CRUCIBLE
DOCUMENTATION

Crucible Documentation Home

Getting Started with Crucible 2.1

Release Notes (What's new)

Quick Start | Installation Guide | About Crucible

Download Crucible

Using/Administering Crucible 2.1

User's Guide

Administrator's Guide

Upgrade Guide

Crucible IDE Integration

Use the Atlassian Connector for Eclipse or the Atlassian Connector for IntelliJ IDEA to work with your Crucible code reviews right there in your development environment.

Resources

Crucible Knowledge Base

Support | Forums | FAQ

Development Hub

FishEye Documentation (Repository Viewer)

Guide to Installing an Atlassian Integrated Suite

Glossary

Previous Versions

Crucible 2.0 documentation
Crucible 1.6 documentation
Crucible 1.5 documentation
Crucible 1.2 documentation
Crucible 1.1 documentation
Crucible 1.0 documentation

Offline Documentation

You can download the Crucible documentation in PDF, HTML or XML format for use offline.

Recently Updated

Crucible 2.1 Release Notes updated by Edwin Dawson [Atlassian Technical Writer] (view change)	13 minutes ago
crucible.png attached by Edwin Dawson [Atlassian Technical Writer]	about an hour ago
The Crucible API updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Crucible SCM Plugins updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Crucible Event Listener Plugins updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Developing Crucible Plugins updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Crucible Reporting plugin updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Basic Servlet Example updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Servlet Examples updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Example Crucible SCM Plugin for JSR-170 (Apache JackRabbit) updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
Crucible Git Plugin updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 2 hours ago
SCM Plugin Examples updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 3 hours ago
Subversion SCM Plugin updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 3 hours ago
Perforce SCM Plugin updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 3 hours ago
File System SCM Plugin updated by Edwin Dawson [Atlassian Technical Writer] (view change)	about 3 hours ago

- More

Crucible Installation & Upgrade Guide

- Crucible Installation Guide
  - System Requirements
  - Installing Crucible
  - Configuring Crucible
  - Configuring Repositories
  - Best Practices for Crucible Configuration
- Crucible Upgrade Guide
  - Upgrading to a New Version of Crucible
  - Upgrading from FishEye to Crucible

Crucible Installation Guide

This guide explains how to get Crucible installed and running as easily as possible. Many references are made to the FishEye documentation.

This document assumes you have extracted your Crucible zip file into a directory called `/FISHEYE_HOME/`.

ⓘ Refer to our explanation of how Crucible works with FishEye.

> ✓ **Knowledge Base**
> You may find some useful information in the Knowledge Base too.

- System Requirements
- Installing Crucible
- Configuring Crucible
- Configuring Repositories
- Best Practices for Crucible Configuration

System Requirements

On this page:

- Java Runtime
- Source code Repository
- External Database
- Web Browser
- Operating System

These are the requirements for Crucible and FishEye. See the FishEye requirements for more specific details, including hardware requirements.

Java Runtime

Crucible supports the following Java platforms:

- JDK or JRE version **1.5** or later.

You can download a Java Runtime for Windows/Linux/Solaris here. On Mac OS X, the JDK is bundled with the operating system.

⚠ We strongly recommend the use of a 32-bit JDK/JRE rather than a 64-bit JDK/JRE. 64-bit JDK/JREs will consume the available RAM more rapidly, and this may result in poor performance.

Source code Repository

Crucible stand-alone supports use of the following systems as a repository:

- Subversion
- Confluence
- The server file system.

Crucible can also store uploaded files in its own database, removing the need for any kind of repository. When Crucible is used with FishEye, Subversion, CVS and Perforce are supported. See the FishEye System Requirements.

External Database

At this time, Crucible supports the following external databases:

- MySQL 5.x onwards
- PostGreSQL 8.x onwards.

Web Browser

Crucible has been tested with thw following browsers:

- Firefox 3
- Internet Explorer 7
- Safari 4
  ℹ Note: IE6 is not supported.

Crucible should work with most modern browsers.

> ⛔ IE6 is not supported and should not be used with Crucible.

Operating System

Crucible is a pure Java application and should run on any platform provided the above requirements are satisfied.

> ✓ **Font size tips**
> (Especially for Linux users.) For best results you may want to tweak your default monospace font and font-size. The default browser font is usually Courier New which can be hard to read in some browsers. We recommend choosing the same font you use in your IDE and selecting a font size approximately 2 points larger than your variable width font. Firefox 3, Internet Explorer 7 and Safari all have excellent font rendering. It is worth taking some time to tweak your fonts for the best experience.

Installing Crucible

This page contains instructions for the initial installation of Crucible.

On this page:

Installing the Crucible Binary Files

Follow these steps to install Crucible:

1. Download the Crucible zip file and extract it. This document assumes you have extracted your Crucible zip file into a directory called `/FISHEYE_HOME/`.
2. Ensure you have installed an appropriate Java runtime - see System Requirements. Ensure that `java` is in the `PATH`, or that the `JAVA_HOME` environment variable is set.

Setting up a Repository for use with Stand-alone Crucible

When accessing repositories from stand-alone Crucible, the SCM client interface will access the repositories on demand. This requires that you configure the plugin.

For complete instructions, see Configuring Repositories.

Setting up a Repository for use with FishEye and Crucible

- If you intend to use Crucible and FishEye with Subversion, please ensure you read the System Requirements

, Subversion client setup, and granting permission to FishEye to scan your repository.
- If you intend to use Crucible and FishEye with Git, please ensure you read the System Requirements and Git Client setup.
- If you intend to use Crucible and FishEye with Perforce, please ensure you read the System Requirements and Perforce Client setup.
- If you intend to use Crucible and FishEye with CVS, please ensure you read the System Requirements and CVS Client setup.

Next: Configuration

To go on with configuration, see the page configuring Crucible.

Configuring Crucible

On this page:

- Running Crucible
- Supplying Administration Password and License Key
- Accessing the Administration Pages
- Setting Up a Perforce Repository in Stand-Alone Crucible
- Setting Up a Subversion Repository in Stand-Alone Crucible
- Setting Up Reviewing of Confluence Pages in Crucible
- Setting Up the File System as a Code Repository For Crucible
- Setting Up a Repository via FishEye
- Setting Up Users
- Setting Up SMTP
- Using Crucible
- Stopping Crucible
- Information on FishEye integration

Running Crucible

This document assumes you have extracted your Crucible zip file into a directory called `/FISHEYE_HOME/`.

To run Crucible for the first time, simply do the following:

- On Windows:

```
C:\> cd FISHEYE_HOME\bin
C:\FISHEYE_HOME\bin> run.bat
```

- On Unix-based systems:

```
$ cd /FISHEYE_HOME/bin
$ ./run.sh
```

Once started, Crucible will run its own HTTP web server on port `8060`. You can access Crucible immediately by going to `http://HOSTNAME:8060/` in a browser.

> ⓘ By default, Crucible will listen on port `8060` for HTTP requests. It also listens on `127.0.0.1:8059` as a control port. You can configure both of these in the Administration screens, or by editing `/FISHEYE_HOME/config.xml` and restarting Crucible.

Supplying Administration Password and License Key

The first time you access the Crucible web server (`http://HOSTNAME:8060/`) you will see a screen like this, and here you will be asked for two things:

1. An administrator password. This password controls access to the Administration screens.
2. A license key. Please note your server ID. You can then get a Crucible evaluation license key here.

Accessing the Administration Pages

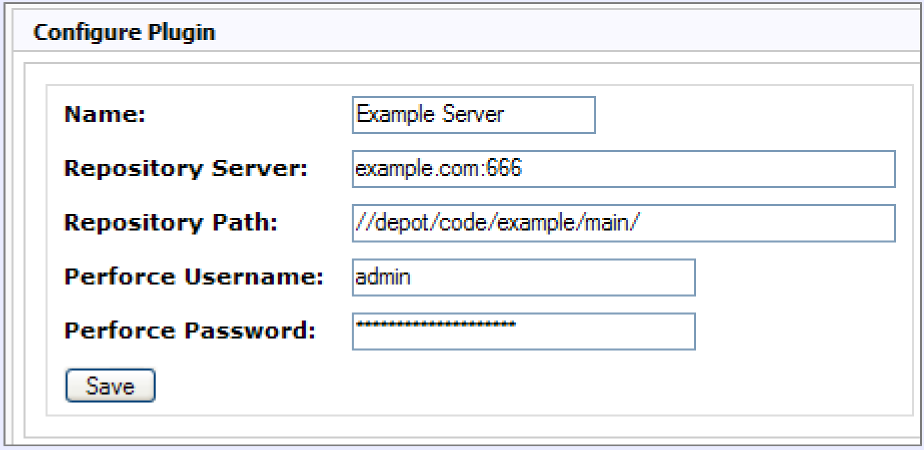
Once you have set up an administrator password (as described above), you can access the Administration screens at `http://HOSTNAME:8060/admin/`.

One of your first steps will be to set up access to a source-control repository, or an alternative form of code storage such as the Local File System or Atlassian Confluence.

Setting Up a Perforce Repository in Stand-Alone Crucible

To set up Perforce in stand-alone Crucible,

1. Ensure that the Perforce executable file is on the system path, in the Crucible server's Environment Variables
2. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
3. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
4. The '**Repository List**' screen opens.
5. Find the **Perforce** repository plugin and click its **Configure Plugin** link.
6. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
7. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: Example Server

Repository Server: example.com:666

Repository Path: //depot/code/example/main/

Perforce Username: admin

Perforce Password: *********************

|| Field || What to enter ||

Name	Choose a unique name for the repository.
Repository Server	Enter the base URL and port for the repository, for example: `example.com:666`.
Repository Path	Add the path to your Perforce repository. For example: `//depot/code/example/main`.
Perforce Username	Enter the username of the Perforce account that Crucible will use. (optional) ⓘ Note that this account should only have read-only access to the repository.
Perforce Password	Enter the password of the Perforce account that Crucible will use. (optional)

8. Click '**Save**'. The view will return to the list of repositories.
9. Your Perforce repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

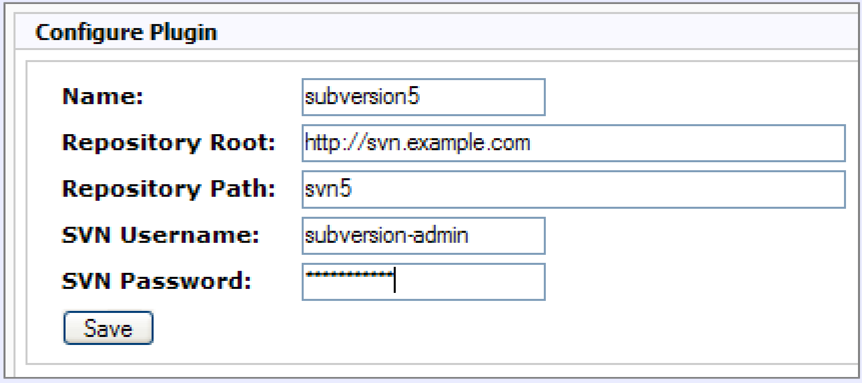
ⓘ There is no 'initial scanning' required in this process, as stand-alone Crucible's access to Perforce is strictly on-demand. Data is not indexed, hence there is no scanning.

ⓘ For those who may be interested, Crucible executes the Perforce command-line tool to enable this functionality.

Setting Up a Subversion Repository in Stand-Alone Crucible

To set up Subversion in stand-alone Crucible,

1. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
2. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
3. The '**Repository List**' screen opens.
4. Find the **SVN** repository plugin and click its **Configure Plugin** link.
5. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
6. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: subversion5

Repository Root: http://svn.example.com

Repository Path: svn5

SVN Username: subversion-admin

SVN Password: ************

|| Field || What to enter ||

Name	Choose a unique name for the repository.
Repository Root	Enter the repository root URL for the repository. If you are not sure what the repository root is, please see the instructions below under "Finding your Repository Root".
Repository Path	Add the path on the base URL where your repository. For example, if you used the root URL above, and the full path to your Subversion instance is 'http://svn.example.com/svn5/', you would enter 'svn5' into this field.
SVN Username	Enter the username of the Subversion account that Crucible will use. ℹ Note that this account should only have read-only access to the repository.
SVN Password	Enter the password of the Subversion account that Crucible will use.

