Eclipse editor
- Add, edit and reply to [review comments](#) in the review view or in the code editor
- [Open the file under review](#), at the commented source code line
- Create a [post-commit review](#) or a [patch review](#)
- [Assign reviewers](#)
- Add [changesets](#) or [patches](#) to an existing review
- Summarise and close the review and perform other [workflow actions](#)
- Work on a review in the [diff view](#) too

Adding a Query to Retrieve and Filter your Reviews

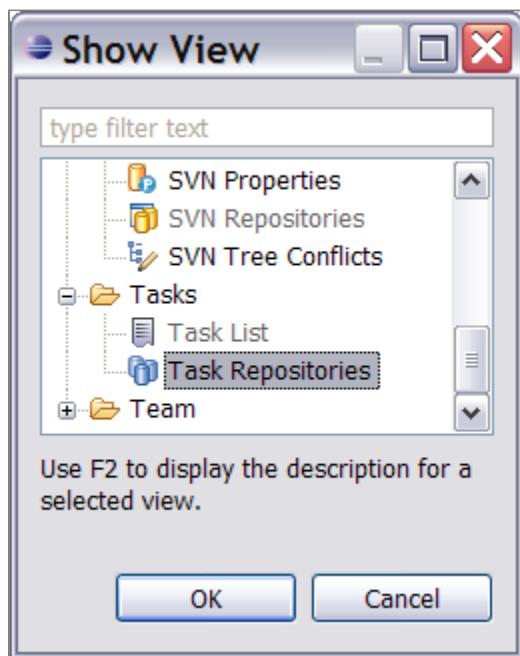
To retrieve your Crucible reviews and display them in your Eclipse task list, you will create an Eclipse Mylyn query. The query allows you to select an existing Crucible filter or define a custom filter.

When you [define a task repository for your Crucible server](#), Eclipse Mylyn will prompt you to add a query for the new Crucible repository. You can also add more queries later, as described below.

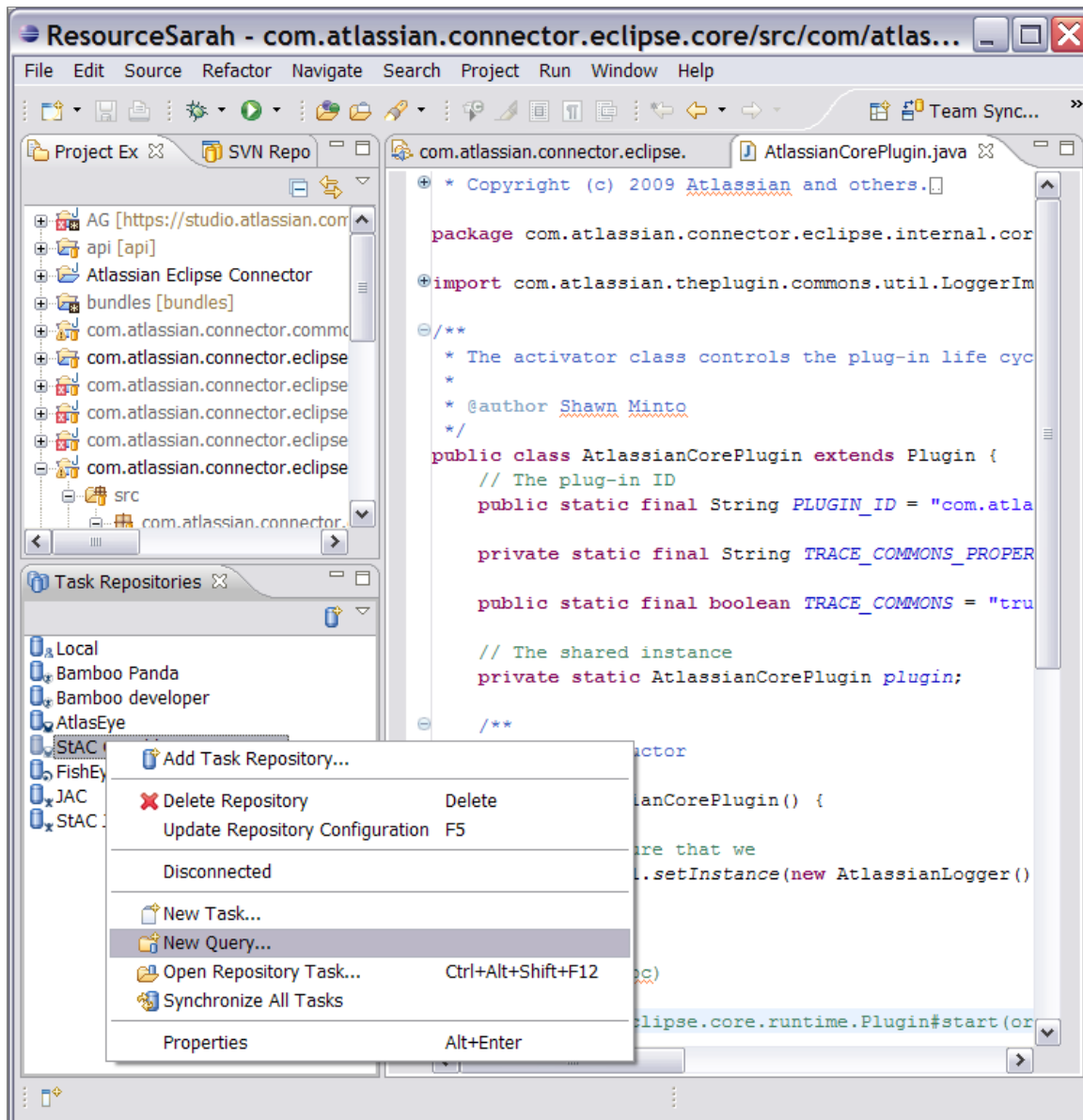
[To add a query for filtering your Crucible reviews,](#)

1. If you have not already set up a task repository for your Crucible server, please do so by following the steps in the [configuration guide](#)
2. Open your Eclipse 'Task Repositories' view and find the task repository that you defined for your Crucible server:
 - In the Eclipse toolbar, select **'Window', 'Show View', 'Other'**.
 - The 'Show View' popup window appears, as shown in [screenshot 1](#) below. Open the **'Tasks'** folder and click **'Task Repositories'**.
 - The 'Task Repositories' view appears, as shown in [screenshot 2](#) below. Right-click the task repository that you defined for your Crucible server and select **'New Query'**.
3. The 'Edit Repository Query' window appears, as shown in [screenshot 3](#) below. Now you can choose to define a custom filter or use a predefined filter that exists on your Crucible server.
 - To define a custom filter:
 - a. Select the **'Custom'** radio button.
 - b. Click **'Next'**.
 - c. Enter a title for your query, as prompted. Select the review field values that you want to filter on. For example, see [screenshot 4](#) below.
 - When selecting the review **'State'**, you can select more than one value by holding down the Ctrl key.
 - d. Click **'Finish'**.
 - To select an existing filter from your Crucible server:
 - a. Select the **'Predefined Filter'** radio button.
 - b. Click one of the filters offered below the radio button, as shown in [screenshot 3](#) below.
 - c. Click **'Finish'**.
4. Your Crucible reviews will appear in your 'Task List' view, as described [below](#).

Screenshot 1: Opening the 'Task Repositories' view



Screenshot 2: Adding a new query



Screenshot 3: Selecting a predefined review filter

Edit Repository Query

Enter query parameters


Enter a title

Title:

☐ Custom

☒ Predefined Filter

- To Review
- Require My Approval
- To Summarize
- Out For Review
- Drafts
- Open
- Closed
- Abandoned
- Open in IDE



Screenshot 4: Defining a custom review filter

Edit Repository Query

Enter query parameters
Select options to create a query

Title: All open Crucible reviews in Connector project

Project: Atlassian Eclipse Connector

Roles

Author: <Any>

Moderator: <Any>

Creator: <Any>

Reviewer: <Any>

Reviewer Status: <Any>

☒ Match All Roles
☐ Match Any Roles

State: Rejected
Review Needs Fixing
Summarize
Under Review

Refresh From Repository

? < Back Next > Finish Cancel

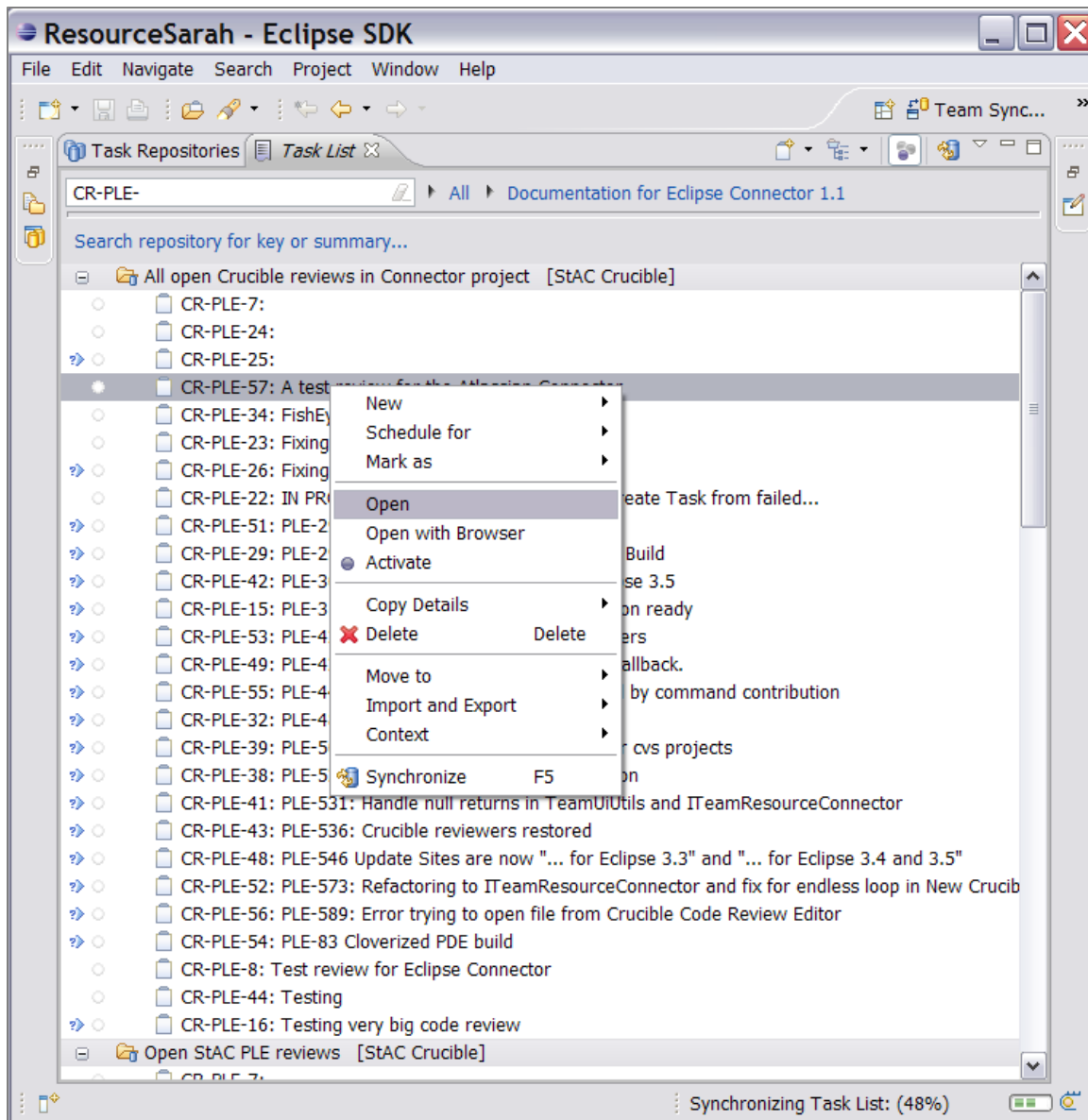
Viewing your Reviews in the Eclipse Task List

Crucible reviews are included in your Eclipse 'Task List' view. You can take advantage of Mylyn's context management to keep track of the files and other assets associated with your review.

To open your Eclipse 'Task List' view,

1. In the Eclipse toolbar, select 'Window', 'Show View', 'Task List'.
2. The 'Task List' view appears, as shown in [screenshot 5](#) below. Open the folder with the name corresponding to the query, i.e. the Crucible filter, that you created earlier.

Screenshot 5: Viewing your Crucible reviews in an Eclipse task list



Next, take a look at how you can [monitor review updates in Eclipse](#).

RELATED TOPICS

- [Installation and Upgrade Guide for the Eclipse Connector](#)
- [Configuring the Eclipse Connector](#)
- [Using Bamboo in the Eclipse Connector](#)
- [Using Crucible in the Eclipse Connector](#)
- [Using FishEye in the Eclipse Connector](#)
- [Using JIRA in the Eclipse Connector](#)
- [Eclipse Connector FAQ](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

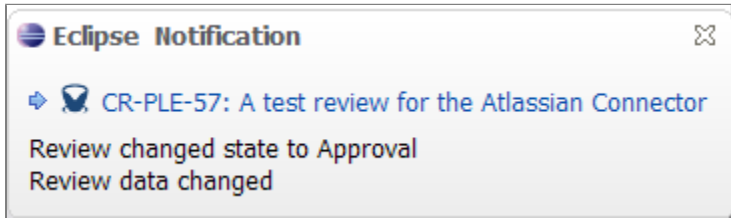
Monitoring Review Updates in Eclipse

You will receive an Eclipse notification whenever one of the following occurs:

- Someone assigns you as a reviewer.
- There is a change to a review that is included in your Eclipse task list, retrieved via the queries that you have set up. (See the guide to [creating a query](#).)

Here is an example of the Eclipse notification popup window.

Screenshot: Crucible review notification



Next, take a look at the [Crucible review details](#), as displayed in an Eclipse editor.

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)


Viewing a Review in Eclipse

Once you have [set up your queries](#) to retrieve and filter your Crucible reviews, you can open a review and see its details without leaving Eclipse. You can open a review from the Eclipse task list or from the code editor.