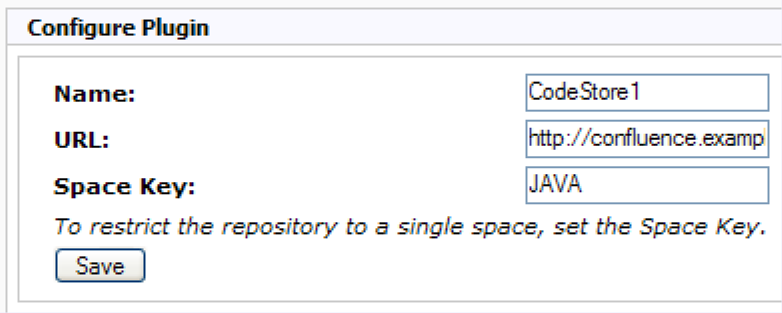
7. Click '**Save**'. The view will return to the list of repositories.
8. Your Subversion repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

ℹ There is no 'initial scanning' required in this process, as stand-alone Crucible's access to Subversion is strictly on-demand. Data is not cached, hence scanning is not required.

Setting Up Reviewing of Confluence Pages in Crucible

To set up Confluence as a Code Repository in Crucible,

1. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
2. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
3. The '**Repository List**' screen opens.
4. Find the **Confluence** repository plugin and click its **Configure Plugin** link.
5. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
6. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: CodeStore1

URL: http://confluence.examp

Space Key: JAVA

To restrict the repository to a single space, set the Space Key.

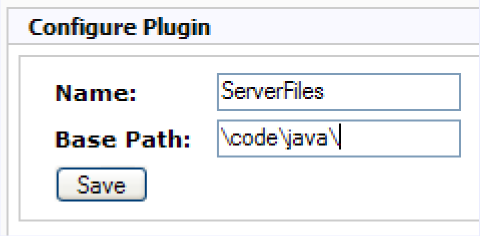
Field	What to enter
Name	Choose a unique name for the repository.
URL	Enter the URL of your Confluence instance.
Space Key	You may optionally enter a space key here to restrict Crucible's view to that key only. If there are many spaces in your Confluence instance you will find that this improves performance. You can set up several Confluence repositories in Crucible, each using the same Confluence instance but covering a different Space.

7. Click '**Save**'. The view will return to the list of repositories.
8. Now, access your Confluence instance. Open the '**Confluence Administration Console**', then select '**Trusted Applications**'. The Confluence '**Trusted Applications Details**' dialog opens.
9. In the '**Trusted Applications Details**' dialog, enter the URL of your Crucible instance into the '**Name**' field and click '**Send Request**'. The '**Application Alias**' will be automatically retrieved from Crucible. Save your changes.
10. Confluence is now set up as a code repository for Crucible. You will be able to select your Confluence from the list of repositories, then select files from the Confluence wiki and add them to reviews.

Setting Up the File System as a Code Repository For Crucible

To set up the File System as a Code Repository in stand-alone Crucible,

1. Start Crucible then open the '**Admin**' menu.
2. Under the '**System**' heading, click '**Plugins**' in the left-hand navigation bar.
3. The '**Plugins**' screen opens.
4. Next to '**File System SCM**', click '**Enable**'.
5. New options appear next to '**File System SCM**': '**Disable**' and '**Configure**'. Click '**Configure**'.
6. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
7. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: ServerFiles

Base Path: \code\java\

Field	What to enter
Name	Choose a unique name for the repository.
Base Path	Choose the lowest level of directory that Crucible will access.

8. Click '**Save**'. The view will return to the list of repositories.
9. The server file system is now set up as a code repository for Crucible. You will be able to select files from it when creating reviews, with the ability to browse the files and directories on the hard drive.

Setting Up a Repository via FishEye

⚠ This section requires a working FishEye installation as well as Crucible.

To use FishEye to access the source control repositories CVS, Subversion or Perforce for Crucible, see the FishEye documentation for how to add a repository.

Once you have added a repository, you can view it through FishEye at `http://HOSTNAME:8060/`.

> ⚠ **Building index and cache**
> FishEye needs to build an index and cache of the contents of your repository, so some information will not appear in FishEye until this is complete. This may take some time to complete, depending on the size of the repositories.
>
> ℹ We recommend you access the repository with a user that has only **read** access to the repository.

Setting Up Users

On initial setup of Crucible, there are no users. Adding user accounts is done via the Administration screens or by configuring Crucible/FishEye to use external authentication.

To add users:

1. Open the FishEye Administration screens at `http://HOSTNAME:8060/admin/`.

2. Click '**Users/Security**' under '**Global Settings**' in the '**Admin Menu**'.

Read more details about the different ways of creating users.

Setting Up SMTP

Crucible can email each review participant on a range of changes. Each user can then set up their own preferences. This is described in the User Profile guide.

First, you must set up the SMTP Server.

Using Crucible

You can access Crucible immediately by going to `http://HOSTNAME:8060/` in a browser

Or you can go directly into the Crucible homepage at `http://HOSTNAME:8060/cru`

Stopping Crucible

To stop the Crucible server:

- On Windows:

```
C:\> cd FISHEYE_HOME\bin
C:\FISHEYE_HOME\bin> stop.bat
```

- On Unix-based systems:

```
$ cd /FISHEYE_HOME/bin
$ ./stop.sh
```

Information on FishEye integration

If you want to know more about how Crucible and FishEye interact, refer to our explanation of how Crucible works with FishEye.

Configuring Repositories

This page contains links to the instructions for configuring the different kinds of repositories that Crucible supports. Click one of the links below to see the desired page.

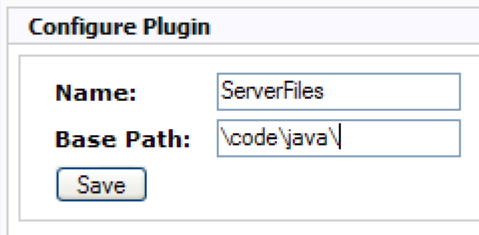
- Enabling Reviews from the Server File System in Crucible
- Setting Up a Git Repository in Stand-Alone Crucible
- Setting Up a Perforce Repository in Stand-Alone Crucible
- Setting Up a Repository via FishEye
- Setting Up a Subversion Repository in Stand-Alone Crucible
- Setting Up Reviewing of Confluence Pages in Crucible

Enabling Reviews from the Server File System in Crucible

Enabling Reviews from the Server File System in Crucible

To set up the File System as a Code Repository in stand-alone Crucible,

1. Start Crucible then open the '**Admin**' menu.
2. Under the '**System**' heading, click '**Plugins**' in the left-hand navigation bar.
3. The '**Plugins**' screen opens.
4. Next to '**File System SCM**', click '**Enable**'.
5. New options appear next to '**File System SCM**': '**Disable**' and '**Configure**'. Click '**Configure**'.
6. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
7. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: ServerFiles

Base Path: \code\java\

Field	What to enter
Name	Choose a unique name for the repository.
Base Path	Choose the lowest level of directory that Crucible will access.

8. Click '**Save**'. The view will return to the list of repositories.
9. The server file system is now set up as a code repository for Crucible. You will be able to select files from it when creating reviews, with the ability to browse the files and directories on the hard drive.

Setting Up a Git Repository in Stand-Alone Crucible

This page contains instructions on how to configure the Crucible Git plugin to access Git repositories.

Usage

The Git plugin is an early-access implementation of a Crucible SCM plugin for Git. It allows users to perform code reviews on a local Git repository (local to the Crucible server). The plugin does not 'pull' updates from a remote master repository. Synchronising with the master repository needs to be executed manually (via the command line), for the changes to appear in the plugin.

Installation

The plugin is installed by placing the .JAR file in the `FISHEYE_INST/var/plugins/user` directory of your Crucible install. Once installed, you need to enable the plugin in the Crucible Admin interface. Detailed instructions on the plugin installation steps can be found at the Managing Plugins page.

The plugin requires the Git command to be available in the system path when starting Crucible.

Configuring the plugin

Once the plugin has been installed, under the '**Administration**' - '**Repository List**' option, there should be a '**Plugin Repository List: Git**' entry. Select '**Configure Plugin**', then '**Add a repository**'. The fields required are:

Field	Description

Name	The name for the repository eg. *Project*
Repository Path	The location of the local Git repository clone

Once configured, the Git repository can be selected as the review source when creating a new review, whereupon reviews can be created either using changesets or by selecting files in the repository view.

Feedback

If you have any feedback on this plugin and its operation, we would appreciate users posting feedback in the Crucible Forums.

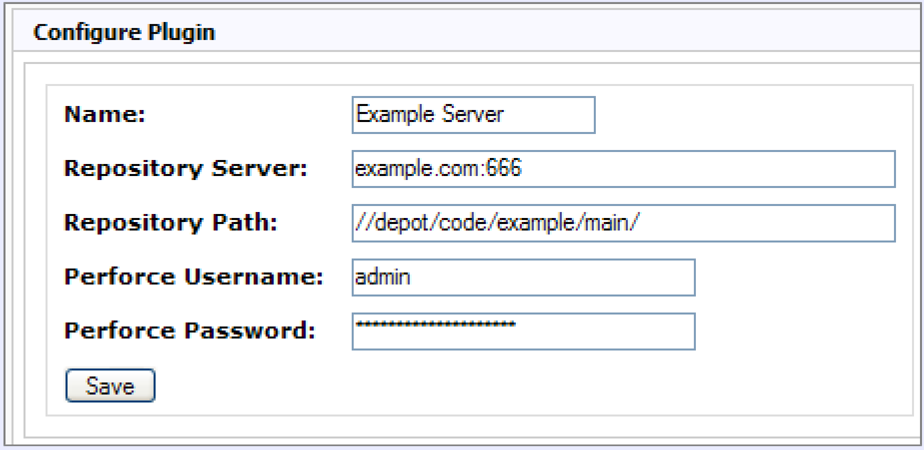
For more information, see the page on the Crucible Git Plugin.

Setting Up a Perforce Repository in Stand-Alone Crucible

Setting Up a Perforce Repository in Stand-Alone Crucible

To set up Perforce in stand-alone Crucible,

1. Ensure that the Perforce executable file is on the system path, in the Crucible server's Environment Variables
2. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
3. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
4. The '**Repository List**' screen opens.
5. Find the **Perforce** repository plugin and click its **Configure Plugin** link.
6. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
7. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name:	Example Server
Repository Server:	example.com:666
Repository Path:	//depot/code/example/main/
Perforce Username:	admin
Perforce Password:	*********************

|| Field || What to enter ||

Name	Choose a unique name for the repository.
Repository Server	Enter the base URL and port for the repository, for example: `example.com:666`.
Repository Path	Add the path to your Perforce repository. For example: `//depot/code/example/main`.
Perforce Username	Enter the username of the Perforce account that Crucible will use. (optional) ℹ Note that this account should only have read-only access to the repository.
Perforce Password	Enter the password of the Perforce account that Crucible will use. (optional)

8. Click '**Save**'. The view will return to the list of repositories.
9. Your Perforce repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

ℹ There is no 'initial scanning' required in this process, as stand-alone Crucible's access to Perforce is strictly on-demand. Data is not indexed, hence there is no scanning.

ℹ For those who may be interested, Crucible executes the Perforce command-line tool to enable this functionality.

Setting Up a Repository via FishEye

Setting Up a Repository in Crucible via FishEye

⚠ This section requires a working FishEye installation as well as Crucible.

To use FishEye to access the source control repositories CVS, Subversion or Perforce for Crucible, see the FishEye documentation for how to add a repository.

Once you have added a repository, you can view it through FishEye at `http://HOSTNAME:8060/`.

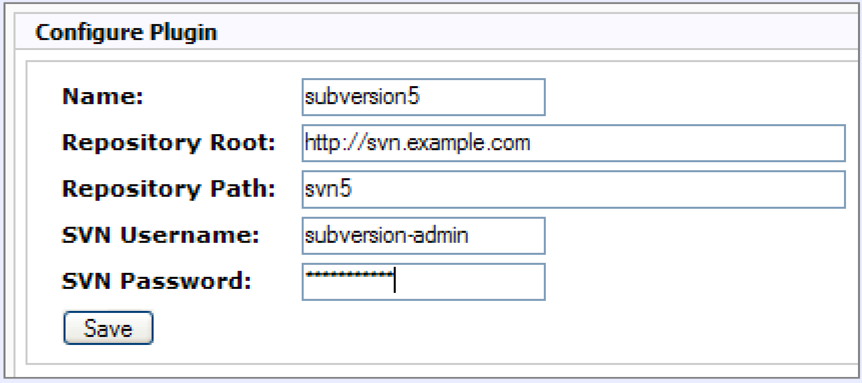
> ⚠ **Building index and cache**
> FishEye needs to build an index and cache of the contents of your repository, so some information will not appear in FishEye until this is complete. This may take some time to complete, depending on the size of the repositories.
>
> ℹ We recommend you access the repository with a user that has only **read** access to the repository.

Setting Up a Subversion Repository in Stand-Alone Crucible

Setting Up a Subversion Repository in Stand-Alone Crucible

To set up Subversion in stand-alone Crucible,

1. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
2. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
3. The '**Repository List**' screen opens.
4. Find the **SVN** repository plugin and click its **Configure Plugin** link.
5. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
6. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: subversion5

Repository Root: http://svn.example.com

Repository Path: svn5

SVN Username: subversion-admin

SVN Password: ************

|| Field || What to enter ||

Name	Choose a unique name for the repository.
Repository Root	Enter the repository root URL for the repository. If you are not sure what the repository root is, please see the instructions below under "Finding your Repository Root".
Repository Path	Add the path on the base URL where your repository. For example, if you used the root URL above, and the full path to your Subversion instance is 'http://svn.example.com/svn5/', you would enter 'svn5' into this field.
SVN Username	Enter the username of the Subversion account that Crucible will use. Note that this account should only have read-only access to the repository.
SVN Password	Enter the password of the Subversion account that Crucible will use.

7. Click '**Save**'. The view will return to the list of repositories.
8. Your Subversion repository is now set up for Crucible. You will be able to select changesets from it when creating reviews.

There is no 'initial scanning' required in this process, as stand-alone Crucible's access to Subversion is strictly on-demand. Data is not cached, hence scanning is not required.

Finding your Repository Root.

Run the following command:

```
svn info SVN_URL
```

Where SVN_URL is the complete URL of the repository you want to add.

You will get something like the following:

```
>svn info http://svn.example.com/svn5/

Path: svn5
URL: http://svn.example.com/svn5/
Repository Root: http://svn.example.com/
Repository UUID:
ce062a09-193b-427a-a7b3-a85007076e5d
Revision: 83
Node Kind: directory
Last Changed Author: ryan
Last Changed Rev: 83
Last Changed Date: 2009-05-07 10:48:41 +1000
(Thu, 07 May 2009)
```

Next to "Repository Root" is the URL you should define as your repository root. The path will be whatever is remaining.

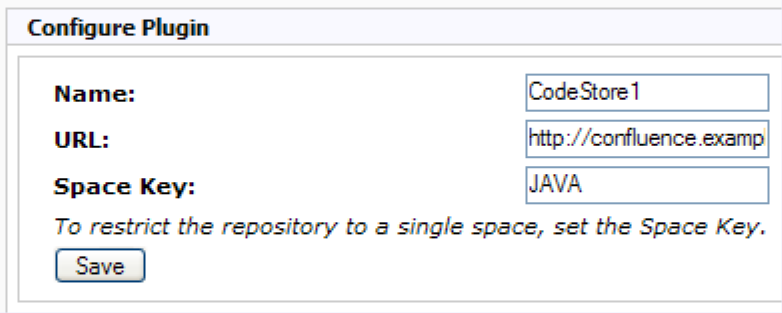
Setting Up Reviewing of Confluence Pages in Crucible

Setting Up Reviewing of Confluence Pages in Crucible

ℹ You need to first install the Confluence Crucible Plugin in the Confluence instance (see instructions on installing Confluence plugins), and you must set up a trust relationship so that your Crucible instance trusts your Confluence instance.

To set up Confluence as a Code Repository in Crucible,

1. Start Crucible then open the '**Admin**' menu by clicking the *Administration* link in the footer of the page.
2. Under the '**Repository Settings**' heading, click '**Repository List**' in the left-hand navigation bar.
3. The '**Repository List**' screen opens.
4. Find the **Confluence** repository plugin and click its **Configure Plugin** link.
5. The '**Configure Plugin**' screen opens. Click '**Add Repository**'.
6. The '**Add Repository**' screen opens. Fill in the fields.



Configure Plugin

Name: CodeStore1

URL: http://confluence.examp

Space Key: JAVA

To restrict the repository to a single space, set the Space Key.

Field	What to enter
Name	Choose a unique name for the repository.
URL	Enter the URL of your Confluence instance.
Space Key	You may optionally enter a space key here to restrict Crucible's view to that key only. If there are many spaces in your Confluence instance you will find that this improves performance. You can set up several Confluence repositories in Crucible, each using the same Confluence instance but covering a different Space.

7. Click '**Save**'. The view will return to the list of repositories.
8. Now, access your Confluence instance. Open the '**Confluence Administration Console**', then select '**Trusted Applications**'. The Confluence '**Trusted Applications Details**' dialog opens.
9. In the '**Trusted Applications Details**' dialog, enter the URL of your Crucible instance into the '**Name**' field and click '**Send Request**'. The '**Application Alias**' will be automatically retrieved from Crucible. Save your changes.
10. Confluence is now set up as a code repository for Crucible. You will be able to select your Confluence from the list of repositories, then select files from the Confluence wiki and add them to reviews.

Best Practices for Crucible Configuration

1. Set up a separate FISHEYE_INST folder location on the same system for Crucible's data.

This will allow for easy upgrades of the core program and neatly separated data backup.

2. Run Crucible on a dedicated machine, accessing its data on the local file system.

This is the best environment for swift Crucible performance. Avoid running Crucible in a virtual environment.

3. Do not give Crucible projects the same key as your JIRA projects.

When naming projects, take care to ensure that the key you assign to them is not the same as any of your JIRA projects. The reason for this is, if one of your Crucible projects has the same key as one of your projects in JIRA, then all links with that key will lead back to Crucible, rather than leading to JIRA, removing the ability to navigate between the two applications.

To avoid this, name your Crucible project keys differently. For example, you could place the following text at the beginning of each project key: `CR-` to distinguish it. So, for this case, if you have an existing JIRA key of '`RHUBARB`', you would create a Crucible key called '`CR-RHUBARB`' so that they do not conflict.

Crucible Upgrade Guide

- Upgrading to a New Version of Crucible
- Upgrading from FishEye to Crucible

- Upgrading from 1.6.x to 1.6.3.

Upgrading to a New Version of Crucible

ℹ Read about how your Crucible installation works with FishEye.

- Before you Start
- Crucible 2.0 Upgrade Notes
- Crucible 1.6 Upgrade Notes
- Upgrade Procedure
  - Method 1 - Using a FISHEYE_INST Directory
  - Method 2 - Without a FISHEYE_INST Directory
  - Method 3 - Without a FISHEYE_INST Directory, but would like to set one up

Before you Start

- Before upgrading you should always read the Release Notes for the version you are upgrading to, as well as any versions you are skipping.
- **We strongly recommend you make a backup of your data before upgrading Crucible**. Refer to the documentation on making a backup.
- Download the Crucible zip file.

Crucible 2.0 Upgrade Notes

- Supported browsers are: Safari 3+, FireFox 3+ and Internet Explorer 7+ (not IE6).

Crucible 1.6 Upgrade Notes

- Please see Crucible 1.6.3 Upgrade Guide

Upgrade Procedure

Method 1 - Using a FISHEYE_INST Directory

If you have Crucible configured to use a `FISHEYE_INST` directory, then simply:

1. Shutdown your existing fisheye server
2. Make a backup of your FISHEYE_INST directory
3. Extract the new Crucible version to a directory.
4. Leave your `FISHEYE_INST` environment variable set to its existing location.
5. Start Crucible from the new installation.

6. Follow any version-specific instructions found in the Release Notes.

ⓘ Read more about the `FISHEYE_INST` environment variable.

Method 2 - Without a FISHEYE_INST Directory

If you are not using `FISHEYE_INST`, you will need to copy some files from your old Crucible installation to your new one.

1. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
2. Delete the `/NEW_FISHEYE/var` directory.
3. Shut down the old Crucible instance if it is running.
4. Copy `/OLD_FISHEYE/config.xml` to `/NEW_FISHEYE/`.
5. Copy (or move) the `/OLD_FISHEYE/var` directory to `/NEW_FISHEYE/var`.
6. If you have a Cenqua-issued Crucible license, copy all `/OLD_FISHEYE/*.license` files to `/NEW_FISHEYE/`. (Atlassian-issued licenses are included within `config.xml`.)
7. Follow any version-specific instructions found in the Release Notes.

Method 3 - Without a FISHEYE_INST Directory, but would like to set one up

1. Shut down the old FishEye instance if it is running.
2. Set up the `FISHEYE_INST` environment variable, then create the FISHEYE_INST directory on your filesystem.
3. Copy the `/OLD_FISHEYE/config.xml` to `/FISHEYE_INST`.
4. Copy the `/OLD_FISHEYE/var` directory to `/FISHEYE_INST`.
5. Download Crucible.
6. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
7. Start Crucible from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
8. Follow the initial configuration steps outlined below.
9. If your configuration is not automatically picked up and you cannot see your existing repositories, check your Administration > Sys-Info page, where you will see information about FISHEYE_HOME and FISHEYE_INST. Check your FISHEYE_INST is pointing to the right directory.

Upgrading from FishEye to Crucible

If you have been using FishEye and now want to move to Crucible, you can do this without losing your FishEye repositories.

ⓘ Read about how your Crucible installation works with FishEye.

On this page:

Before you Start

We strongly recommend you make a backup of your data before following the steps below. Refer to the documentation on making a backup.

Upgrade Procedure

- Follow **Method 1** if you have a default configuration and are not using a FISHEYE_INST directory (that is, your FishEye binaries and data are all stored under the same location, in the default FISHEYE_HOME directory).
- Follow **Method 2** below if you have FishEye configured to use a FISHEYE_INST directory (that is, your FishEye binaries are stored in the FISHEYE_HOME directory, separate from your FishEye data in the FISHEYE_INST directory).
- Follow **Method 3** if you are not using a FISHEYE_INST directory but would now like to start using one.

ⓘ Read more about the FISHEYE_INST environment variable.

Method 1 - Without a FISHEYE_INST Directory (default)

1. Download Crucible.
2. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
3. Delete the `/NEW_FISHEYE/var` directory.
4. Shut down the old FishEye instance if it is running.
5. Copy `/OLD_FISHEYE/config.xml` to `/NEW_FISHEYE/`.
6. Copy the `/OLD_FISHEYE/var` directory to `/NEW_FISHEYE/var`.
7. If you have a Cenqua-issued FishEye license, copy `/OLD_FISHEYE/fisheye.license` to `/NEW_FISHEYE/`. (Atlassian-issued licenses are included within `config.xml`.)
8. Follow any version-specific instructions found in the Release Notes.
9. Start Crucible from the new installation by running NEW_FISHEYE/bin/run.sh. (Use run.bat on Windows).
10. Follow the initial configuration steps outlined below.

Method 2 - Using a FISHEYE_INST Directory

1. Shutdown your existing fisheye server.
2. Make a backup of your FISHEYE_INST directory.
3. Download Crucible and unzip the archive into a folder. This document assumes you have extracted your Crucible zip file into a directory called `/NEW_FISHEYE/`.
4. Leave your `FISHEYE_INST` environment variable set to its existing location.
5. Start Crucible from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
6. Follow the initial configuration steps outlined below.

Method 3 - Without a FISHEYE_INST Directory, but intending to set one up

1. Shut down the old FishEye instance if it is running.
2. Set up the `FISHEYE_INST` environment variable, then create the FISHEYE_INST directory on your filesystem.
3. Copy the `/OLD_FISHEYE/config.xml` to `/FISHEYE_INST`.
4. Copy the `/OLD_FISHEYE/var` directory to `/FISHEYE_INST`.
5. Download Crucible.
6. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
7. Start Crucible from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
8. Follow the initial configuration steps outlined below.
9. If you configuration is not automatically picked up and you cannot see your existing repositories, check your Administration > Sys-Info page, where you will see information about FISHEYE_HOME and FISHEYE_INST.

Check your FISHEYE_INST is pointing to the right directory.