To open a Crucible review from the Eclipse task list,

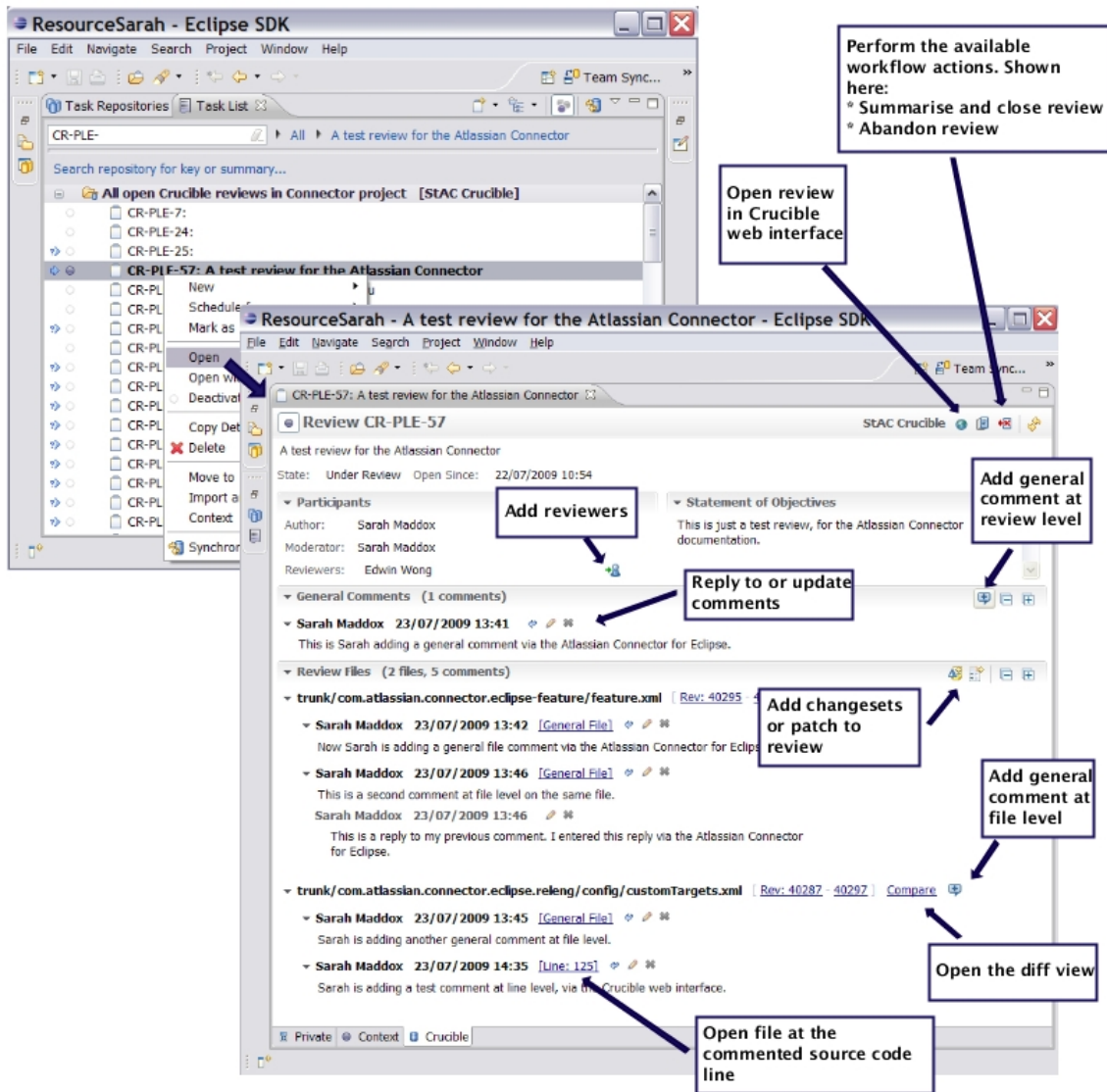
1. If you have not already done so, define your Crucible filters as described in the guide to [creating a query](#).
2. Open your Eclipse 'Task List' view: In the Eclipse toolbar, select '**Window**', '**Show View**', '**Task List**'.
3. Open the folder with the name corresponding to the query that you want, i.e. the Crucible filter that you created earlier.
4. Double-click the line showing the review you want, or right-click the line and select '**Open**'.
5. The review details appear in an Eclipse editor, as shown in the [diagram](#) below.

To open a Crucible review from the Eclipse code editor,

1. Make sure that the Crucible review task is **active** in Mylyn Eclipse. To activate the Mylyn task, right-click the review in your Eclipse Mylyn task list and select '**Activate**'.
2. Scroll down your code editor until you see a review comment marked by a Crucible icon  in the left-hand ruler.
3. Hover over the Crucible icon to see the comment text and other details.
4. Press F2 to bring the comment into focus.
5. The review comment box becomes an inline dialogue box. Click '**Open Review**' to open the review in the Eclipse review editor.

 If you have trouble finding the Crucible comments in code editor, take a look at the full [instructions with screenshots](#).

Diagram: Viewing the Crucible review details



The Crucible review details screen opens in an Eclipse editor. It displays the following:

- The review summary, state, date of opening and statement of objectives.
- The participants in the review, i.e. the author, moderator and reviewers.
- General comments — the comments at review level.
- The files included in the review, and the general comments at file level for each file.
- Comments at line level within the file.
- Further options to:
 - Open the review in your browser, displaying Crucible's web interface. See the [diagram above](#).
 - Manage comments at review and file level: Add, reply to, edit and remove comments. See [Managing Review Comments in Eclipse](#).
 - Open the file under review, at the commented source code line. See the [diagram above](#).
 - Perform the workflow actions that are relevant to the review's current state, such as summarise and close the review or abandon the review. See the [diagram above](#).
 - Add reviewers to and remove reviewers from the review. See [Assigning Reviewers in Eclipse](#).
 - Add changesets or a patch to the review. See [Adding Changesets or Patches to Review in Eclipse](#).
 - Open a diff view and conduct your review in the diff. See [Working on a Review in the Diff View in Eclipse](#).

Next, take a look at [managing your review comments](#).

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)

- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Managing Review Comments in Eclipse

Once you have set up your Crucible server as an [Eclipse task repository](#), you can perform a Crucible review without leaving Eclipse. The Atlassian Connector for Eclipse allows you to create and manage comments in a special Eclipse editor that contains the review, or directly in the Eclipse code editor. We describe both methods below. You can also manage your review comments in the diff view, as described in [another topic](#).



Can't see or add comments in your code editor?

If you want to work with your Crucible review in the Eclipse *code editor*, the review *task* must be **active** in Mylyn Eclipse. See [below](#) for instructions.

On this page:

- [Managing Comments in the Review Editor](#)
- [Viewing Comments in the Code Editor](#)
- [Adding, Updating and Replying to Comments in the Code Editor](#)

Managing Comments in the Review Editor

You can open your Crucible review in a special Eclipse editor and add, reply to or update review comments within the editor view.

[To manage your comments in the review editor,](#)

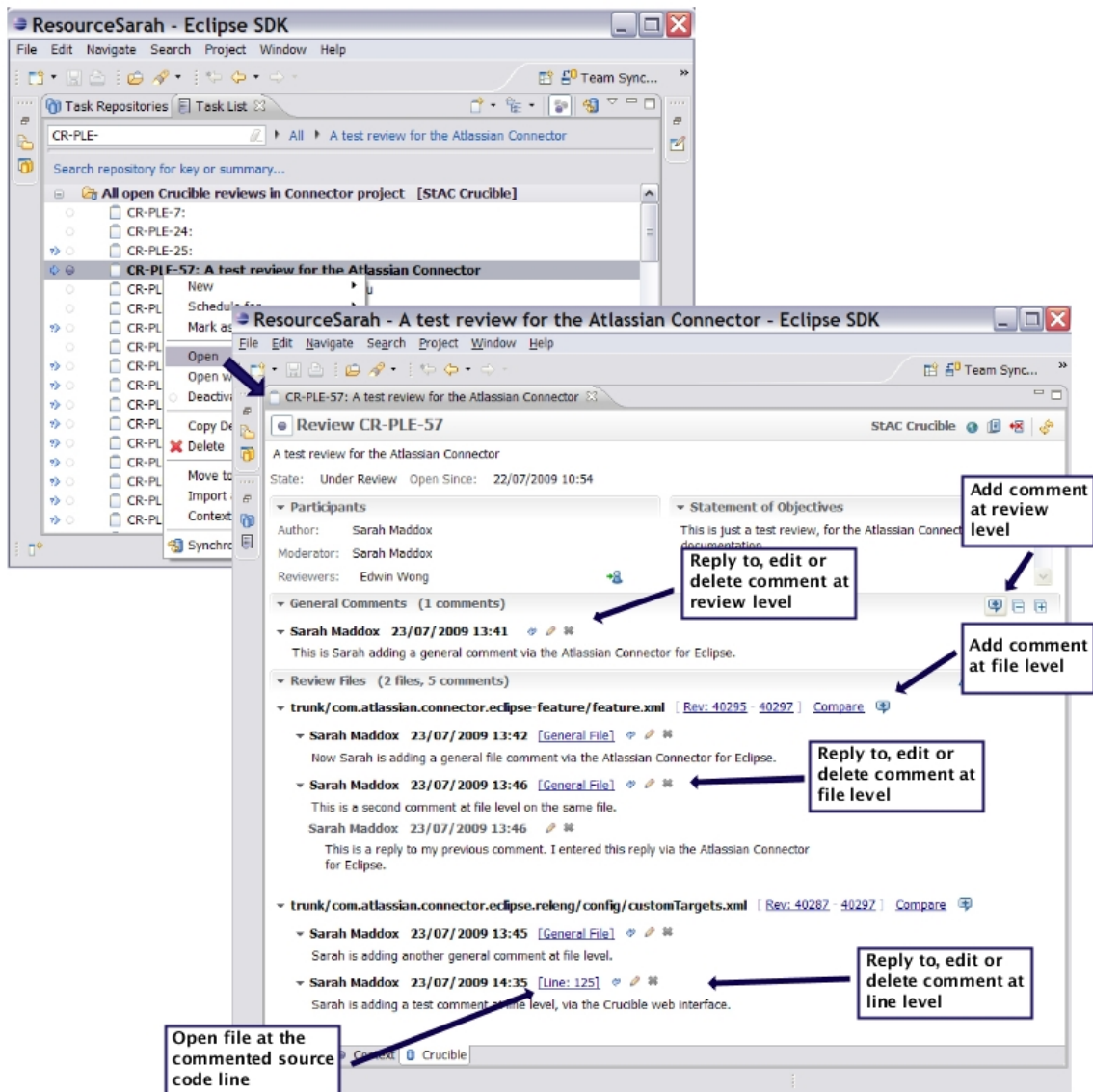
1. If you have not already done so, define your Crucible filters as described in the guide to [creating a query](#).
2. Open the review in an Eclipse editor:
 - Open your Eclipse 'Task List' view: In the Eclipse toolbar, select **'Window', 'Show View', 'Task List'**.
 - Open the folder with the name corresponding to the query that you want, i.e. the Crucible filter that you created earlier.
 - Double-click the line showing the review you want, or right-click the line and select **'Open'**.
3. The review details appear in an Eclipse editor. Now you can conduct your review in the Eclipse editor, by adding and responding to review comments as shown in [diagram 1](#) below:
 - Add, edit, delete and reply to comments applicable to the review as a whole. These are called **'general review comments'**.
 - Add, edit, delete and reply to comments applicable to a file. These are called **'general file comments'**.
 - Edit, delete and reply to comments applicable to a line or block of source code. These are called **'line comments'**.
 - To add a line comment, you need to open the source file in the Eclipse editor. See [below](#). You cannot add line comments from the review details editor.
 - Open the file under review, at the commented source code line. This will open the file in the Eclipse code editor, so that you can manage your comments there as described [below](#).



Working with code and review side by side

This is a handy way to work with the comments in the review editor while viewing the commented code in another open window: Open the source file in your code editor and the Crucible review in the review editor, with the two editor windows side by side. Click the text of a line comment in the review editor. The corresponding code lines will be highlighted in the code editor. You may find this easier than reading the code and hovering over the Crucible icon to see the comment.

[Diagram 1: Managing your review comments in the review editor](#)



Viewing Comments in the Code Editor

When working with your code in an Eclipse editor, you can see the Crucible review comments as well. See [diagram 2](#) below.


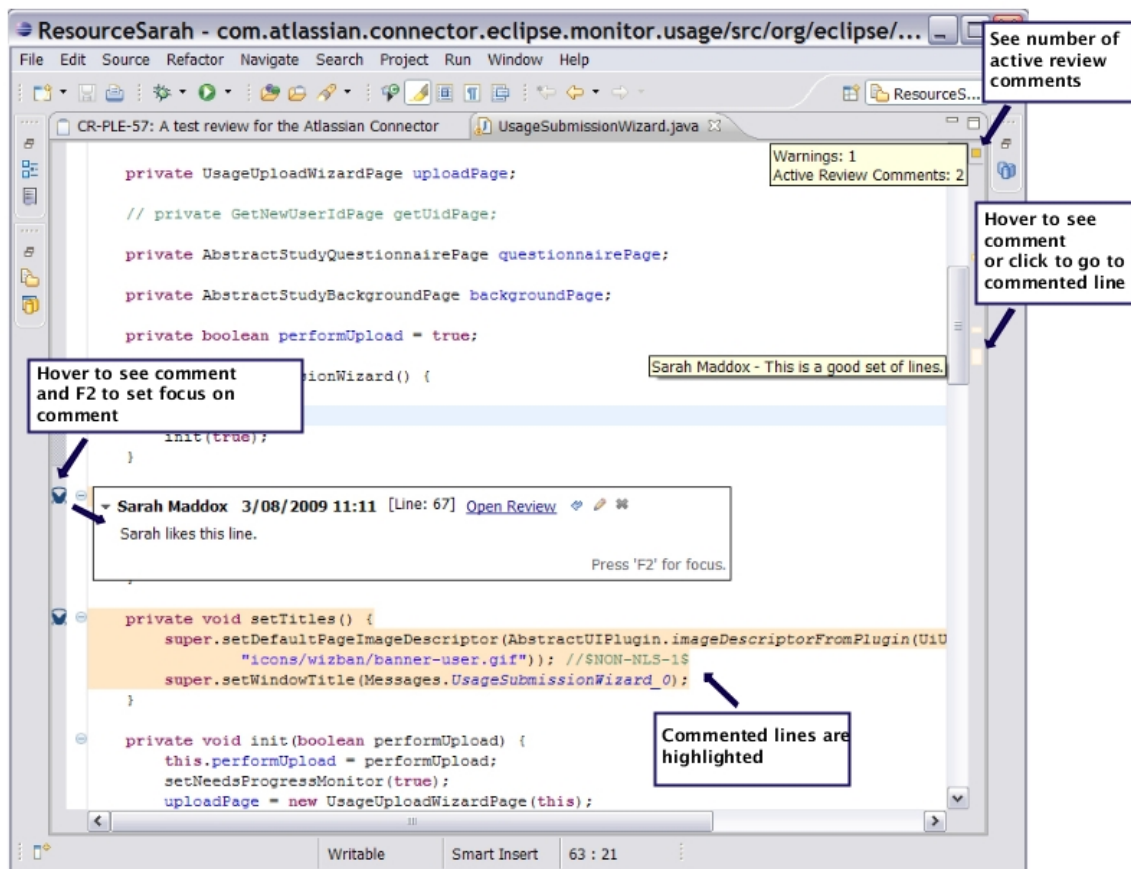
- Review comments are annotated with the Crucible icon  in the left-hand ruler of the Eclipse editor.
 - Hover over the Crucible icon to see the comment text and other details of the review comment.
 - Press F2 when viewing the comment details to bring the comment into focus, so that you can update or reply to the comment. See [below](#) for more details on how to conduct the review.
- The code lines to which the comment applies are highlighted in the editor body.
- An annotation at the top of the right-hand ruler shows, on hover, how many active review comments there are for the file.
- Annotations in the right-hand ruler show every line comment in the entire file, including the code sections that are not currently visible in your window.
 - Click an annotation to jump to the commented code line.
 - Hover over an annotation to see a tooltip containing the comment text.