Initial Crucible Configuration

1. You can access FishEye immediately by going to `http://HOSTNAME:8060/` in a browser.
2. The first time you run FishEye, enter your Crucible license key. To do this, update your Crucible license by opening '**Administration**', then '**Sys-Info/Support**'. On this screen, you can enter your Crucible license key. You can view your license key here. The Crucible functionality will be instantly unlocked.
3. If you do not already have user accounts configured, you will need to do this via the Administration screens or by configuring Crucible/FishEye to use external authentication.
   To add users:
   - Open the FishEye Administration screens at `http://HOSTNAME:8060/admin/`.
   - Click '**Users/Security**' under '**Global Settings**' in the '**Admin Menu**'.
     Read more details about the different ways of creating users.
4. Crucible can email each review participant on a range of changes. Each user can then set up their own preferences. This is described in the User Profile guide. First, you must set up the SMTP Server.

Crucible User's Guide

This page is an index of the content in the Crucible User's Guide. Click on a link below to see the desired page.

    - Deleting an Abandoned Review
    - Moving a Review to Another Project
- Using the Dashboard
    - Using the Project Dashboard
- Defining your Workflow
- Using Favourites
- Using Keyboard Shortcuts in Crucible
- Using RSS Feeds in Crucible
- Using Wiki Markup in Crucible

> **Manage your Crucible reviews inside your IDE**
> Use the Atlassian Connector for Eclipse or the Atlassian Connector for IntelliJ IDEA to work with your Crucible code reviews right there in your development environment.

Getting Started with Crucible

This page contains a basic overview of Crucible workflows, followed by a simple example showing a code review between two people.

Crucible is a flexible application that caters for a wide range team sizes and work styles. You will need to know about the basic roles used in Crucible.

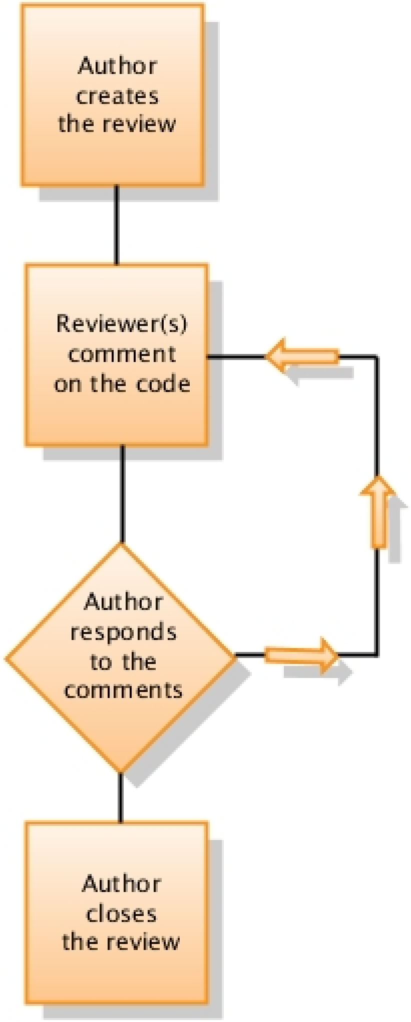
Roles:
There are several roles that review participants can take up:

- Author: Usually the creator of the code; the person who will act on the review's outcome.
- Reviewer: A participant that will comment on the source files in the review, raising points and discussion on the work that was done.
- Moderator: Usually the person who starts the review and is responsible for deciding the outcomes and closing it.

You will also need to understand how workflow is conducted in Crucible. This is configurable, but the most basic example follows.

Crucible Workflow:
There are a number of different ways in which you can use Crucible for code reviews. The following diagram shows the basic workflow that applies to most Crucible code reviews.

Diagram: Workflow for One-to-One Reviews


Author creates the review
Reviewer(s) comment on the code
Author responds to the comments
Author closes the review

Need more information? Read more about the different forms of workflow in Crucible.

Next, we explore the workflow in a two-person code review in Crucible.

Example Workflow: Two Participant Code Review

This is a simplified set of instructions for executing a one-to-one review involving two people. In this example, the code author wears "three hats", acting as review creator, moderator and code author, managing the review process as well as taking final responsibility for closing the review. The second person is the reviewer.

- Example Workflow: Two Participant Code Review
  - 1. The Author Starts the Review
  - 2. The Reviewer Comments on the Code
  - 3. The Author Responds to the Comments

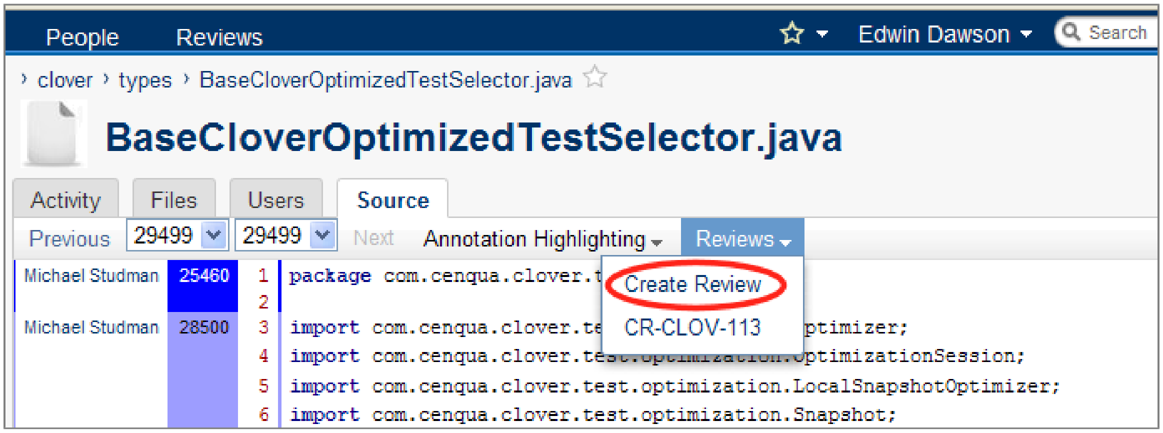
- 4. The Author Closes the Review

For instructions on Crucible workflow with more than two people, see this page.

1. The Author Starts the Review

To begin, the code author sets up the review. There are a number of ways to do this, but for this example, the author starts from the FishEye Source view of the file he wants to review:

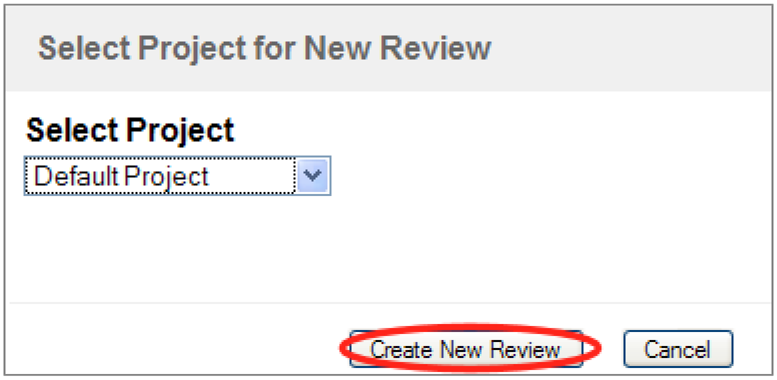
Screenshot: Opening a review from the FishEye Source view



```
package com.cenqua.clover.t

import com.cenqua.clover.te                             ptimizer;
import com.cenqua.clover.te                             ptimizationSession;
import com.cenqua.clover.test.optimization.LocalSnapshotOptimizer;
import com.cenqua.clover.test.optimization.Snapshot;
```

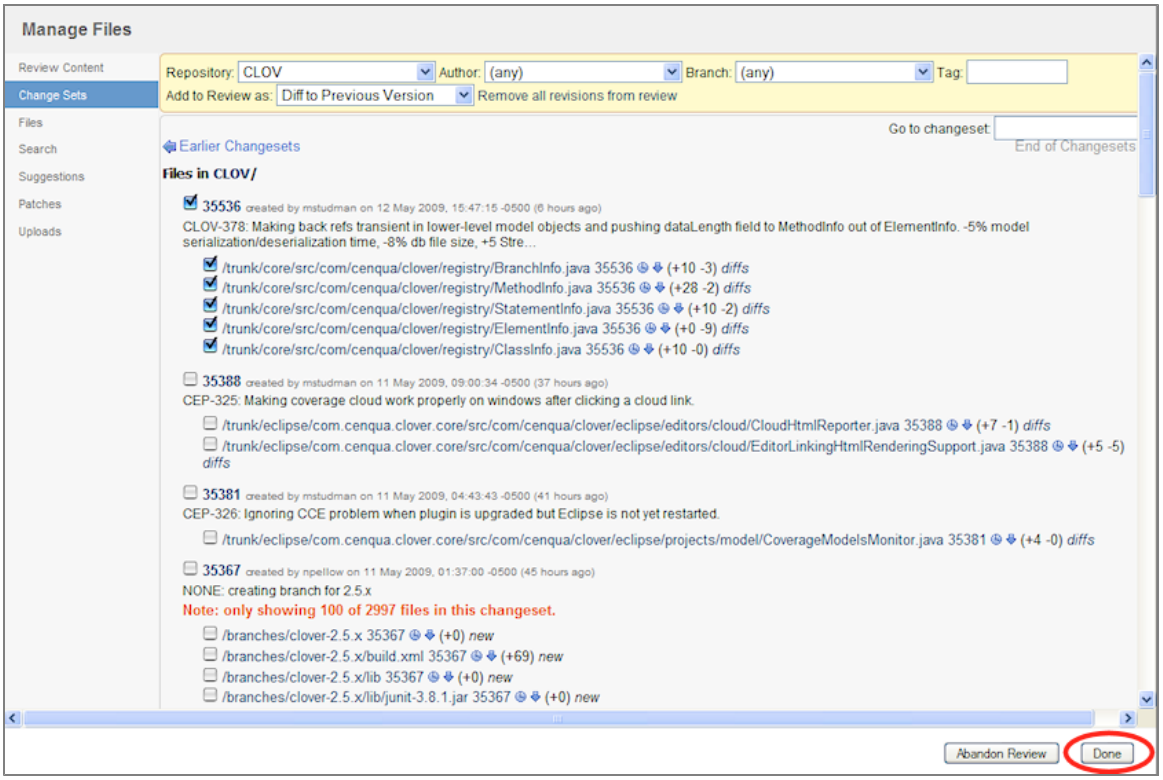
From the FishEye Source view, the author clicks the '**Reviews**' drop-down menu above the source view, then selects '**Create New Review**'. If there are multiple projects, the Project Selection dialog opens.

Screenshot: The Project Selection dialog



In the Project Selection dialog, you are prompted to choose a project for this review from the drop-down list. Once the selection is made, the author clicks the '**Create New Review**' button. The Manage Files dialog opens.

Screenshot: The Crucible Manage Files dialog



Manage Files

Repository: CLOV Author: (any) Branch: (any) Tag:
Add to Review as: Diff to Previous Version Remove all revisions from review

Go to changeset
End of Changesets

Earlier Changesets

Files in CLOV/

☑ 35536 created by mstudman on 12 May 2009, 15:47:15 -0500 (6 hours ago)
CLOV-378: Making back refs transient in lower-level model objects and pushing dataLength field to MethodInfo out of ElementInfo. -5% model serialization/deserialization time, -8% db file size, +5 Stre...

☑ /trunk/core/src/com/cenqua/clover/registry/BranchInfo.java 35536 (+10 -3) *diffs*
☑ /trunk/core/src/com/cenqua/clover/registry/MethodInfo.java 35536 (+28 -2) *diffs*
☑ /trunk/core/src/com/cenqua/clover/registry/StatementInfo.java 35536 (+10 -2) *diffs*
☑ /trunk/core/src/com/cenqua/clover/registry/ElementInfo.java 35536 (+0 -9) *diffs*
☑ /trunk/core/src/com/cenqua/clover/registry/ClassInfo.java 35536 (+10 -0) *diffs*

□ 35388 created by mstudman on 11 May 2009, 09:00:34 -0500 (37 hours ago)
CEP-325: Making coverage cloud work properly on windows after clicking a cloud link.

□ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/CloudHtmlReporter.java 35388 (+7 -1) *diffs*
□ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/EditorLinkingHtmlRenderingSupport.java 35388 (+5 -5) *diffs*

□ 35381 created by mstudman on 11 May 2009, 04:43:43 -0500 (41 hours ago)
CEP-326: Ignoring CCE problem when plugin is upgraded but Eclipse is not yet restarted.

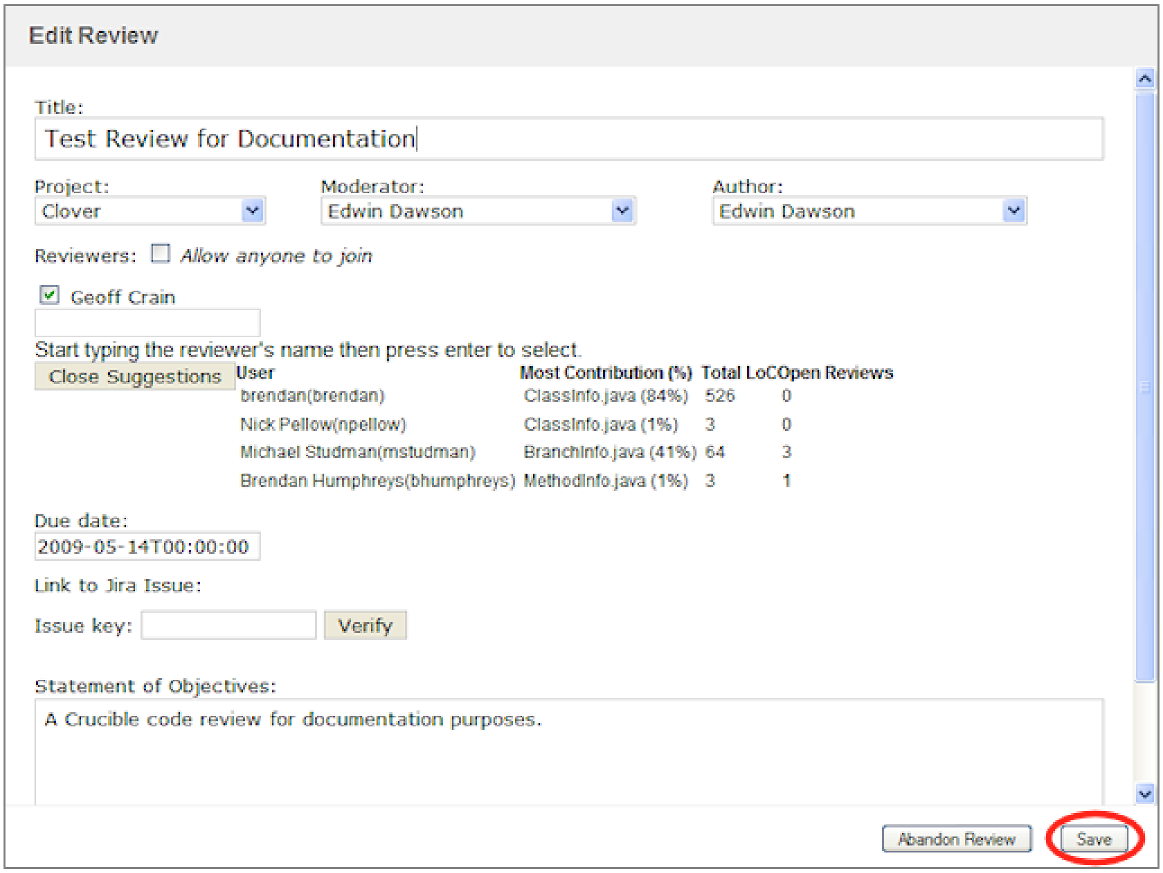
□ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/projects/model/CoverageModelsMonitor.java 35381 (+4 -0) *diffs*

□ 35367 created by npellow on 11 May 2009, 01:37:00 -0500 (45 hours ago)
NONE: creating branch for 2.5.x
Note: only showing 100 of 2997 files in this changeset.

□ /branches/clover-2.5.x 35367 (+0) *new*
□ /branches/clover-2.5.x/build.xml 35367 (+69) *new*
□ /branches/clover-2.5.x/lib 35367 (+0) *new*
□ /branches/clover-2.5.x/lib/junit-3.8.1.jar 35367 (+0) *new*

In the Manage Files dialog, the author selects the source files they want to include in the review, by clicking the checkboxes next to the desired files. Once finished, the author clicks '**Done**'. The Edit Review dialog appears, where the author can create and issue the review.

Screenshot: Creating a review in the Edit Review dialog



Edit Review

Title:
Test Review for Documentation

Project: Clover
Moderator: Edwin Dawson
Author: Edwin Dawson

Reviewers: □ *Allow anyone to join*

☑ Geoff Crain

Start typing the reviewer's name then press enter to select.

User	Most Contribution (%)	Total LoC	Open Reviews
brendan(brendan)	ClassInfo.java (84%)	526	0
Nick Pellow(npellow)	ClassInfo.java (1%)	3	0
Michael Studman(mstudman)	BranchInfo.java (41%)	64	3
Brendan Humphreys(bhumphreys)	MethodInfo.java (1%)	3	1

Due date:
2009-05-14T00:00:00

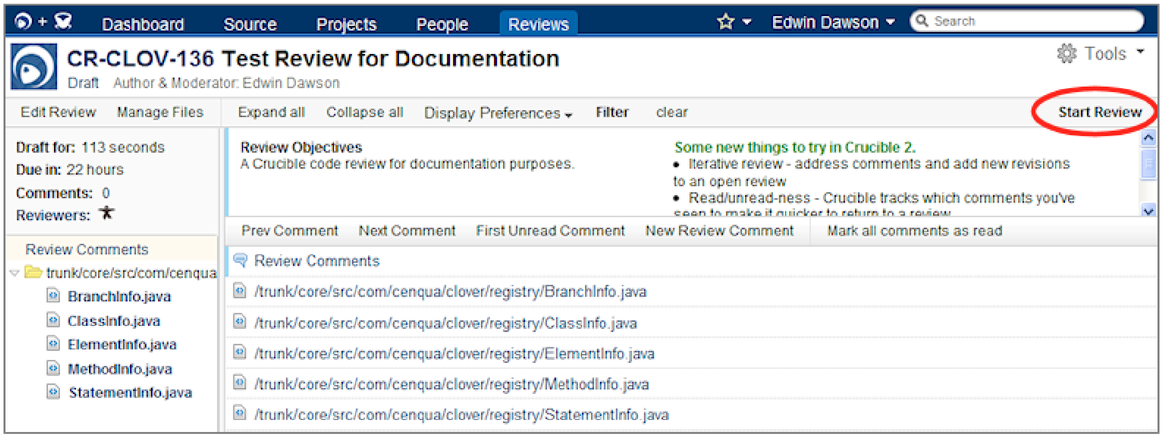
Link to Jira Issue:

Issue key:

Statement of Objectives:
A Crucible code review for documentation purposes.

In the Edit Review dialog, the author enters information needed for the review. This includes entering a title and description for the review, selecting reviewers, a due date and the key for a related JIRA issue (if any). The project, moderator and author are pre-selected (for this example, the author should select himself as a moderator. When finished, the author clicks '**Save**'. The review will now be created in a draft form.

Screenshot: A newly created Crucible review



CR-CLOV-136 Test Review for Documentation

Draft   Author & Moderator: Edwin Dawson

Draft for: 113 seconds
Due in: 22 hours
Comments: 0
Reviewers: ✱

Review Objectives
A Crucible code review for documentation purposes.

Some new things to try in Crucible 2.
- Iterative review - address comments and add new revisions to an open review
- Read/unread-ness - Crucible tracks which comments you've seen to make it quicker to return to a review

Review Comments
trunk/core/src/com/cenqua
BranchInfo.java
ClassInfo.java
ElementInfo.java
MethodInfo.java
StatementInfo.java

Review Comments
/trunk/core/src/com/cenqua/clover/registry/BranchInfo.java
/trunk/core/src/com/cenqua/clover/registry/ClassInfo.java
/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java
/trunk/core/src/com/cenqua/clover/registry/MethodInfo.java
/trunk/core/src/com/cenqua/clover/registry/StatementInfo.java

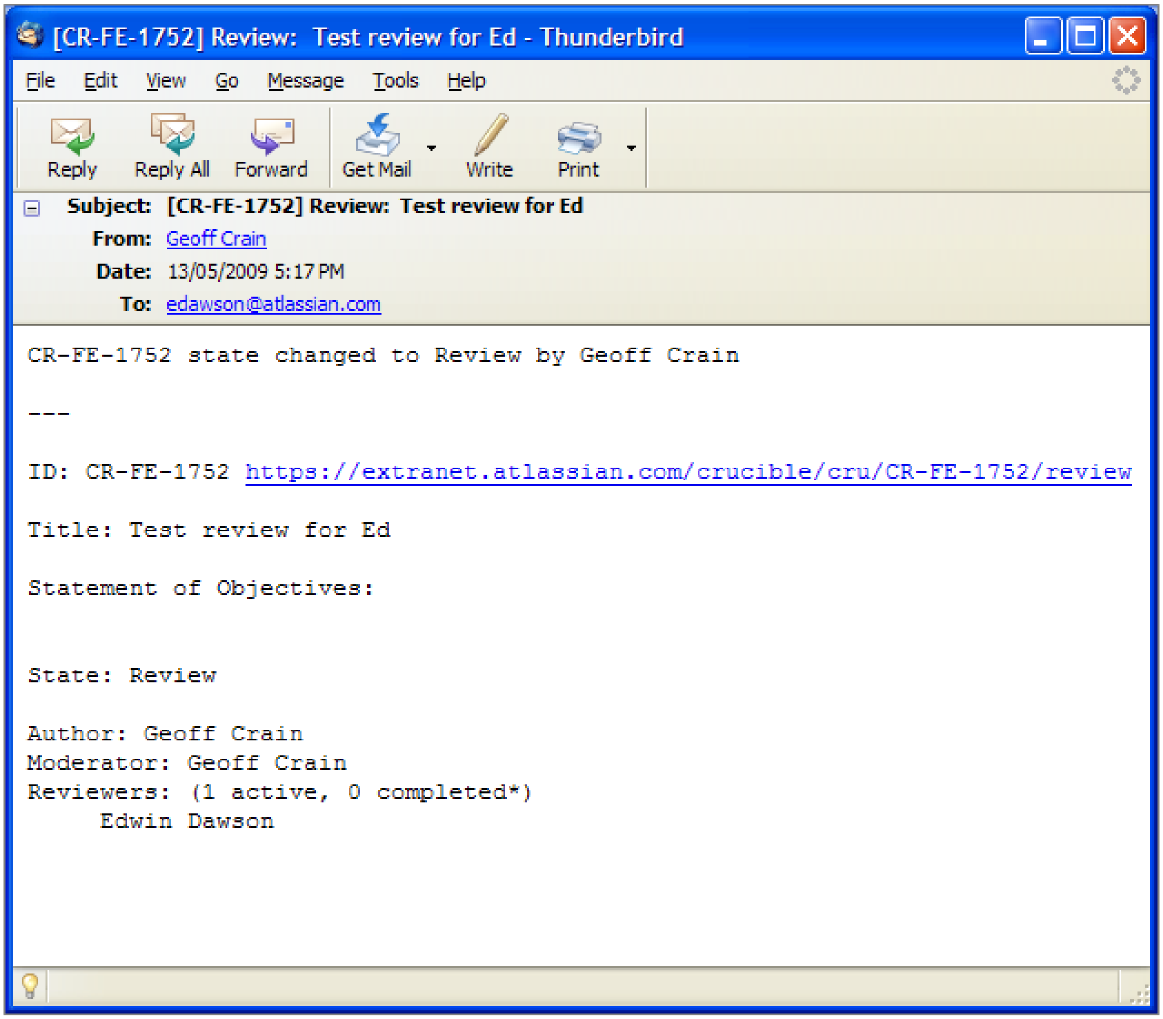
The draft review opens. In the draft stage, the author can check the contents of the review files to ensure they are correct and put in any notes for reviewers as comments. During the draft phase, no notification email is sent out to reviewers. Once the author is finished with the draft phase, he clicks '**Start Review**'.

The review will now be started and notification email will go out to all participants. Crucible will now send out an email notification to all the participants. This lets them know that the review is under way and prompts them to take action, providing a URL for direct access to the review. (You can also subscribe to an RSS feed.)

2. The Reviewer Comments on the Code

The reviewer will receive an email from Crucible (or an RSS feed update) with a link that they can follow to the review.