Diagram 2: Viewing the review comments in the code editor



Adding, Updating and Replying to Comments in the Code Editor

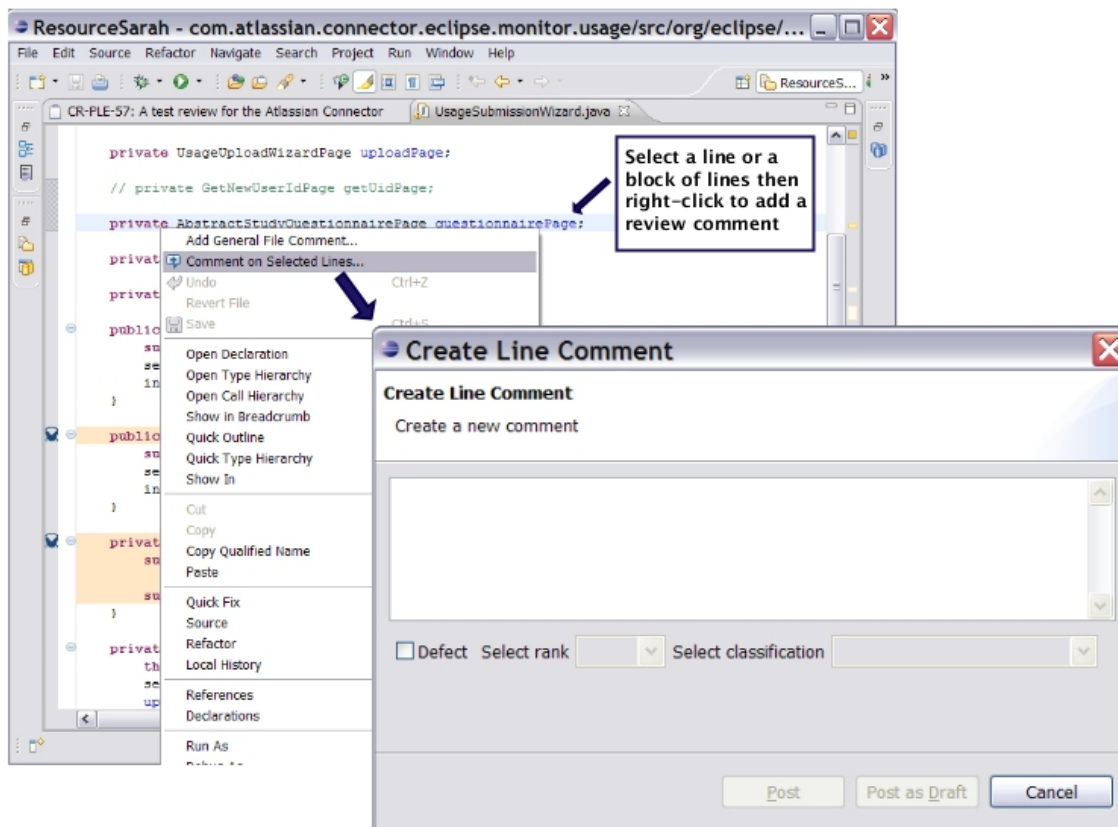
When viewing the source code in an Eclipse editor, you can:


- Add general file comments.
- Add, reply to and update line comments.

To add comments in the code editor,

1. Make sure that the Crucible review task is **active** in Mylyn Eclipse. To activate the Mylyn task, you can do one of the following:
 - Open the review in its own Eclipse view, as described [above](#).
 - Or right-click the review in your Eclipse Mylyn task list and select '**Activate**'.
2. Select a code line or a block of lines in the Eclipse code editor and right-click.
3. Select either '**Add General File Comment**' or '**Comment on Selected Lines**'. In [diagram 3](#) below, we show you how to add a line comment.
4. A popup window will ask you for the comment details. Enter the comment text, supply the other information if necessary, and '**Post**' the comment.

Diagram 3: Adding review comments in the code editor



Review comments are annotated with the Crucible icon  in the left-hand ruler of the Eclipse editor. Also, annotations in the right-hand ruler show every line comment in the entire file, including the code sections that are not currently visible in your window. See [above](#) for more details about viewing comments in the code editor.

To reply to and update line comments in the code editor,


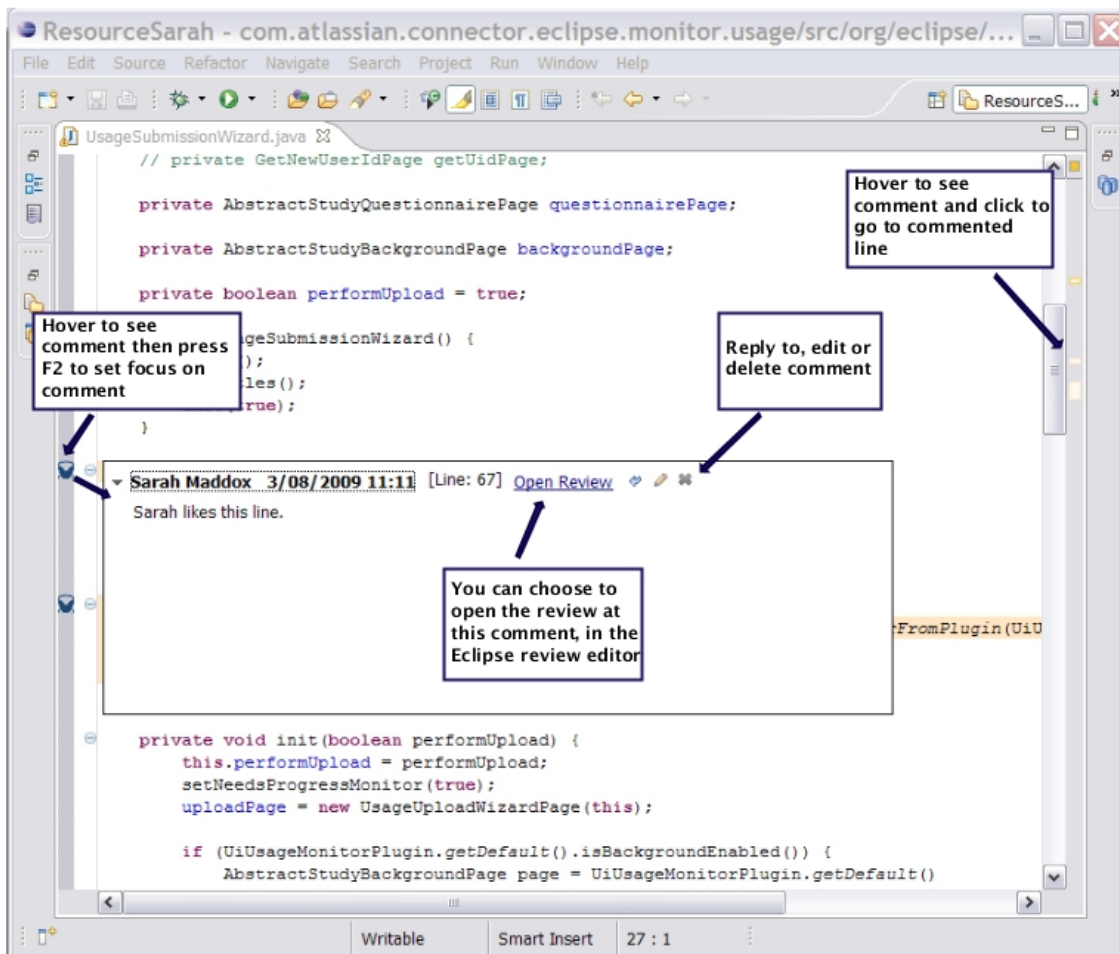
1. Make sure that the Crucible review task is **active** in Mylyn Eclipse. To activate the Mylyn task, you can do one of the following:
 - Open the review in its own Eclipse view, as described [above](#).
 - Or right-click the review in your Eclipse Mylyn task list and select '**Activate**'.
2. Find the comment that you want to reply to or update in the code editor:
 - Hover over the annotations in the right-hand ruler to see a tooltip containing the comment text.
 - Click the annotation in the right-hand ruler to jump to the comment you want.
3. Open the comment for editing or replying:
 - Hover over the Crucible icon  in the left-hand ruler to see the comment text.
 - Press F2 when viewing the comment details to bring the comment into focus.
4. The review comment box becomes an inline dialogue box. Now you can reply to the comment, edit it or delete it, depending on your permissions, as shown in [diagram 4](#) below. You can also choose to open the review in the Eclipse review editor instead. See [above](#) for instructions on managing comments in the review editor.

Diagram 4: Replying to and updating line comments in the code editor



Next, take a look at working on a review in the diff view.

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Working on a Review in the Diff View in Eclipse

Once you have set up your Crucible server as an [Eclipse task repository](#), you can perform a Crucible review without leaving Eclipse. The Atlassian Connector for Eclipse allows you to create and manage comments in your Eclipse code editor, as described in [another topic](#). You can also manage your review comments in the diff view, as described below.



Can't see comments in your code editor?

If you want to work with your Crucible review in the Eclipse *code editor*, the review *task* must be **active** in Mylyn Eclipse. See the instructions on [managing review comments](#).

To add review comments in a diff view,


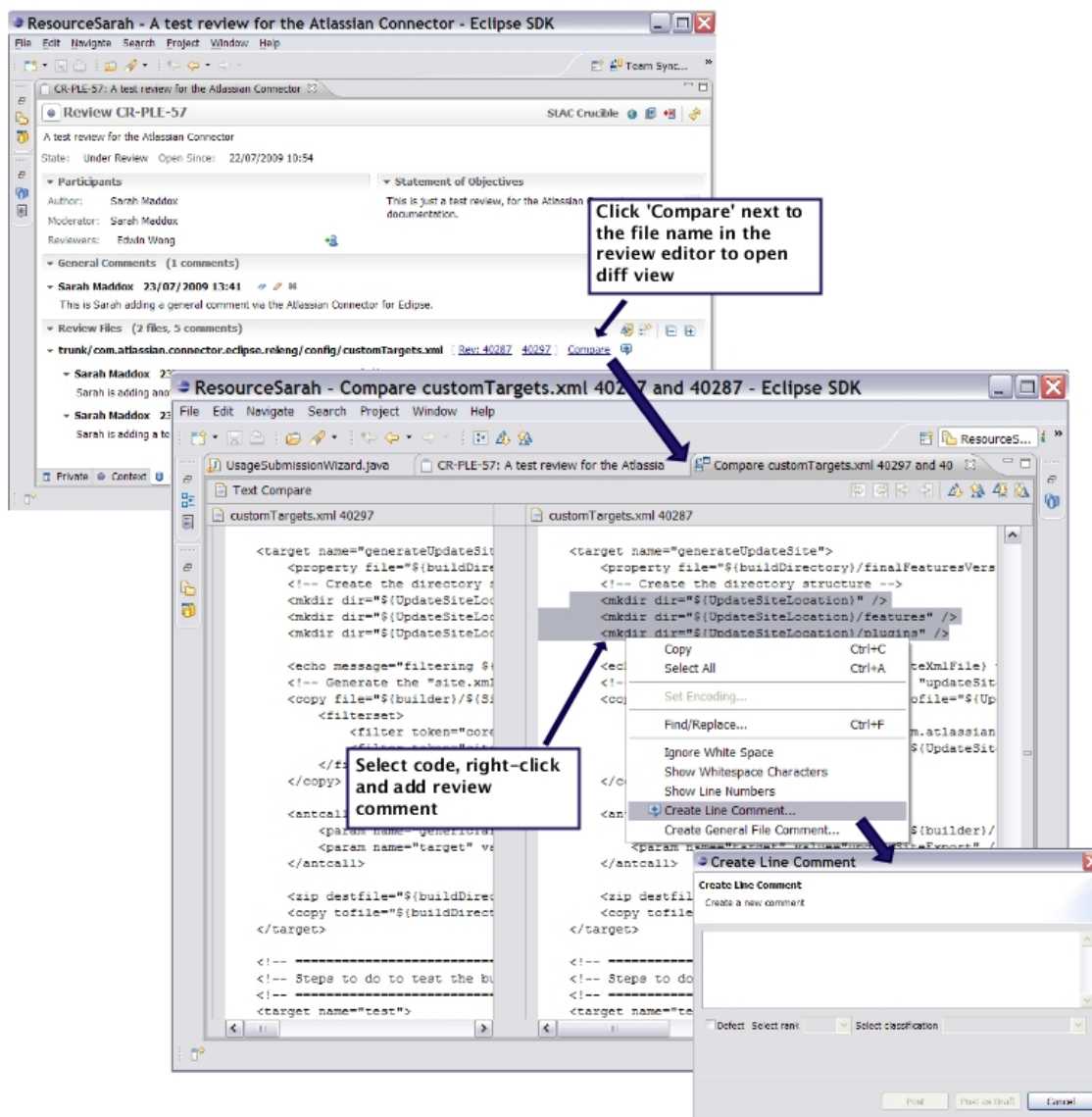
1. Open the Crucible review in an Eclipse review editor. Follow these instructions to open the review from your code editor, or see the complete instructions on [opening a review](#):
 - Make sure that the Crucible review task is **active** in Mylyn Eclipse. To activate the Mylyn task, right-click the review in your Eclipse Mylyn task list and select **'Activate'**.
 - Scroll down your code editor until you see a review comment marked by a Crucible icon  in the left-hand ruler.
 - Hover over the Crucible icon to see the comment text and other details.
 - Press F2 to bring the comment into focus.
 - The review comment box becomes an inline dialogue box. Click **'Open Review'** to open the review in the Eclipse review editor.
2. Open the diff view by clicking **'Compare'** next to the file you want, in the 'Review Files' section of the review editor as shown in [diagram 1](#) below.
3. The 'Compare' view opens, as shown in [diagram 1](#) below. Select a code line or a block of lines in the Eclipse code editor and right-click.
4. Select either **'Create General File Comment'** or **'Create Line Comment'**. In [diagram 1](#) below, we show you how to add a line comment.
5. A popup window will ask you for the comment details. Enter the comment text, supply the other information if necessary, and **'Post'** the comment.

Diagram 1: Adding review comments in a diff view



To reply to and update review comments in a diff view,



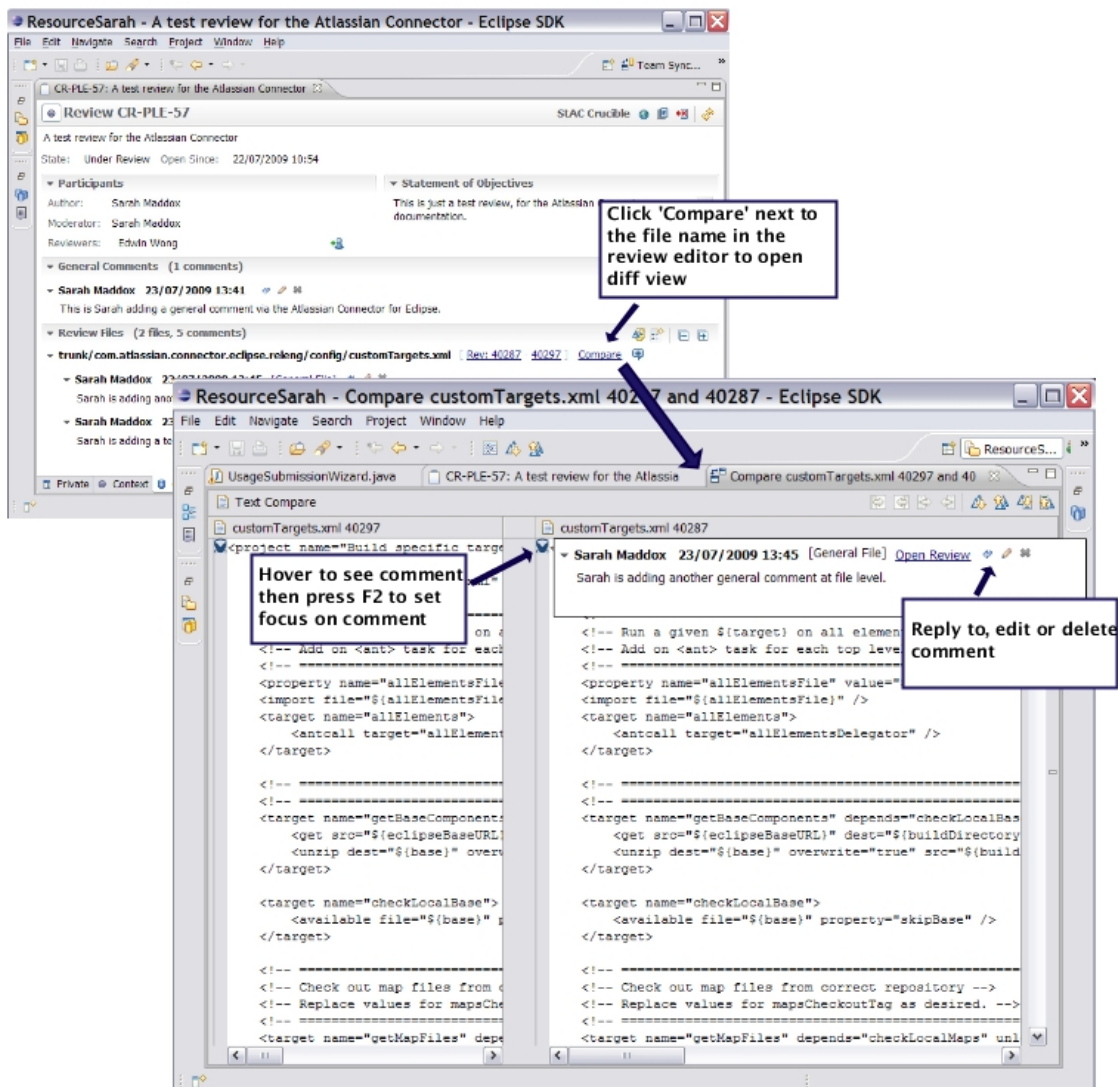
1. Open the Crucible review in an Eclipse review editor. Follow these instructions to open the review from your code editor, or see the complete instructions on [opening a review](#):
 - Make sure that the Crucible review task is **active** in Mylyn Eclipse. To activate the Mylyn task, right-click the review in your Eclipse Mylyn task list and select '**Activate**'.
 - Scroll down your code editor until you see a review comment marked by a Crucible icon  in the left-hand ruler.
 - Hover over the Crucible icon to see the comment text and other details.
 - Press F2 to bring the comment into focus.
 - The review comment box becomes an inline dialogue box. Click '**Open Review**' to open the review in the Eclipse review editor.
2. Open the diff view by clicking '**Compare**' next to the file you want, in the 'Review Files' section of the review editor as shown in [diagram 2](#) below.
3. The 'Compare' view opens, as shown in [diagram 2](#) below. Scroll down until you see a review comment marked by a Crucible icon  in the left-hand ruler.
 - Hover over the Crucible icon to see the comment text and other details.
 - Press F2 to bring the comment into focus.
 - The review comment box becomes an inline dialogue box. Now you can reply to the comment, edit it or delete it, depending on your permissions.

Diagram 2: Working with review comments in a diff view



Next, take a look at [creating a post-commit review](#).

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Creating a Post-Commit Review from within Eclipse

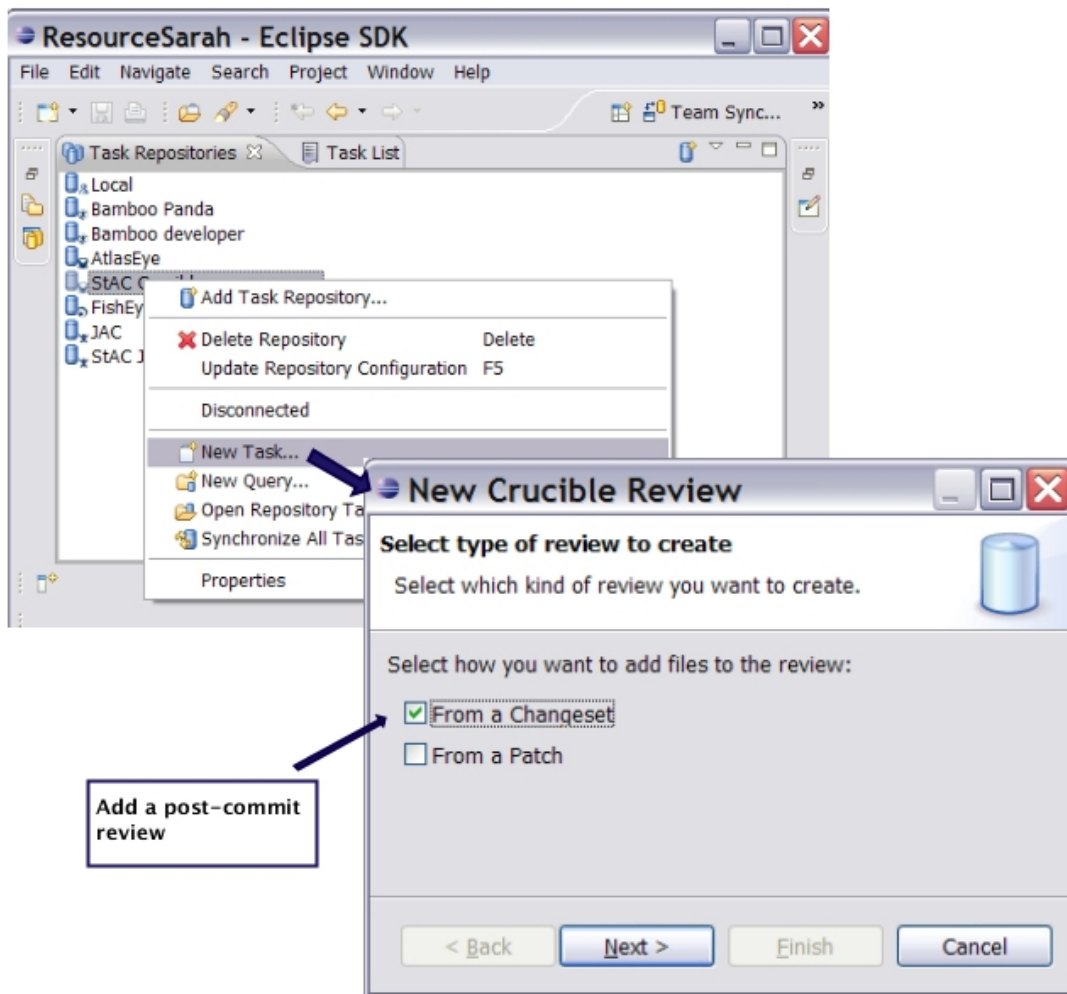
Once you have set up your Crucible server as an [Eclipse task repository](#), you can create a Crucible review without leaving Eclipse. The Atlassian Connector for Eclipse allows you to create post-commit reviews, as described below, and also patch reviews, as described in [another topic](#).

A post-commit review allows you to ask your colleagues to review changesets that have already been committed to your source repository.

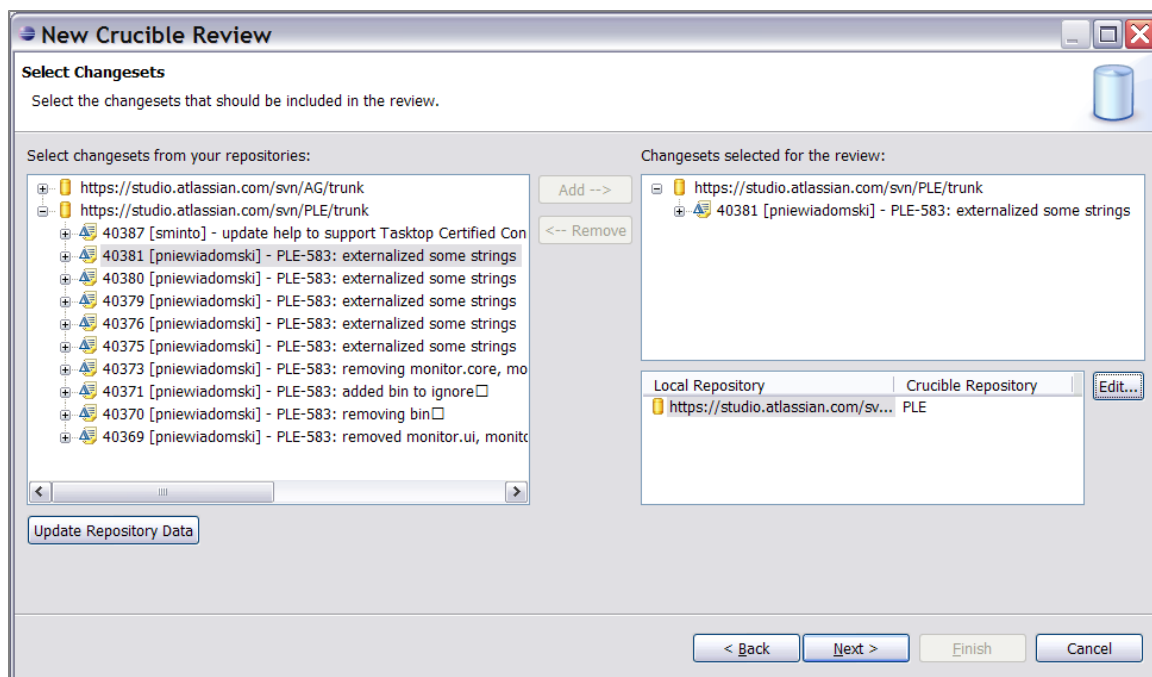
To create a post-commit review,

1. If you have not already set up a task repository for your Crucible server, please do so by following the steps in the [configuration guide](#)
2. Open your Eclipse 'Task Repositories' view and find the task repository that you defined for your Crucible server:
 - In the Eclipse toolbar, select '**Window**', '**Show View**', '**Other**'.
 - The 'Show View' popup window appears. Open the '**Tasks**' folder and click '**Task Repositories**'.
 - The 'Task Repositories' view appears, as shown in [diagram 1](#) below. (If you have trouble finding the 'Task Repositories' view, try the [full instructions with screenshots](#).)
3. Right-click the task repository that you defined for your Crucible server and select '**New Task**'.
4. The 'New Crucible Review' window appears, as shown in [diagram 1](#) below. Select '**From a Changeset**' and click '**Next**'.
5. The 'New Crucible Review' window now shows recent changesets from your repository, as shown in [screenshot 2](#) below. Select one or more changesets from the list on the left. (Hold down the Ctrl key to select more than one changeset at once.)
6. Click '**Add**' to move the selected changeset(s) into the box on the right.
7. If prompted, click the '**Edit**' button to map your local repository to the repository on your Crucible server.
8. Click '**Next**'.
9. The 'New Crucible Review' window now prompts you for the review title and other details, as shown in [screenshot 3](#) below. Enter the details and click '**Finish**'.

[Diagram 1: Adding a review as a task in your Crucible repository](#)



Screenshot 2: Adding changesets to your review



Screenshot 3: Adding the review details

New Crucible Review

Enter the details of the review.

Title: Just a test review created while documenting the Atlassian Connector for Eclipse

Project: Atlassian Eclipse Moderator: Sarah Maddox Author: Sarah Maddox

Objectives:

I'm creating this review while documenting the Atlassian Connector for Eclipse

Update Repository Data

Reviewers:

ed

- ☐ Andreas Fredriksson
- ☐ Bogdan Dziedzic
- ☐ Dave Tredinnick
- ☐ Eddie Kua
- ☐ Edwin Dawson
- ☒ Edwin Wong
- ☐ Fredrik Holmqvist
- ☐ Fred Simon

☒ Allow anyone to join

☒ Start review immediately (if permitted)

< Back Next > Finish Cancel

Next, take a look at creating a patch review.

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Creating a Patch Review from within Eclipse

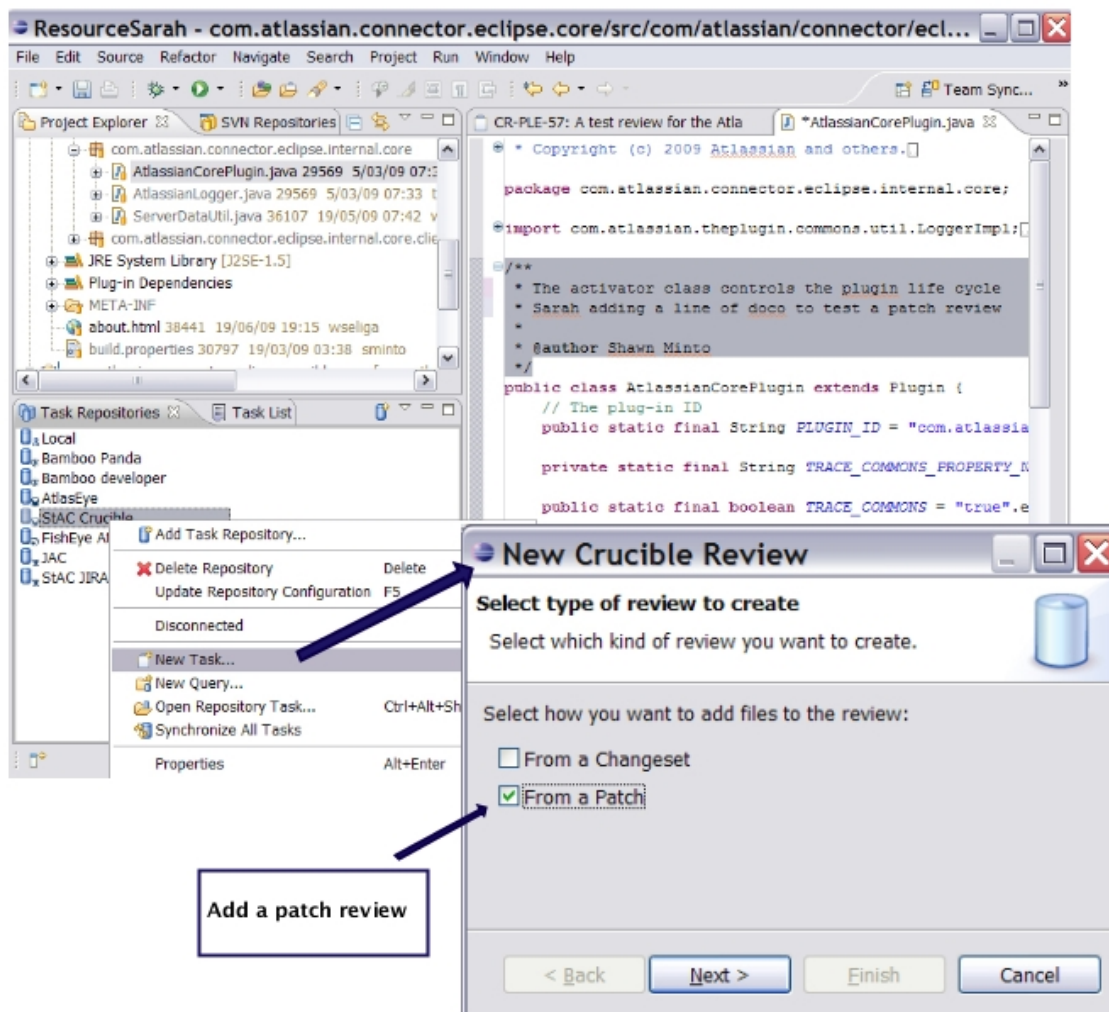
Once you have set up your Crucible server as an [Eclipse task repository](#), you can create a Crucible review without leaving Eclipse. The Atlassian Connector for Eclipse allows you to create patch reviews, as described below, and also post-commit reviews, as described in [another topic](#).