Screenshot: A Crucible review notification email

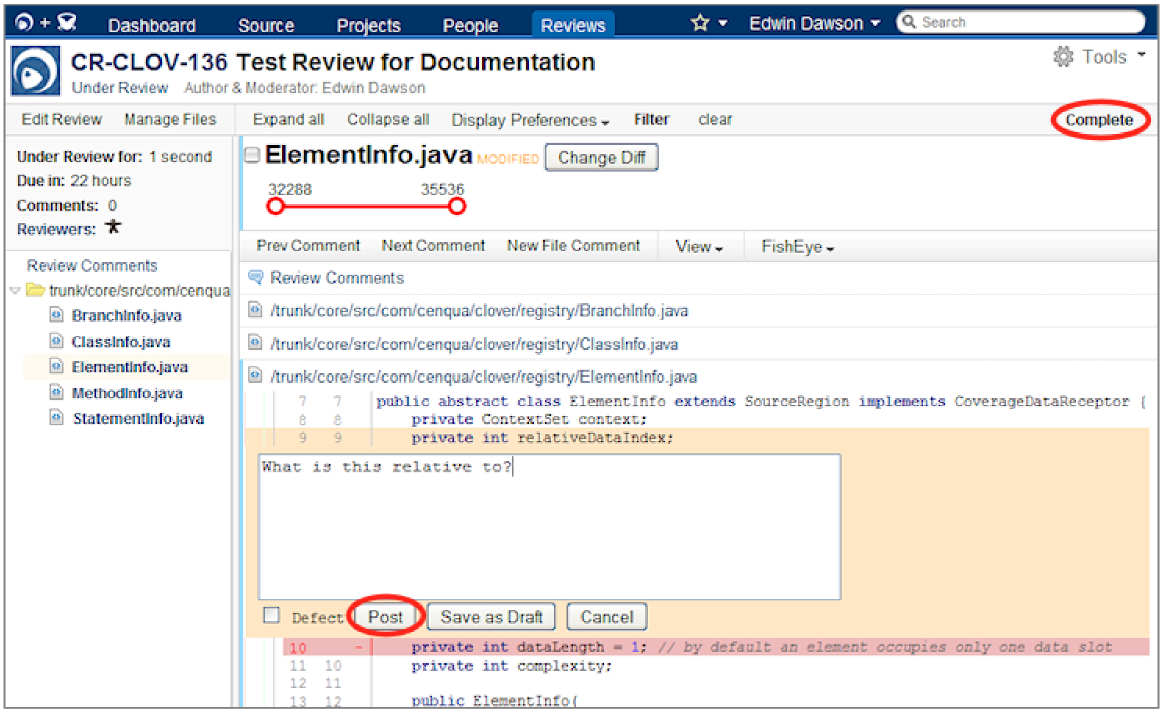

[CR-FE-1752] Review: Test review for Ed - Thunderbird
File Edit View Go Message Tools Help
Reply
Reply All
Forward
Get Mail
Write
Print

Subject: [CR-FE-1752] Review: Test review for Ed
From: Geoff Crain
Date: 13/05/2009 5:17 PM
To: edawson@atlassian.com

```
CR-FE-1752 state changed to Review by Geoff Crain

---

ID: CR-FE-1752 https://extranet.atlassian.com/crucible/cru/CR-FE-1752/review

Title: Test review for Ed

Statement of Objectives:


State: Review

Author: Geoff Crain
Moderator: Geoff Crain
Reviewers: (1 active, 0 completed*)
     Edwin Dawson
```

When the reviewer clicks the link in the notification email, the Crucible Review screen opens.

Screenshot: The Crucible Review screen



```
public abstract class ElementInfo extends SourceRegion implements CoverageDataReceptor {
    private ContextSet context;
    private int relativeDataIndex;
```

What is this relative to?

```
-   private int dataLength = 1; // by default an element occupies only one data slot
    private int complexity;

    public ElementInfo(
```

On the Crucible Review screen, the code changes under review are displayed. The reviewer clicks filenames to expand the code for in-line reviewing. As the reviewer reads the changes, they can simply click on any line to enter a comment there (multiple lines can be selected by clicking and dragging).

The reviewer clicks the '**Post**' button when each comment is finished.
The reviewer repeats this process for all files in the review. Reviewers can leave the session and resume it later; their work is automatically saved.

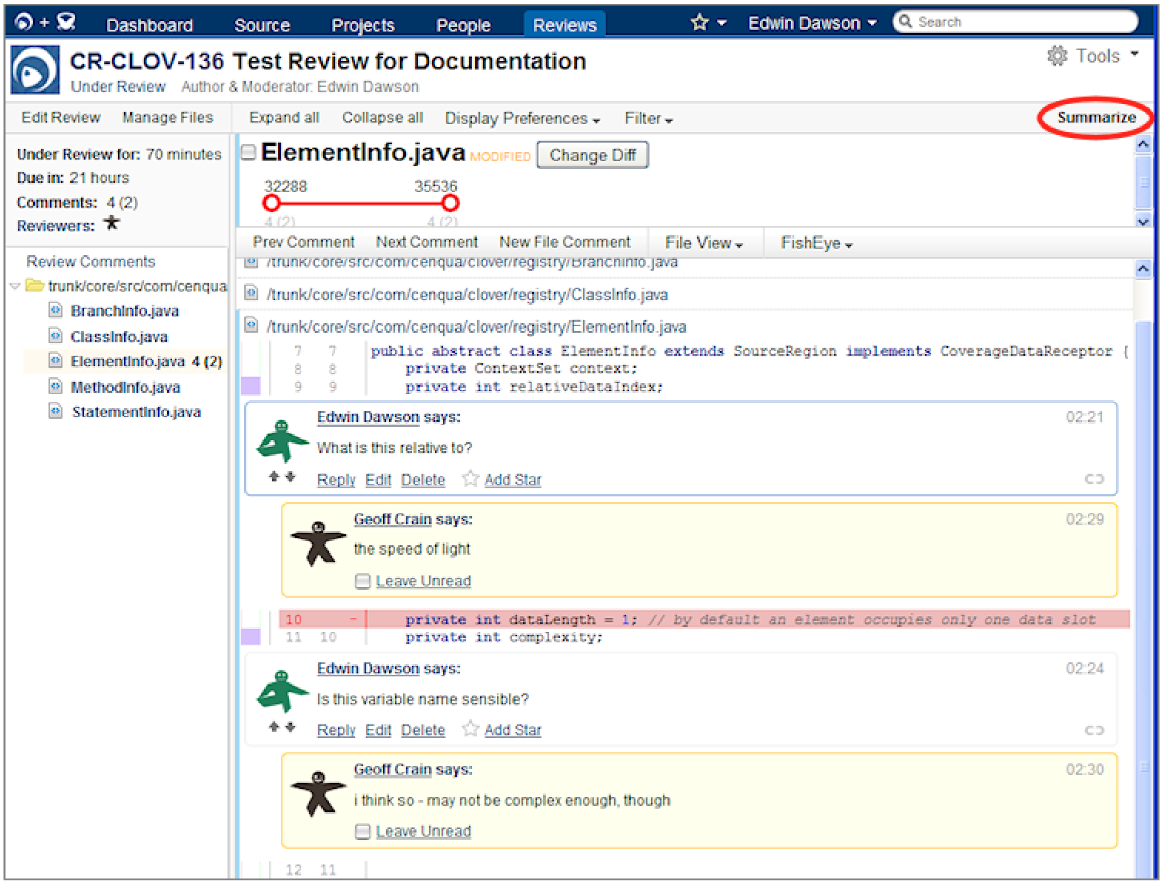
When the reviewer has finished their code review work, they click the Complete button.

> By default, an email is sent to participants every time a comment is posted. This is an individual setting. Each reviewer can configure their own profiles to adjust the list of events that will trigger email notifications.

3. The Author Responds to the Comments

During the review process, the author/moderator can also make contributions, responding to reviewer comments and making corrections.

Screenshot: Comment threads in Crucible



CR-CLOV-136 Test Review for Documentation

Under Review   Author & Moderator: Edwin Dawson

Under Review for: 70 minutes
Due in: 21 hours
Comments: 4 (2)
Reviewers:

trunk/core/src/com/cenqua
- BranchInfo.java
- ClassInfo.java
- ElementInfo.java 4 (2)
- MethodInfo.java
- StatementInfo.java

ElementInfo.java MODIFIED

32288   35536

/trunk/core/src/com/cenqua/clover/registry/BranchInfo.java
/trunk/core/src/com/cenqua/clover/registry/ClassInfo.java
/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java

```
7   7   public abstract class ElementInfo extends SourceRegion implements CoverageDataReceptor {
8   8       private ContextSet context;
9   9       private int relativeDataIndex;
```

Edwin Dawson says: 02:21
What is this relative to?

Geoff Crain says: 02:29
the speed of light

```
10   -      private int dataLength = 1; // by default an element occupies only one data slot
11  10      private int complexity;
```

Edwin Dawson says: 02:24
Is this variable name sensible?

Geoff Crain says: 02:30
i think so - may not be complex enough, though

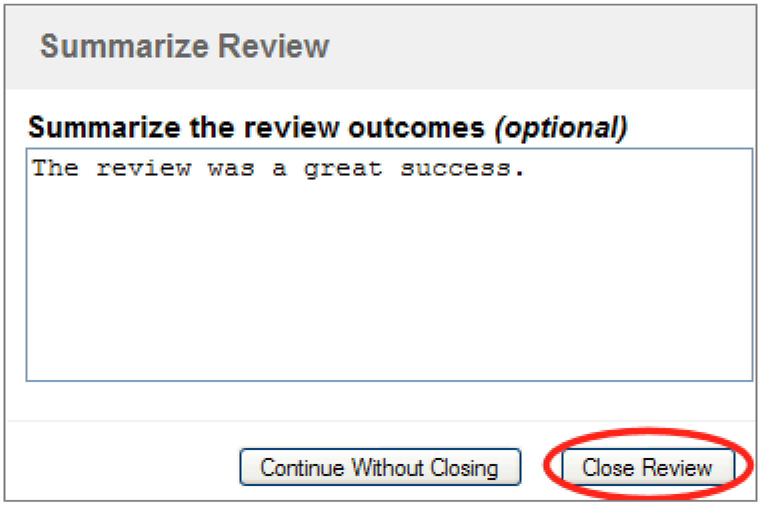
```
12  11
```

4. The Author Closes the Review

When all reviewers have *Completed* their reviews, the author/moderator is notified via email. The author/moderator clicks the link in the notification email, returning to the Review screen.

The author/moderator will then add any final comments, then click the Summarize button when finished. The Crucible Summarize Review screen opens.

Screenshot: Summarizing a review in Crucible



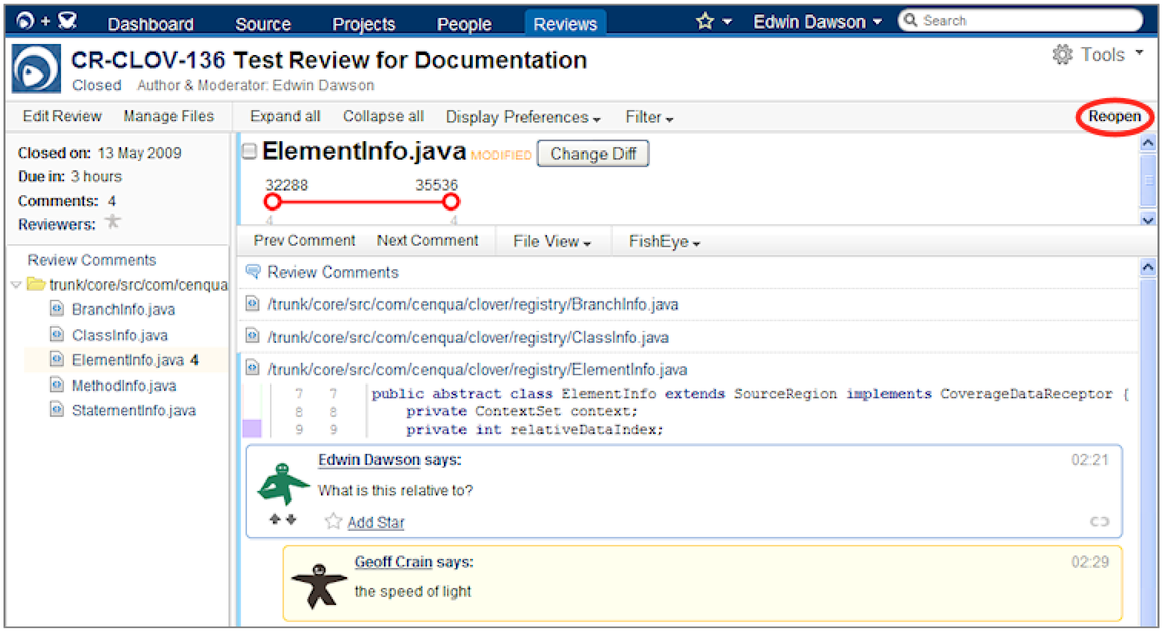
Summarize Review

Summarize the review outcomes *(optional)*

The review was a great success.