A patch review allows you to ask your colleagues to review a proposed change before you have committed it to your source repository.

To create a patch review,

1. Create the patch from the changes you have made to your source file in the Eclipse editor, and **copy the patch to your clipboard**. Here are some brief instructions, in case you have not done this before:
 - Save the changes you have made to your source file locally. There is no need to commit these changes to your source repository.
 - Right-click on your source file in the Eclipse editor and select **'Team', 'Create Patch'**.
 - The 'Create Patch' window appears. Select **'Save to Clipboard'**.
 - Click **'Next'**.
 - Leave the 'Advanced options' set to the default values. Click **'Finish'**.
2. If you have not already set up a task repository for your Crucible server, please do so by following the steps in the [configuration guide](#)
3. Open your Eclipse 'Task Repositories' view and find the task repository that you defined for your Crucible server:
 - In the Eclipse toolbar, select **'Window', 'Show View', 'Other'**.
 - The 'Show View' popup window appears. Open the **'Tasks'** folder and click **'Task Repositories'**.
 - The 'Task Repositories' view appears, as shown in [diagram 1](#) below. (If you have trouble finding the 'Task Repositories' view, try the [full instructions with screenshots](#).)
4. Right-click the task repository that you defined for your Crucible server and select **'New Task'**.
5. The 'New Crucible Review' window appears, as shown in [diagram 1](#) below. Select **'From a Patch'** and click **'Next'**.
6. The 'New Crucible Review' window now prompts you to add your patch from your clipboard, as shown in [screenshot 2](#) below. Your patched change will appear in the box on the page, because you have already copied it to your clipboard.
7. Select **'Include this Patch from the clipboard to the review'**.
8. Select your repository on your Crucible server.
9. Click **'Next'**.
10. The 'New Crucible Review' window now prompts you for the review title and other details, as shown in [screenshot 3](#) below. Enter the details and click **'Finish'**.

Diagram 1: Adding a review as a task in your Crucible repository



New Crucible Review

Enter the details of the review.

Title: A test review created from a patch

Project: Atlassian Eclipse Connector Moderator: Sarah Maddox Author: Sarah Maddox

Objectives:

This is Sarah creating a patch review via the Atlassian Connector for Eclipse.

Reviewers:

ed

- ☐ Andreas Fredriksson
- ☐ Bogdan Dziedzic
- ☐ Dave Tredinnick
- ☐ Eddie Kua
- ☐ Edwin Dawson
- ☒ Edwin Wong
- ☐ Fredrik Holmqvist
- ☐ Fred Simon
- ☐ Igor Sereda
- ☐ Jed Wesley-Smith
- ☐ Mikhail Fedko
- ☐ Murad Mamedov
- ☐ Ted Rice

☒ Allow anyone to join

☒ Start review immediately (if permitted)

Update Repository Data

< Back Next > Finish Cancel

Next, take a look at assigning reviewers.

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Assigning Reviewers in Eclipse

Once you have set up your Crucible server as an [Eclipse task repository](#), you can add reviewers to or remove reviewers from a Crucible review without leaving Eclipse.

To assign reviewers in the Eclipse review editor,


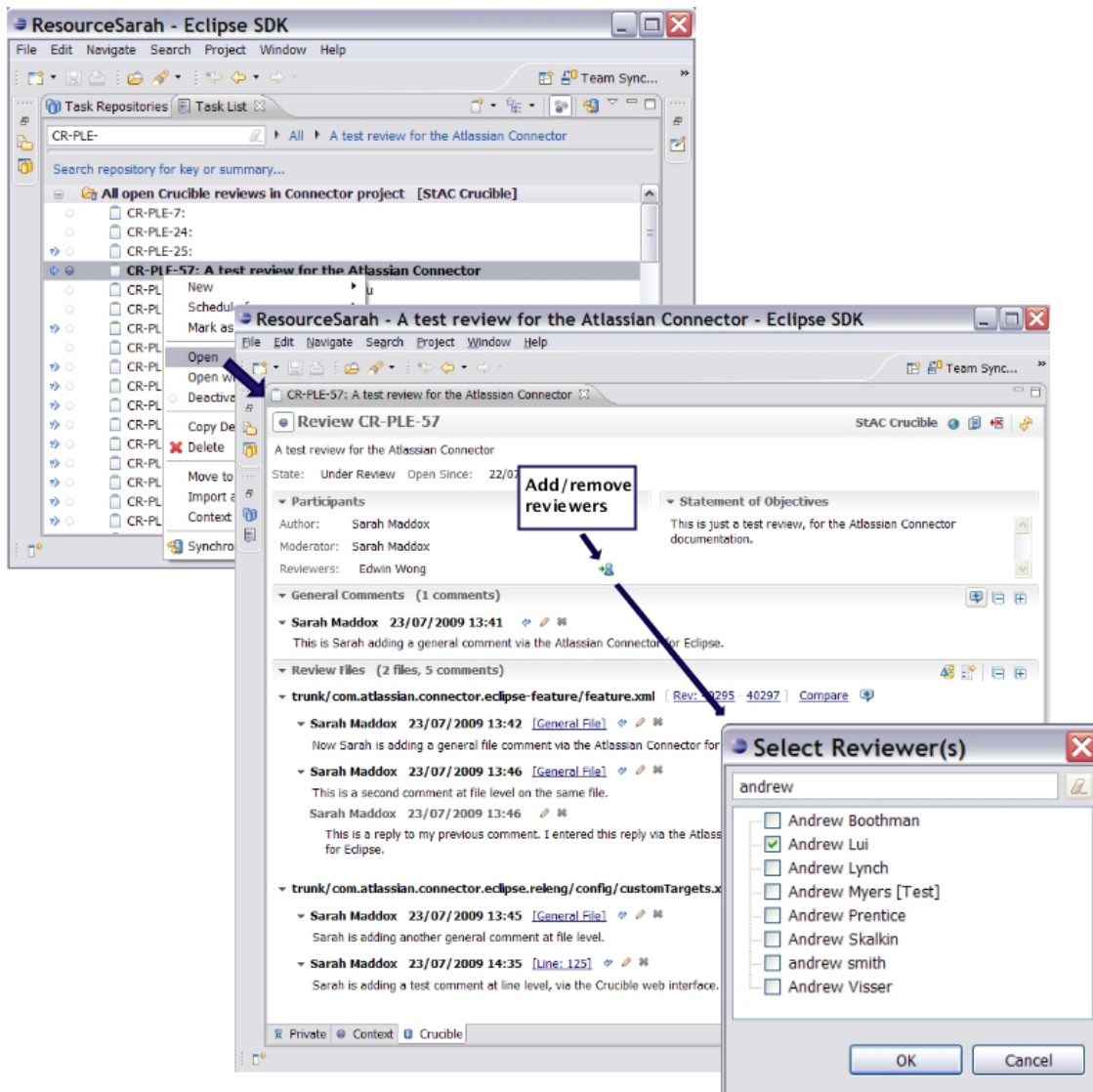
1. If you have not already done so, define your Crucible filters as described in the guide to [creating a query](#).
2. Open the review in an Eclipse editor:
 - Open your Eclipse 'Task List' view: In the Eclipse toolbar, select **'Window', 'Show View', 'Task List'**.
 - Open the folder with the name corresponding to the query that you want, i.e. the Crucible filter that you created earlier.
 - Double-click the line showing the review you want, or right-click the line and select **'Open'**.
3. The review details appear in an Eclipse editor, as shown in the [diagram](#) below. Click the **'Add/Remove Reviewers'** icon  in the 'Participants' section.
4. The 'Select Reviewer(s)' screen appears, showing a list of Crucible users available for the review. Narrow down the list of users by entering part of a name in the text box at the top.
 - To add a reviewer to the review, select the checkbox next to their name.
 - To remove a reviewer from the review, deselect the checkbox next to their name.

Diagram: Adding and removing reviewers in Eclipse



Next, take a look at adding changesets or patches to a review.

RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)


- Viewing a Review in Eclipse
- Managing Review Comments in Eclipse
- Working on a Review in the Diff View in Eclipse
- Creating a Post-Commit Review from within Eclipse
- Creating a Patch Review from within Eclipse
- Assigning Reviewers in Eclipse
- Adding Changesets or Patches to Review in Eclipse

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Adding Changesets or Patches to Review in Eclipse

Once you have set up your Crucible server as an [Eclipse task repository](#), you can add add changesets and patches to a Crucible review without leaving Eclipse.

To add a changeset to a Crucible review,

1. If you have not already done so, define your Crucible filters as described in the guide to [creating a query](#).
2. Open the review in an Eclipse editor:
 - Open your Eclipse 'Task List' view: In the Eclipse toolbar, select '**Window**', '**Show View**', '**Task List**'.
 - Open the folder with the name corresponding to the query that you want, i.e. the Crucible filter that you created earlier.
 - Double-click the line showing the review you want, or right-click the line and select '**Open**'.
3. The review details appear in an Eclipse editor, as shown in [diagram 1](#) below. Click the '**Add Changesets**' icon  in the 'Review Files' section.
4. The 'Add Changeset' screen appears, showing a list of committed changesets in your repository, as shown in [screenshot 2](#) below. Select one or more changesets from the list on the left. (Hold down the Ctrl key to select more than one changeset at once.)
5. Click '**Add**' to move the selected changeset(s) into the box on the right.
6. Click the '**Edit**' button to map your local repository to the repository on your Crucible server.
7. Click '**Finish**'.