On the Crucible Summarize Review screen, the author/moderator enters an optional summary of the review's results, then clicks Close Review. This closes the review, signalling the end of work. A final email notification will be sent to the review participants, informing them that the review is now closed. The closed review screen will load, displaying the summary and archiving the completed review as read-only.

Screenshot: Viewing a closed review



Dashboard
Source
Projects
People
Reviews
Edwin Dawson
Search
Tools
CR-CLOV-136 Test Review for Documentation
Closed Author & Moderator: Edwin Dawson
Edit Review
Manage Files
Expand all
Collapse all
Display Preferences
Filter
Reopen
ElementInfo.java MODIFIED
Change Diff
32288
35536
Prev Comment
Next Comment
File View
FishEye
Review Comments
trunk/core/src/com/cenqua
BranchInfo.java
ClassInfo.java
ElementInfo.java 4
MethodInfo.java
StatementInfo.java
/trunk/core/src/com/cenqua/clover/registry/BranchInfo.java
/trunk/core/src/com/cenqua/clover/registry/ClassInfo.java
/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java
Add Star

Closed on: 13 May 2009
Due in: 3 hours
Comments: 4
Reviewers:

```
public abstract class ElementInfo extends SourceRegion implements CoverageDataReceptor {
    private ContextSet context;
    private int relativeDataIndex;
```

Edwin Dawson says: 02:21
What is this relative to?

Geoff Crain says: 02:29
the speed of light

If the author/moderator ever needs to resume work on the closed review, they can simply click Reopen when viewing this screen. Doing this will return the review's status to "open".

For more information on workflow in Crucible and best practices for code reviews, see Requesting and Conducting a Review.

Using the Crucible Screens

This page contains an overview of the Crucible interface and the actions that can be carried out in the application.

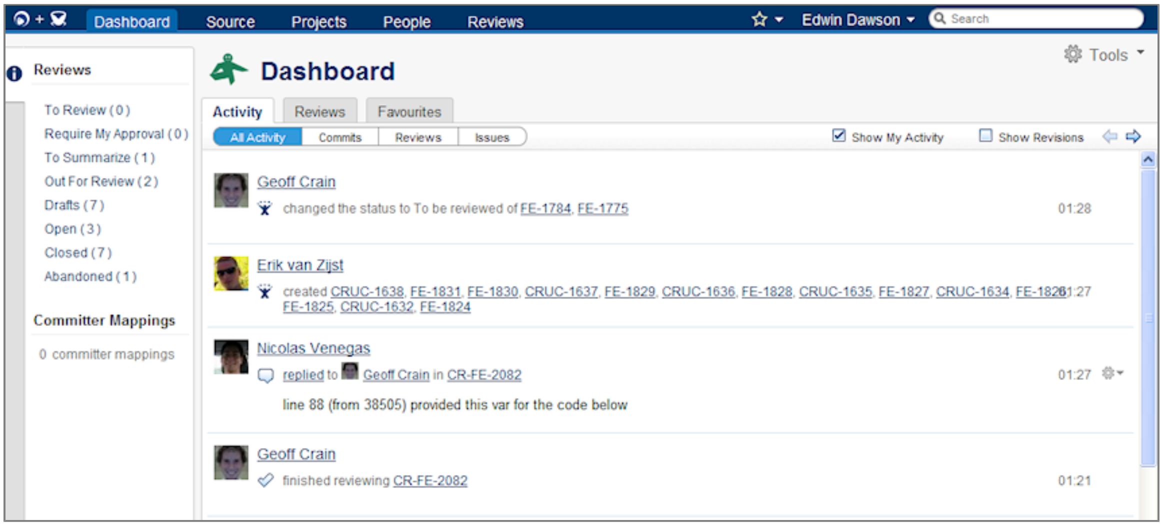
On this page:

- Tour of the Crucible Interface
- Left Navigation Sidebar
- Related Links

Tour of the Crucible Interface

When you first log in to Crucible, the Dashboard Screen opens, as shown in the screenshot below. This view shows recent general activity in Crucible.

Screenshot: The Dashboard Screen in Crucible



Geoff Crain
changed the status to To be reviewed of FE-1784, FE-1775 01:28

Erik van Zijst
created CRUC-1638, FE-1831, FE-1830, CRUC-1637, FE-1829, CRUC-1636, FE-1828, CRUC-1635, FE-1827, CRUC-1634, FE-1826 01:27
FE-1825, CRUC-1632, FE-1824

Nicolas Venegas
replied to Geoff Crain in CR-FE-2082 01:27
line 88 (from 38505) provided this var for the code below

Geoff Crain
finished reviewing CR-FE-2082 01:21

The table below explain the top-level tabs in the Crucible User Interface. Click on the name of a tab for more information.

Element name	Function	Appears
Dashboard Tab	Displays reviews and system activity related to you.	All screens.
Source Tab	Displays contents of connected source repositories.	Only when FishEye is used with Crucible.
Projects Tab	Displays reviews and content from specific projects.	All screens.
People Tab	Displays metrics on the users of the Crucible instance.	All screens.
Reviews Tab	Allows you to search and report on reviews.	All screens.

Left Navigation Sidebar

The navigation bar at the left of the screen applies specific filters to what is shown in the centre pane. See the page on Browsing Your Reviews for more information.

ⓘ The left navigation sidebar can be hidden or displayed by clicking the blue 'information' icon ⓘ at the top left of the sidebar.

Related Links

Browsing Your Reviews
Browsing Source Files
Browsing Projects
Viewing People's Statistics
Viewing Reports
Searching in Crucible
Using RSS Feeds in Crucible
Changing your User Profile

Browsing Your Reviews

This page contains information on browsing reviews in Crucible.

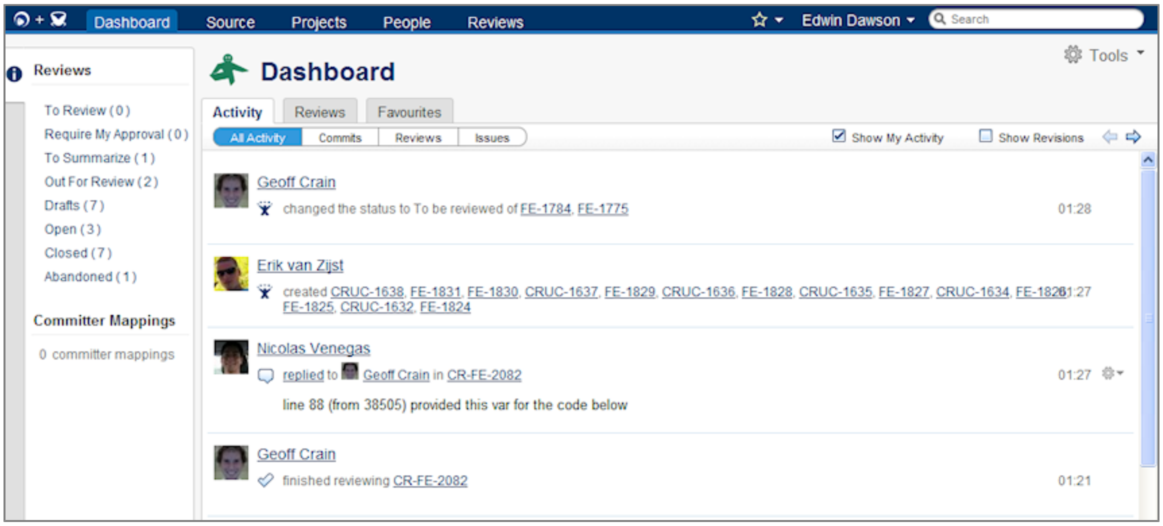
On this page:

- Dashboard Screen Overview
    - Filtering Your View

Dashboard Screen Overview

To browse all your reviews (reviews you are participating in), use the **Dashboard** tab at the top of the page. The Dashboard Screen opens, as shown in the screenshot below.

Screenshot: The Dashboard Screen in Crucible



Geoff Crain
changed the status to To be reviewed of FE-1784, FE-1775 01:28

Erik van Zijst
created CRUC-1638, FE-1831, FE-1830, CRUC-1637, FE-1829, CRUC-1636, FE-1828, CRUC-1635, FE-1827, CRUC-1634, FE-1826, FE-1825, CRUC-1632, FE-1824 01:27

Nicolas Venegas
replied to Geoff Crain in CR-FE-2082 01:27
line 88 (from 38505) provided this var for the code below

Geoff Crain
finished reviewing CR-FE-2082 01:21

By default, the Activity Stream is shown. This is a mix of all activity that is occurring related to Crucible, such as people making review comments, reviews opening and closing, files being committed to a linked repository, or updates to linked JIRA issues.

Filtering Your View

To filter your view, use the constraint options in the sub-nav and the side panel.

Activity Stream filters

You can also filter the items shown in the Activity Stream. To do this, click one of the options in second layer of tabs, as listed in the table below.

Tab Name	Sub-Nav Options
Activity Tab	Sub-Nav Options: - **All Activity** — Shows all activity with no filtering. - **Commits** — Shows commits. - **Reviews** — Shows reviews. - **Issues** — Shows JIRA issues. - **Show/Hide My Activity** — Show/hide your own activity. - **Show/Hide Revisions** — Show/hide activity from other users. - Earlier / Later Activity (arrow buttons) — Pages through activity items, stepping backward or forward through the results.

Reviews Tab (Shows the result of the filter clicked in the left side panel)	Sub-Nav Options: **RSS** — Opens a page with the RSS feed for the current selection.
Favourites Tab (Shows all items you have marked as a 'favourite'.)	Sub-Nav Options: None.

Side Panel

The left navigation panel of the Dashboard shows the number of reviews in different states. Click on any of these states to show the list of reviews in the left-hand panel.

To Review	Reviews where the user still needs to complete their work.
Require My Approval	The user has been assigned the role of moderator for these reviews and needs to approve them.
To Summarize	The user has been assigned the role of moderator for these reviews and needs to summarise and close them.
Out For Review	Reviews created by the user that are currently in progress.
Drafts	These are reviews created by the user that have not yet been moved to the 'Approval' or the 'Require Approval' states.
Open	All open reviews that the user is participating in.
Closed	These are reviews that the user has been involved in and are now closed.
Abandoned	Reviews that are no longer relevant and can be deleted.

Browsing Source Files

When FishEye is installed with Crucible, you have the additional **'Source'** tab available in the navigation tabs at the top of the screen.

To view source files, click the '**Source**' tab. The '**Repositories**' view opens, showing summary information if you have multiple repositories set up. Click on the desired repository to view its contents.

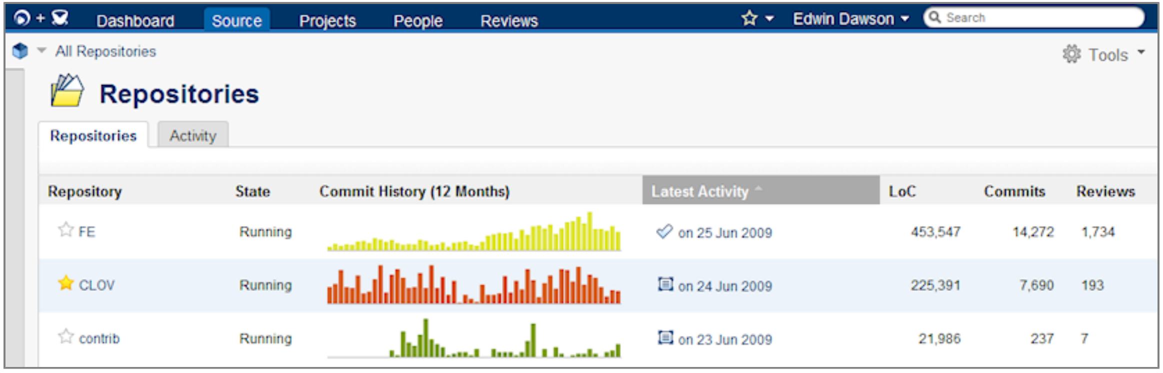
The file explorer mode of the source view will open. Here, you can navigate through the repository by selecting files and folders on the tree in the left navigation bar. When you reach a source file, a summary page is shown, displaying recent revisions to the file.