To add a patch to a Crucible review,


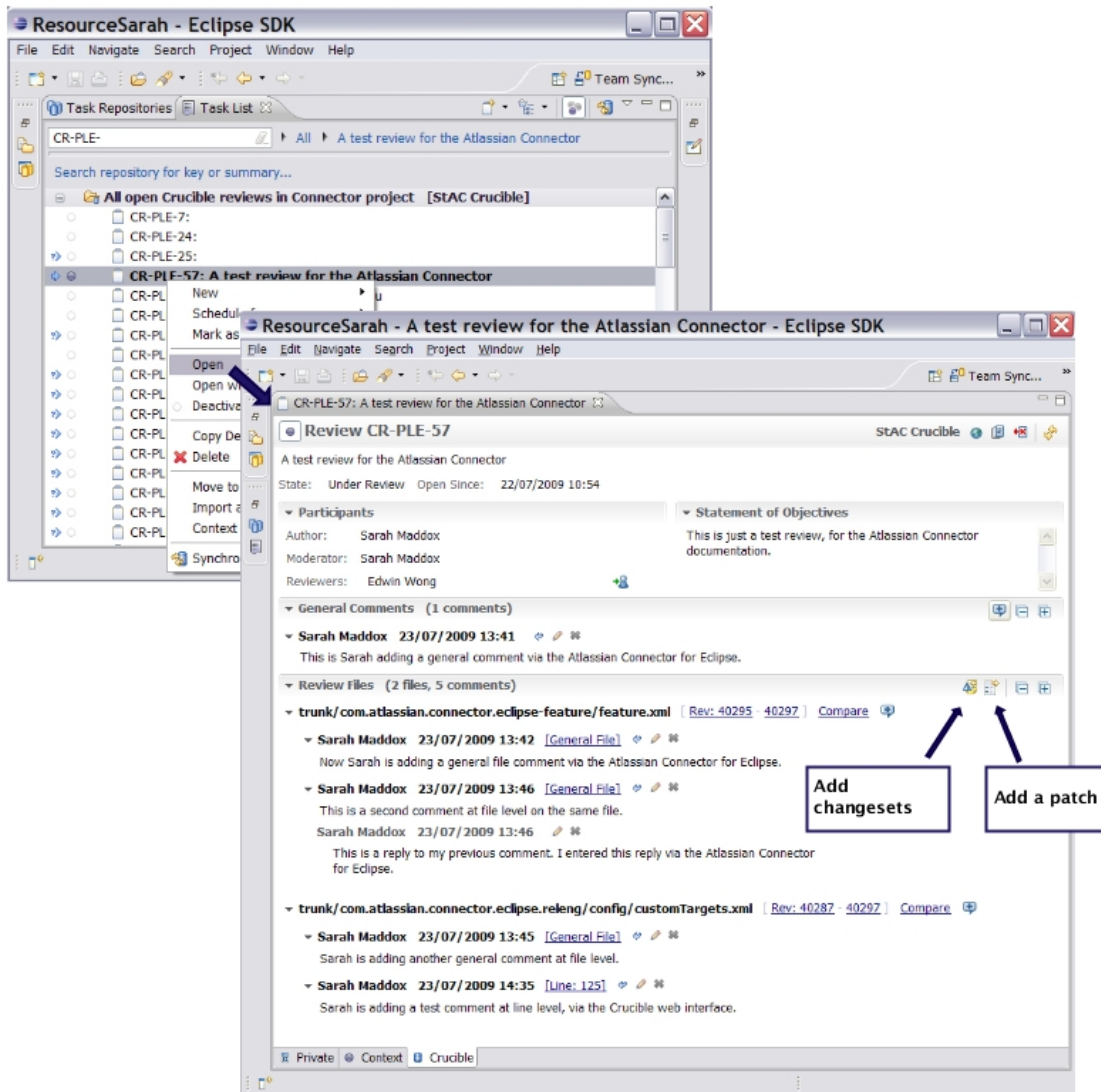
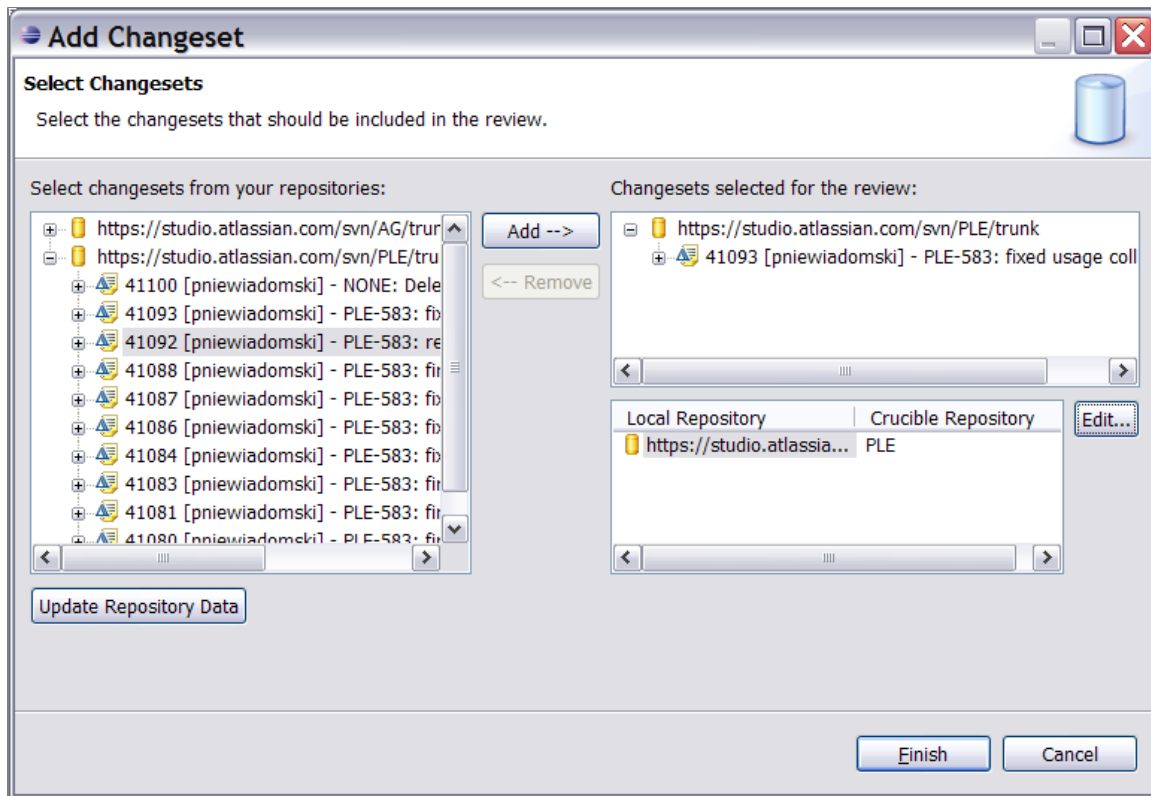
1. Create the patch from the changes you have made to your source file in the Eclipse editor, and **copy the patch to your clipboard**. Here are some brief instructions, in case you have not done this before:
 - Save the changes you have made to your source file locally. There is no need to commit these changes to your source repository.
 - Right-click on your source file in the Eclipse editor and select '**Team**', '**Create Patch**'.
 - The 'Create Patch' window appears. Select '**Save to Clipboard**'.
 - Click '**Next**'.
 - Leave the 'Advanced options' set to the default values. Click '**Finish**'.
2. If you have not already done so, define your Crucible filters as described in the guide to [creating a query](#).
3. Open the review in an Eclipse editor:
 - Open your Eclipse 'Task List' view: In the Eclipse toolbar, select '**Window**', '**Show View**', '**Task List**'.
 - Open the folder with the name corresponding to the query that you want, i.e. the Crucible filter that you created earlier.
 - Double-click the line showing the review you want, or right-click the line and select '**Open**'.
4. The review details appear in an Eclipse editor, as shown in [diagram 1](#) below. Click the '**Add a Patch**' icon  in the 'Review Files' section.
5. The 'Add Patch' window appears, as shown in [screenshot 3](#) below. Your patched change will appear in the box on the page, because you have already copied it to your clipboard. Select '**Include this Patch from the clipboard to the review**'.
6. Select your repository on your Crucible server.
7. Click '**Finish**'.

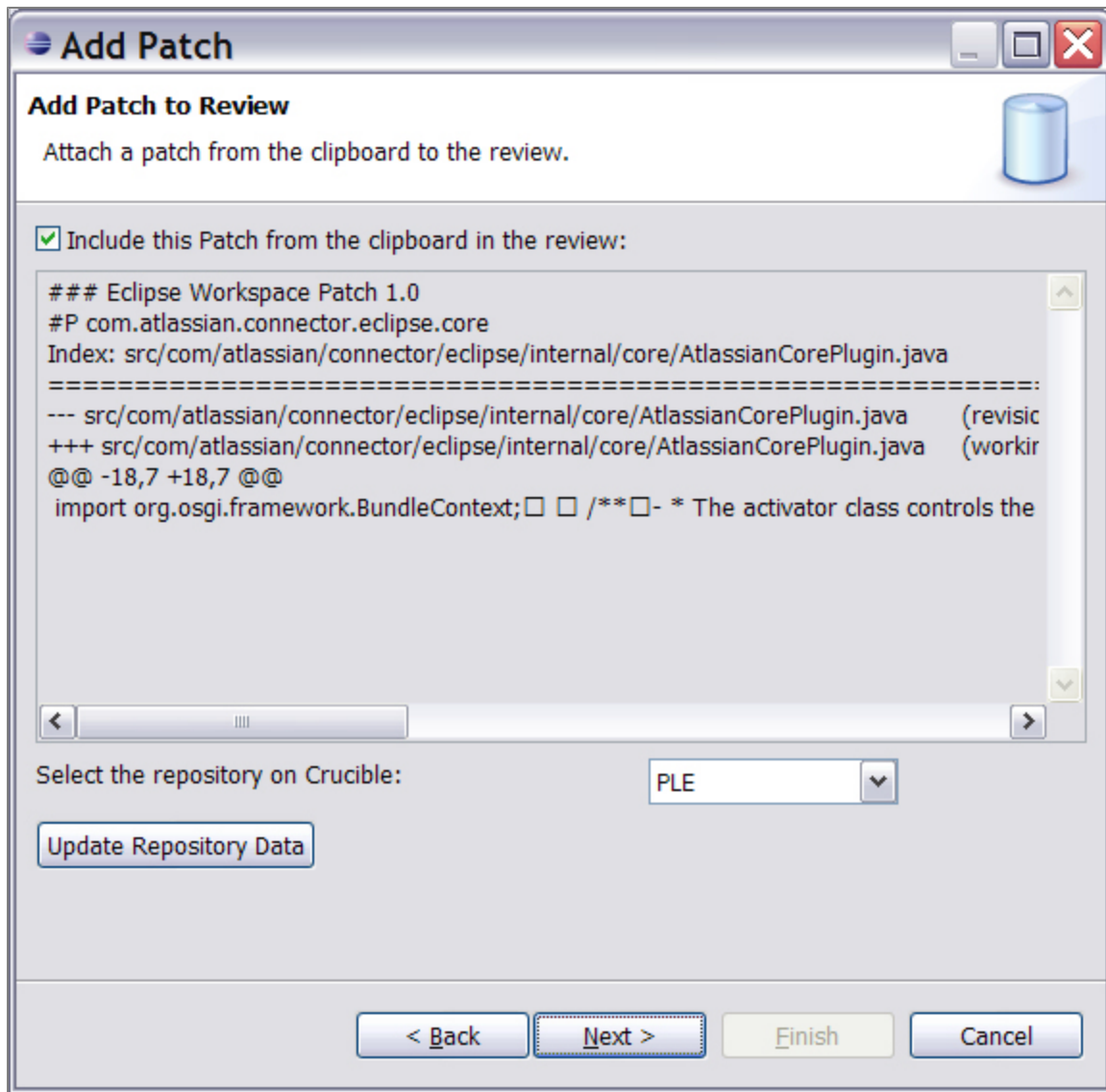
Diagram 1: Adding changesets and patches to a review



Screenshot 2: Changesets



Screenshot 3: Patch



RELATED TOPICS

Working with Crucible Reviews in Eclipse

- [Monitoring Review Updates in Eclipse](#)
- [Viewing a Review in Eclipse](#)
- [Managing Review Comments in Eclipse](#)
- [Working on a Review in the Diff View in Eclipse](#)
- [Creating a Post-Commit Review from within Eclipse](#)
- [Creating a Patch Review from within Eclipse](#)
- [Assigning Reviewers in Eclipse](#)
- [Adding Changesets or Patches to Review in Eclipse](#)

Refer to the [Crucible documentation](#) for more information about Crucible reviews.

Using FishEye in the Eclipse Connector

Using the Atlassian Connector for Eclipse, you can move quickly between your IDE and your [FishEye](#) view of your source repository.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

- [Configuring your FishEye Options in Eclipse](#)
- [More about Configuring FishEye Repositories in Eclipse](#)

- [Working with your FishEye Repository View in Eclipse](#)

RELATED TOPICS

[Installation and Upgrade Guide for the Eclipse Connector](#)

Refer to the [FishEye documentation](#) for more information about FishEye repository insight.

Configuring your FishEye Options in Eclipse

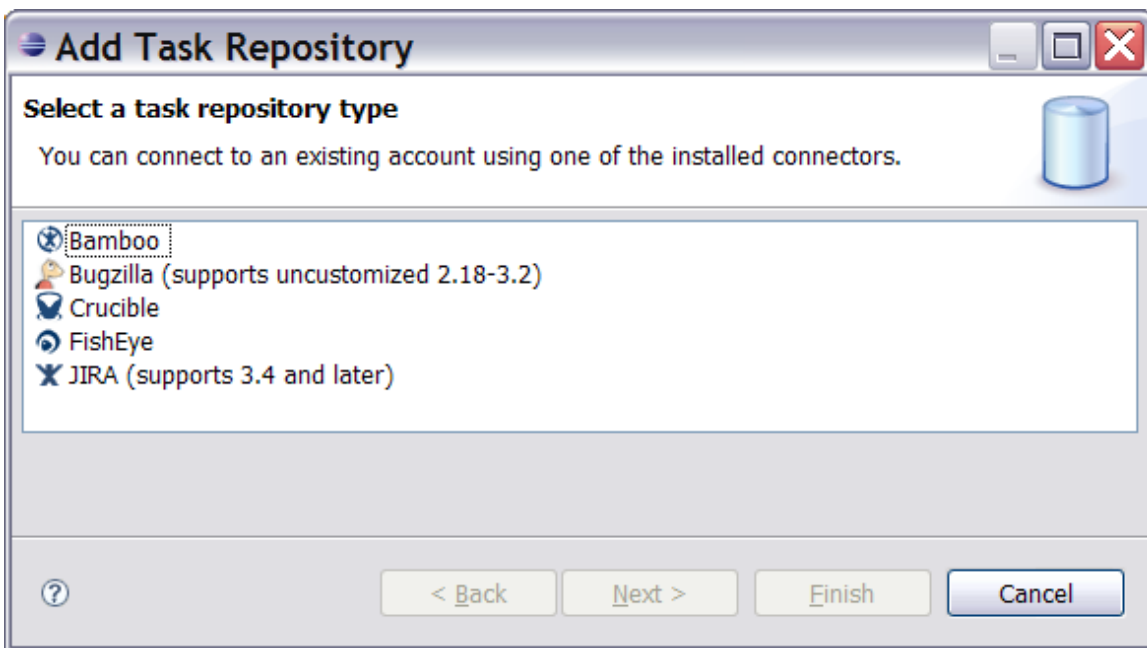
Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

On this page:

- [Adding a Task Repository](#)
- [Setting Up a FishEye Task Repository](#)

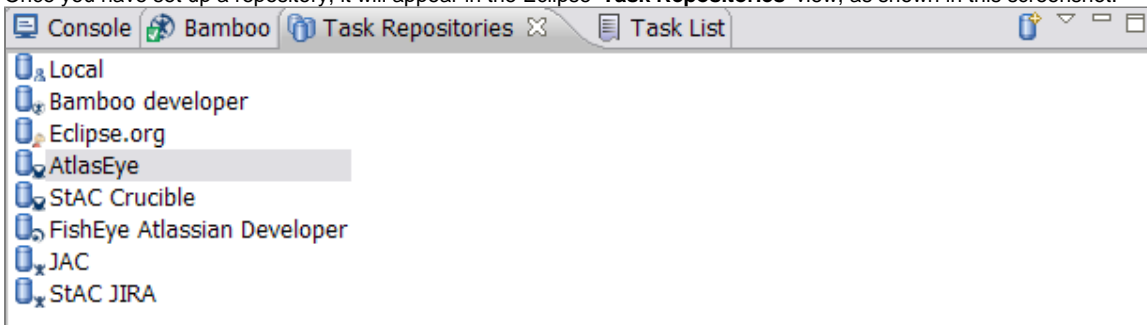
Adding a Task Repository

1. In Eclipse Mylyn, open the '**Task Repositories**' view:
 - Click '**Window**', '**Show View**', '**Other**'.
 - Open the '**Tasks**' category and select the '**Task Repositories**' view.
2. Click the '**Add Task Repository**' icon.
3. The 'Add Task Repository' screen appears, as shown below:



4. Now you can select and configure one or more of the task repositories and configure them to match your server(s), as described below.

Once you have set up a repository, it will appear in the Eclipse '**Task Repositories**' view, as shown in this screenshot:



Setting Up a FishEye Task Repository




Single server for both Crucible and FishEye = single task repository

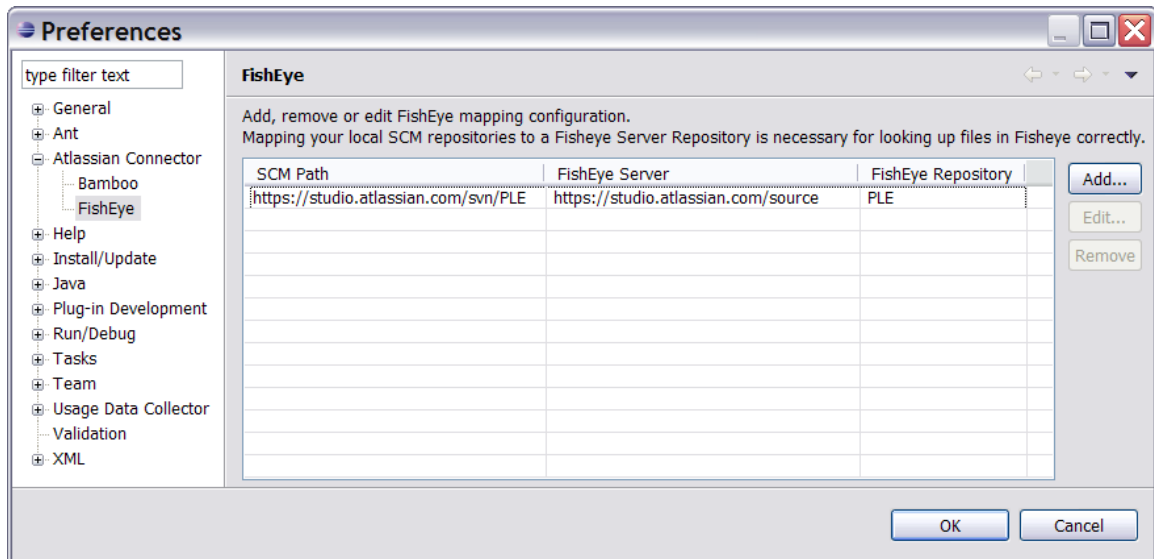
If you have a single server for both FishEye and Crucible, you should set up the Crucible task repository and then select the '**Crucible Server Contains FishEye Instance**' option as described [above](#). Due to Mylyn requirements, you cannot set up two separate task repositories for the same URL.

If you do not have a Crucible server, you can hook up your FishEye server as an Eclipse task repository without the additional Crucible functionality.

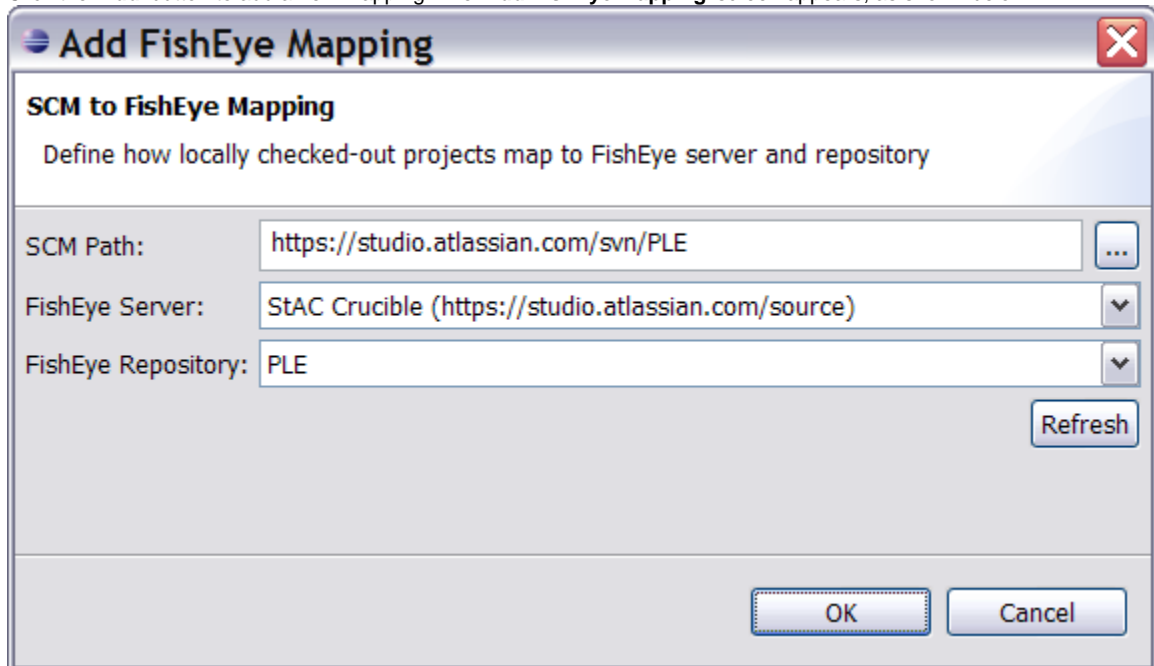
1. On the Eclipse 'Add Task Repository' screen, select the '**FishEye**' task repository type and click '**Next**'.
2. The 'FishEye Repository Settings' screen appears, as shown below:

3. Enter the following information:
 - **Server** — The location (URL) of your FishEye server.
 - **Label** — A descriptive name for your FishEye server, e.g. 'FishEye Atlassian Developer'.
 - **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
 - **User ID** — The username you use to connect to your FishEye server.
 - **Password** — Your password on the FishEye server, matching the above username.
 - **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.


 If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.
4. Click the '**Validate Settings**' button to verify the information you have entered.
5. Click '**Finish**' to save the changes.
6. The new task repository appears in your Eclipse '**Task Repositories**' view.
7. Now you need to map your source code to your FishEye repository and FishEye project.
 - Select '**Window**', '**Preferences**' then open the FishEye settings panel as shown below:



- Click the 'Add' button to add a new mapping. The 'Add FishEye Mapping' screen appears, as shown below:



- Enter the '**SCM Path**' — Supply the path to your source code as used locally by the project.
- Select the '**FishEye Server**' as configured in your task repositories. (See instructions above.)
- The connector will retrieve a list of FishEye repositories from the server you have supplied. Select the appropriate '**FishEye Repository**' to map your local project to the FishEye repository.

 Some notes about the FishEye integration:

- You can configure one or more FishEye repositories, i.e. you can connect to more than one FishEye server.
- We have chosen to configure FishEye servers as standard Mylyn task repositories. They are not really regular repositories like JIRA or Crucible, because there are no tasks associated with a FishEye task repository. We are using the task repository concept to take advantage of various infrastructural facilities available in Mylyn and to make user experience more consistent with the rest of the Atlassian products. See [More about Configuring FishEye Repositories in Eclipse](#).

RELATED TOPICS

[Installation and Upgrade Guide for the Eclipse Connector](#)

More about Configuring FishEye Repositories in Eclipse

This page contains additional information that you may find useful when setting up your FishEye server for the Atlassian Connector for Eclipse. You will find the step-by-step instructions in [Configuring the Eclipse Connector](#).

On this page:

- [Background to the FishEye Configuration and Mappings](#)
- [Limitations](#)

Background to the FishEye Configuration and Mappings

As an Eclipse plugin, the Atlassian connector needs additional configuration for the following reasons:

- The connector needs to know which FishEye server is serving the selected source file, and in which FishEye repository the file exists.
- The connector needs to know the current revision (or version) of the local file, in order to take the user to the proper place in the history on the FishEye server.
- Moreover files may come from various places in the repository, such as trunk, branches, tags, modules for CVS, etc.
- FishEye and Eclipse may use different access protocols to access the same source. For example, FishEye may use `file://` for SVN, while Eclipse uses `https://`.
- As a developer, you may have many projects open in a single Eclipse workspace. These projects come from different source repositories (SVN, CVS, etc) that are potentially indexed by different FishEye servers.

It would be time-consuming and unnecessary to configure FishEye for each project, because often many projects come from the same repository and are served by single FishEye instance and maybe by one or two FishEye repositories in that instance.

The solution:

- Configure the FishEye mappings not per project but per SCM setup (path) used locally by the project.
- Combine the selected SCM path (as seen locally in Eclipse) with the FishEye server (as configured in the Mylyn task repositories) and the FishEye repository from the selected FishEye server.
- Given a source file, the Atlassian Connector for Eclipse checks the SCM path from which the file was checked out, and finds the appropriate mapping (the combination as defined above). Then it knows which FishEye server and which FishEye repository the selected file belongs to.

Limitations

Please take note of the following:

- The configuration does not keep track of re-configuration in the Mylyn task repositories. If the mapped FishEye server is removed from the list of task repositories, your FishEye functionality may stop working.
- The file may be deleted on the server side or FishEye may be out of sync with the SCM repository, so you may get an HTTP 404 error.
- Please refer to the notes on support [for specific source control repositories](#).

RELATED TOPICS

[Configuring the Eclipse Connector](#)
[IDE Connector Documentation](#)

Working with your FishEye Repository View in Eclipse

Using the Atlassian Connector for Eclipse, you can move quickly between your IDE and your [FishEye](#) view of your source repository.

Prerequisites

Please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [installation guide](#), and defined at least one FishEye or Crucible+FishEye server, as described in the [configuration guide](#).

 Make sure that you have **mapped your source code to your FishEye repository and FishEye project**, as described in the [configuration guide](#). The options described below will not work if the FishEye mappings are not configured correctly. You will receive an **HTTP 404** error ('404 – Not Found') when you attempt to open the FishEye resource.

Opening a File in FishEye from Eclipse

[To open a file from Eclipse in FishEye, or to send a FishEye link to your colleagues,](#)

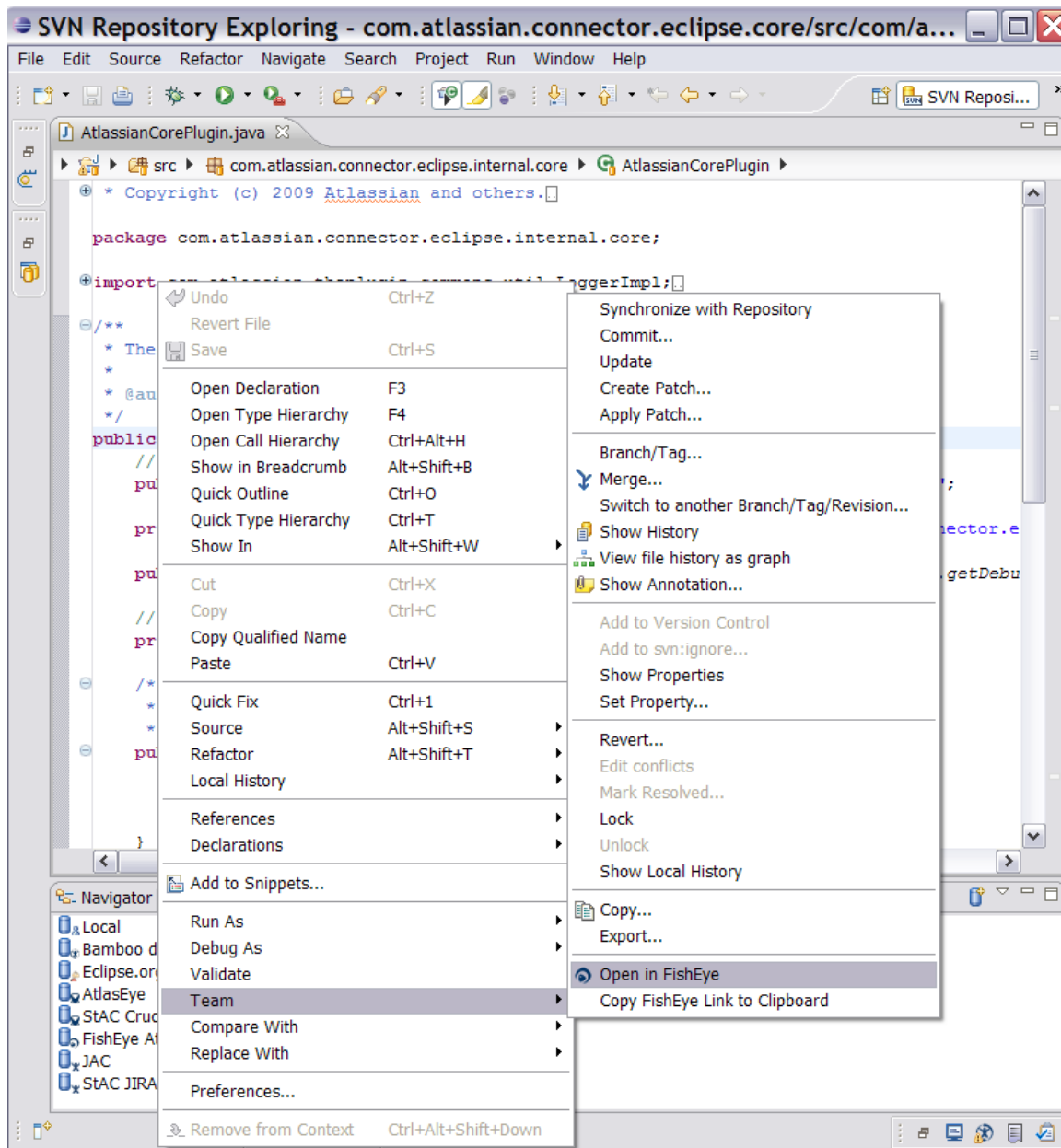
- Right-click the file in Eclipse and select one of the following:
 - **'Team'** then **'Open in FishEye'**.
 - Or **'Team'** then **'Copy FishEye Link to Clipboard'**.

This works from a number of Eclipse views. The only requirement is that the object you right-click represents a resource under version control with the FishEye mapping correctly configured. Here are some examples of where it will work:

- Editor — In this case, the link will point to the exact line that you right-clicked. Only standard text editors are supported.
- Package Explorer
- Project Explorer
- Outline view
- Hierarchy view
- And others too

Below we illustrate how to do this from the Eclipse editor.

Screenshot: Right-click menu in Eclipse



RELATED TOPICS

Using JIRA in the Eclipse Connector

The Atlassian Connector for Eclipse allows you to monitor, add and update [JIRA](#) issues while remaining in your integrated development environment.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

- [Configuring your JIRA Options in Eclipse](#)
- [Working with JIRA Issues in Eclipse](#)

RELATED TOPICS

[Installation and Upgrade Guide for the Eclipse Connector](#)

Refer to the [JIRA documentation](#) for more information about JIRA issue tracking.

Configuring your JIRA Options in Eclipse

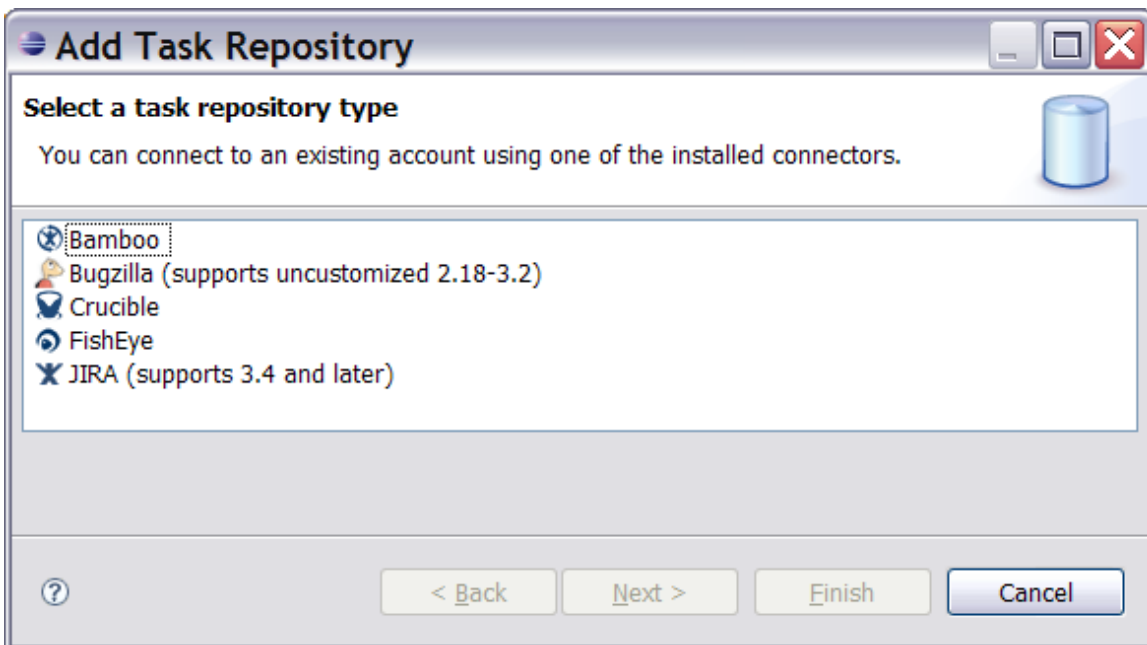
Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [Installation Guide](#).

On this page:

- [Adding a Task Repository](#)
- [Setting Up a JIRA Task Repository](#)

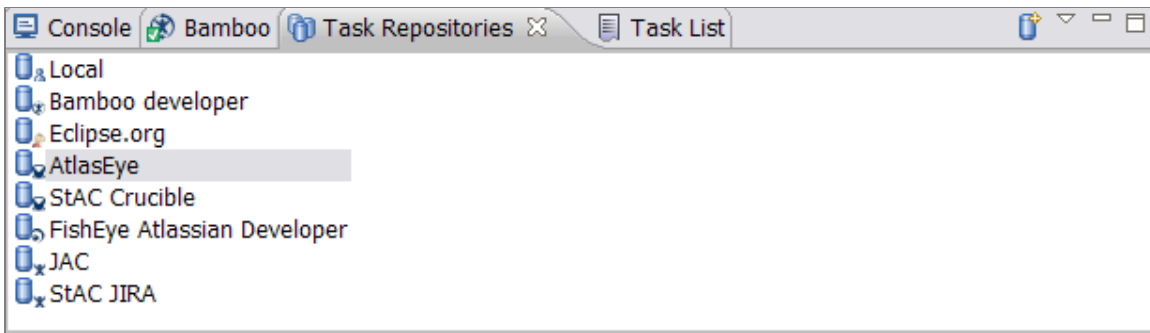
Adding a Task Repository

1. In Eclipse Mylyn, open the '**Task Repositories**' view:
 - Click '**Window**', '**Show View**', '**Other**'.
 - Open the '**Tasks**' category and select the '**Task Repositories**' view.
2. Click the '**Add Task Repository**' icon.
3. The 'Add Task Repository' screen appears, as shown below:



4. Now you can select and configure one or more of the task repositories and configure them to match your server(s), as described below.

Once you have set up a repository, it will appear in the Eclipse '**Task Repositories**' view, as shown in this screenshot:




Setting Up a JIRA Task Repository

1. On the Eclipse 'Add Task Repository' screen, select the '**JIRA**' task repository type and click '**Next**'.
2. The 'JIRA Repository Settings' screen appears, as shown below:

Add Task Repository...

JIRA Repository Settings

 Authentication credentials are valid.

Server:

Label: ☐ Disconnected

User ID:

Password: ☐ Save Password

Additional Settings

Compression: ☐ Enabled

Refresh configuration: ☐ Automatically

Completed tasks: ☐ Based on resolution

Subtasks: ☐ Show linked tasks

Follow redirects: ☐ Enabled

Time tracking: working days per week

working hours per day

Search results: ☒ Limit

► Advanced Configuration

Character encoding: ☒ Default (UTF-8)

☐ Other:

► Http Authentication

► Proxy Server Configuration

Task Editor Settings

☐ Plain Text


☒ Confluence (default)

☐ MediaWiki

☐ TWiki

☐ Textile

☐ TracWiki




3. Enter the following information:

- **Server** — The location (URL) of your JIRA server.
- **Label** — A descriptive name for your JIRA server, e.g. 'JAC'.
- **Disconnected** — If necessary, you can tick this checkbox to disable a particular server without deleting it. This is useful if your servers are behind a firewall and you do not have access to them at a particular point in time.
- **User ID** — The username you use to connect to your JIRA server.
- **Password** — Your password on the JIRA server, matching the above username.
- **Save Password** — Put a tick in the checkbox if you want to save your password on disk. Leave the checkbox unticked if you want to be asked for a password every time you start your IDE.

 If you choose to save the password, it is stored on your computer in a file that is difficult, but not impossible, for an intruder to read.

4. If necessary, you can change the additional settings. If in doubt, leave these settings as set by default.
 - The '**Task Editor Settings**' section is used to configure the markup language used by the task editor when Mylyn's [WikiText](#) extension is installed. By default, the Atlassian Connector for Eclipse will set the markup language to '**Confluence**', so that WikiText recognises Confluence wiki markup as used by JIRA. See the [Mylyn WikiText User Guide](#) for more information.
5. Click '**Finish**' to save the changes.
6. Eclipse Mylyn will prompt you to add a new query for the new JIRA repository. This is where you will choose your JIRA filter, to determine which issues appear in your task list. You can do this now or skip this step and do it later. You can also add more queries later.
7. The new task repository appears in your Eclipse '**Task Repositories**' view. Your JIRA issues will appear in your 'Task List' view.

 You can configure one or more JIRA repositories, i.e. you can connect to more than one JIRA server.

RELATED TOPICS

[Working with JIRA Issues in Eclipse](#)
[Installation and Upgrade Guide for the Eclipse Connector](#)

Working with JIRA Issues in Eclipse

The Atlassian Connector for Eclipse allows you to monitor, add and update [JIRA](#) issues while remaining in your integrated development environment.

Before reading the information below, please make sure that you have installed the Atlassian Connector for Eclipse, as described in the [installation guide](#), and defined a task repository for at least one JIRA server, as described in the [configuration guide](#).

User Guide

For information on setting up your JIRA server in Eclipse, please read the [configuration guide](#).

We have not yet written the documentation on working with JIRA issues. Please refer to the [JIRA Mylyn documentation](#) for guidelines.

RELATED TOPICS

- [Installation and Upgrade Guide for the Eclipse Connector](#)
- [Configuring the Eclipse Connector](#)
- [Using Bamboo in the Eclipse Connector](#)
- [Using Crucible in the Eclipse Connector](#)
- [Using FishEye in the Eclipse Connector](#)
- [Using JIRA in the Eclipse Connector](#)
- [Eclipse Connector FAQ](#)

Refer to the [JIRA documentation](#) for more information about JIRA functionality.

Eclipse Connector FAQ

Atlassian Connector for Eclipse — FAQ

Known issues, hints and tips and answers to commonly raised questions about the Atlassian Connector for Eclipse:

Configuration FAQ for the Eclipse Connector

- [Using NTLM Authentication with the Eclipse Connector](#)

Usage FAQ for the Eclipse Connector

Installation FAQ for the Eclipse Connector

- [Error while Collecting Items to be Installed](#)
- [Installation Error - Cannot Find a Solution Satisfying the Following Requirements](#)

Configuration FAQ for the Eclipse Connector

Here is a list of all entries in the configuration FAQ, plus the first few lines of content. Click a link to see the full text for each entry.

- [Using NTLM Authentication with the Eclipse Connector](#) — NTLM support is limited by the libraries that Mylyn uses for network communication. A suggestion is to take a look at the [NTLM authorisation proxy](#), which has been reported to work with the Mylyn JIRA connector.

Using NTLM Authentication with the Eclipse Connector

NTLM support is limited by the libraries that Mylyn uses for network communication. A suggestion is to take a look at the [NTLM authorisation proxy](#), which has been reported to work with the Mylyn JIRA connector.

Usage FAQ for the Eclipse Connector

Here is a list of all entries in the usage FAQ, plus the first few lines of content. Click a link to see the full text for each entry.

Installation FAQ for the Eclipse Connector

Here is a list of all entries in the installation troubleshooting section. Click a link to see the full text for each entry.

- [Error while Collecting Items to be Installed](#)
- [Installation Error - Cannot Find a Solution Satisfying the Following Requirements](#)

Error while Collecting Items to be Installed

Symptoms

During installation, after the license agreement screen, you may receive an error message similar to this:

```
An error occurred while collecting items to be installed
No repository found containing: com.atlassian.connector.commons/osgi.bundle/0.5.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.bamboo.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible.core/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.crucible.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.subclipse/org.eclipse.update.feature/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.subclipse.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.atlassian.connector.eclipse.ui/osgi.bundle/1.0.0.I20090320-BETA
No repository found containing: com.thoughtworks.xstream/osgi.bundle/1.3.0.v20090224-1640
No repository found containing: org.apache.commons.io/osgi.bundle/1.3.2.v20080604-1500
No repository found containing: org.jdom.jaxen/osgi.bundle/1.0.0.20081203-1100
No repository found containing: org.joda.time/osgi.bundle/1.6.0.v20081202-0100
```

Cause

P2 (Eclipse's update mechanism in Eclipse 3.4 and higher) has cached some incorrect information about the update site where the plugins reside for the Atlassian Connector for Eclipse.

Resolution

Follow these steps to clear the cached information:

1. Remove the Atlassian update site:

```
http://update.atlassian.com/atlassian-eclipse-plugin/e3.4
```

2. Add the Atlassian update site again:

```
http://update.atlassian.com/atlassian-eclipse-plugin/e3.4
```

3. Try to install the Atlassian Connector for Eclipse again.

More Information

See issue [PLE-393](#) in our bug tracker.

RELATED PAGES

[Installing the Eclipse Connector](#)

Installation Error - Cannot Find a Solution Satisfying the Following Requirements

Symptoms

During installation you may receive an error message similar to one of these:

- *Cannot complete the request. See the details.
Cannot find a solution satisfying the following requirements Match[requiredCapability: org.eclipse.equinox.p2.iu/org.eclipse.swt/[3.4.0.v3448f,3.4.0.v3448f]].*
- *Cannot complete the request. See the details.
Cannot find a solution satisfying the following requirements org.eclipse.pde.feature.jar [3.4.0.v20080603-7T7U-E_EkMNRNkb39WcMC].*
- *Cannot complete the install because one or more required items could not be found.
Software being installed: Atlassian Connector for Eclipse (recommended) 1.1.0.v20090624 (com.atlassian.connector.eclipse.feature.group 1.1.0.v20090624)
Missing requirement: Atlassian Connector for Eclipse (recommended) 1.1.0.v20090624 (com.atlassian.connector.eclipse.feature.group 1.1.0.v20090624) requires 'org.eclipse.mylyn.jira_feature.feature.group [3.2.0.I20090529,3.3.0]' but it could not be found*

Cause

This is most likely an error from P2 (Eclipse's update mechanism in Eclipse 3.4 and higher). The problem seems to be that the Mylyn update site is not enabled in your copy of Eclipse. There is a bug in Eclipse that these sites may not be automatically enabled even though we specify that we require them for installation.

If any of the installed features have unsatisfied dependencies or if features were previously installed from the extras or incubator update site, P2 may fail with an error similar to the one above.

Resolution

Follow these steps to enable the required sites:

1. In Eclipse, open **'Help', 'Software Updates'**.
2. Click the **'Available Software'** tab.
3. Click **'Manage Sites'**.
4. Look for either the **'Mylyn Extras'** update site or a URL that looks like this:

```
http://download.eclipse.org/tools/mylyn/update/extras
```

5. Make sure that the site or URL is checked.
6. Look for either the **'Mylyn For Eclipse 3.4'** update site or a URL that looks like this:

```
http://download.eclipse.org/tools/mylyn/update/e3.4
```

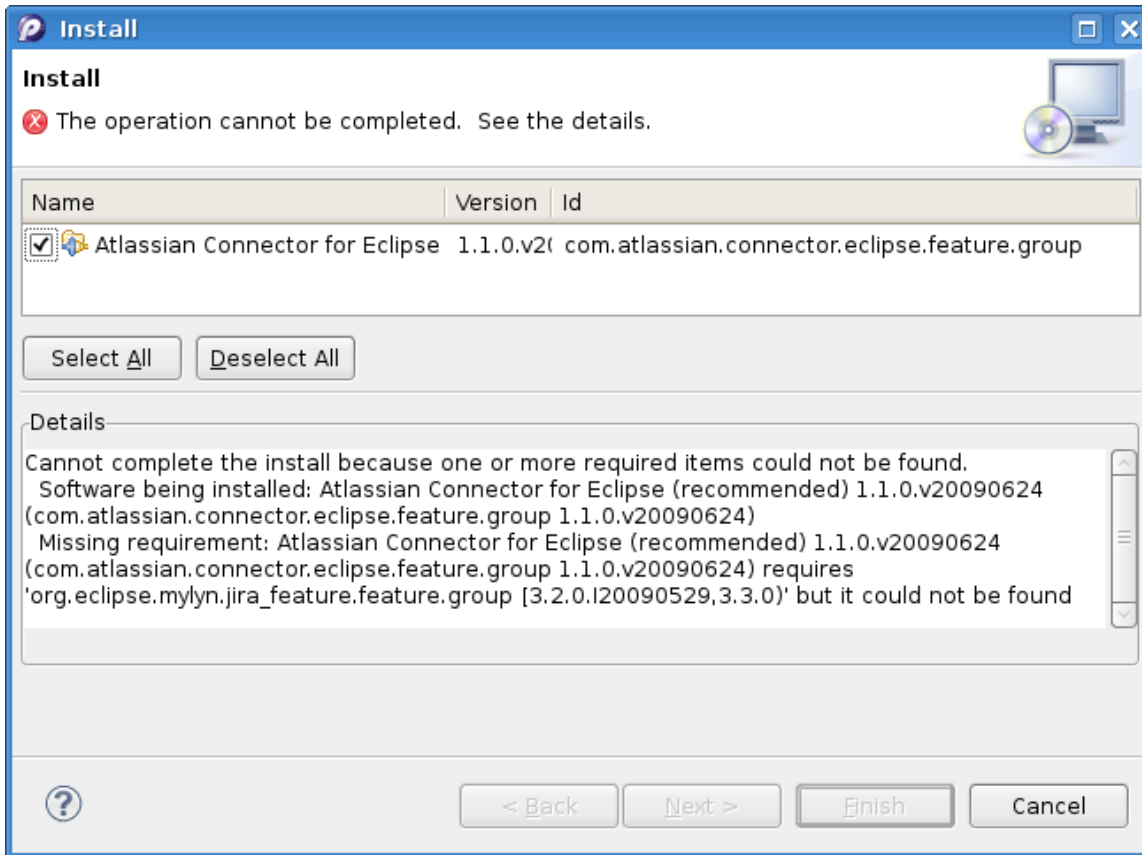
7. Make sure that the site or URL is checked.
8. Now you should update all installed features to the latest version. Click the **'Installed Software'** tab.
9. Click **'Update'**.
10. Restart if prompted.
11. Try to install the Atlassian Connector for Eclipse again.

More Information

- Issue [PLE-576](#).
- Issue [PLE-393](#).
- Issue [PLE-506](#).
- The [Mylyn FAQ](#).

Illustration

Here is a screenshot showing an example of one such error message:



RELATED PAGES

[Installing the Eclipse Connector](#)