At this file level, there are five sub-tabs you can select. Their functions are outlined in the following table.

Sub-Tab Name	Description
Revisions	When viewing a file, shows the latest revisions of the file.
Files	When viewing a folder, shows the contents of the directory.

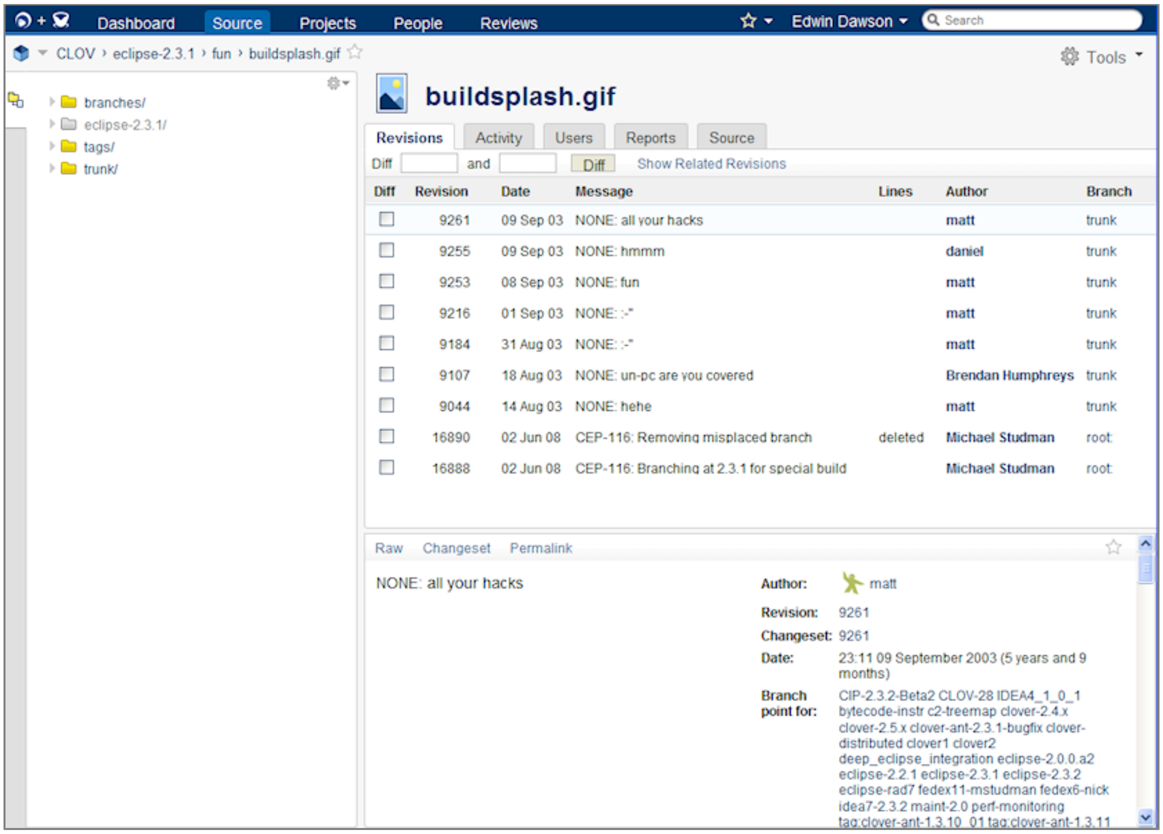
Activity	Shows recent activity on the item. There are a number of sub-options here: - All Activity — The default view, showing commits, reviews and JIRA issues. - Commits — Shows commits in the activity stream. - Reviews — Shows review activity in the activity stream. - Scroll to Changeset — Opens the changeset ID specified in the text field (press Enter to carry out the action). - Filter — Applies constraints to the current activity stream. - Show Revisions — If this is selected, then changeset items are automatically expanded to show modified files. - Earlier Activity (Left Arrow icon) — Loads a page of earlier activity. - Later Activity (Right Arrow icon) — Loads a page of later activity.
Users	Shows the commit history of the different users that have committed changes on the item.
Reports	Shows activity charts for the item. Various chart options can be selected in the left navigation bar.
Source	Shows the contents of the file.
Query	Allows you to run an advanced search.

Screenshot: The Repositories View



Repository	State	Commit History (12 Months)	Latest Activity	LoC	Commits	Reviews
FE	Running		on 25 Jun 2009	453,547	14,272	1,734
CLOV	Running		on 24 Jun 2009	225,391	7,690	193
contrib	Running		on 23 Jun 2009	21,986	237	7

Screenshot: The Source View Item Activity



buildsplash.gif

Diff	Revision	Date	Message	Lines	Author	Branch
□	9261	09 Sep 03	NONE: all your hacks		matt	trunk
□	9255	09 Sep 03	NONE: hmmm		daniel	trunk
□	9253	08 Sep 03	NONE: fun		matt	trunk
□	9216	01 Sep 03	NONE: :-"		matt	trunk
□	9184	31 Aug 03	NONE: :-"		matt	trunk
□	9107	18 Aug 03	NONE: un-pc are you covered		Brendan Humphreys	trunk
□	9044	14 Aug 03	NONE: hehe		matt	trunk
□	16890	02 Jun 08	CEP-116: Removing misplaced branch	deleted	Michael Studman	root:
□	16888	02 Jun 08	CEP-116: Branching at 2.3.1 for special build		Michael Studman	root:

NONE: all your hacks

Author: matt
Revision: 9261
Changeset: 9261
Date: 23:11 09 September 2003 (5 years and 9 months)
Branch point for: CIP-2.3.2-Beta2 CLOV-28 IDEA4_1_0_1 bytecode-instr c2-treemap clover-2.4.x clover-2.5.x clover-ant-2.3.1-bugfix clover-distributed clover1 clover2 deep_eclipse_integration eclipse-2.0.0.a2 eclipse-2.2.1 eclipse-2.3.1 eclipse-2.3.2 eclipse-rad7 fedex11-mstudman fedex6-nick idea7-2.3.2 maint-2.0 perf-monitoring tag:clover-ant-1.3.10_01 tag:clover-ant-1.3.11

Browsing Projects

To browse the content in a project, click the **Projects** tab at the top of the page. The '**Projects**' view opens.

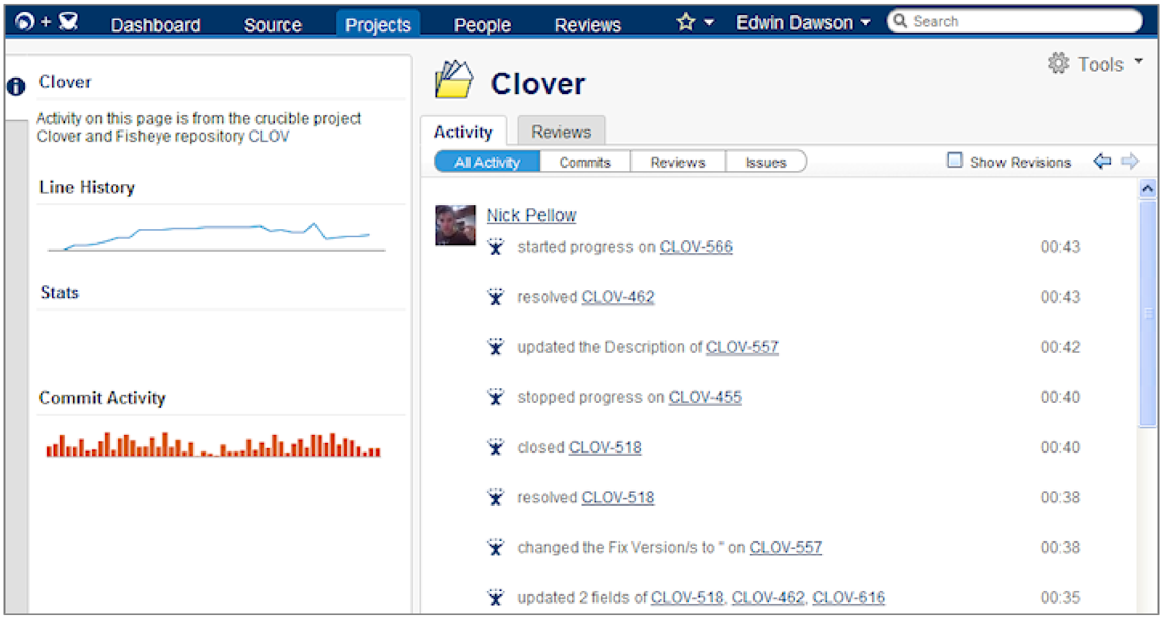
A list of projects will be shown if there is more than one. Click the name of the desired project to open it. The '**Project Activity**' page opens. In the left navigation bar, charts showing overall project statistics are displayed.

There are a number of sub-tabs on this page, listed in the table below.

Sub-Tab Name	Description
Activity	• All Activity — The default view. • Commits — Shows commits in the project (visible when using FishEye). • Reviews — Shows reviews in the project. • Issues — Shows JIRA issues related to this project. ℹ Only visible if you have set up JIRA Integration in Crucible. • Show Revisions — Shows or hides revisions in the project (visible when using FishEye). • Earlier Activity (Left Arrow icon) — Loads a page of earlier project activity. • Later Activity (Right Arrow icon) — Loads a page of later project activity.
Reviews	Shows recent reviews in the project.

ℹ The Projects tab is only visible in Crucible. Read more about the definition of a project.

Screenshot: The Crucible Projects View



Clover

Activity on this page is from the crucible project Clover and Fisheye repository CLOV

Line History

Stats

Commit Activity

Clover

Nick Pellow

started progress on CLOV-566 00:43

resolved CLOV-462 00:43

updated the Description of CLOV-557 00:42

stopped progress on CLOV-455 00:40

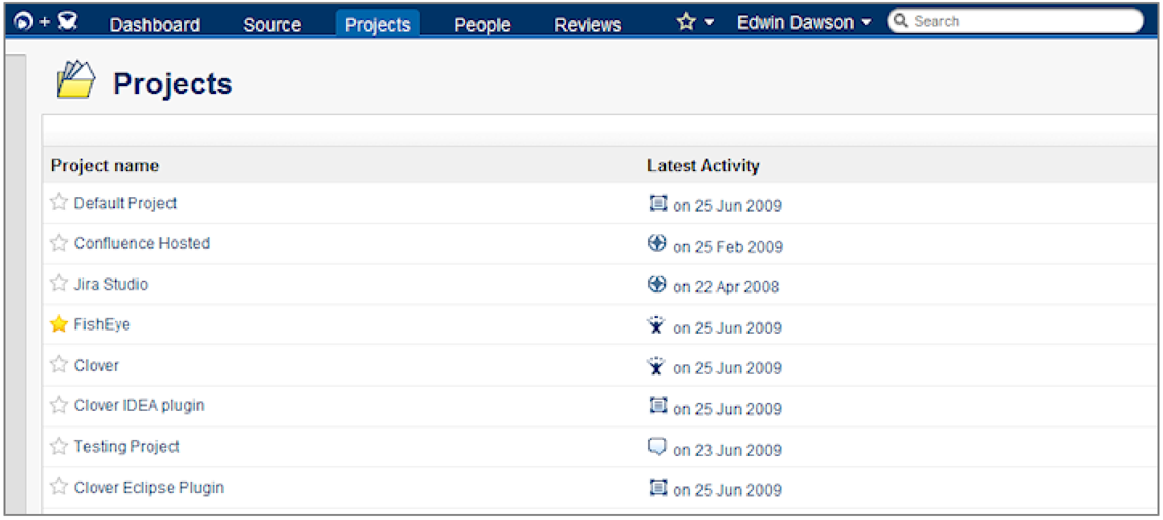
closed CLOV-518 00:40

resolved CLOV-518 00:38

changed the Fix Version/s to '' on CLOV-557 00:38

updated 2 fields of CLOV-518, CLOV-462, CLOV-616 00:35

Screenshot: The Crucible Projects Index



Projects

Project name	Latest Activity
☆ Default Project	on 25 Jun 2009
☆ Confluence Hosted	on 25 Feb 2009
☆ Jira Studio	on 22 Apr 2008
★ FishEye	on 25 Jun 2009
☆ Clover	on 25 Jun 2009
☆ Clover IDEA plugin	on 25 Jun 2009
☆ Testing Project	on 23 Jun 2009
☆ Clover Eclipse Plugin	on 25 Jun 2009

Viewing Project Statistics

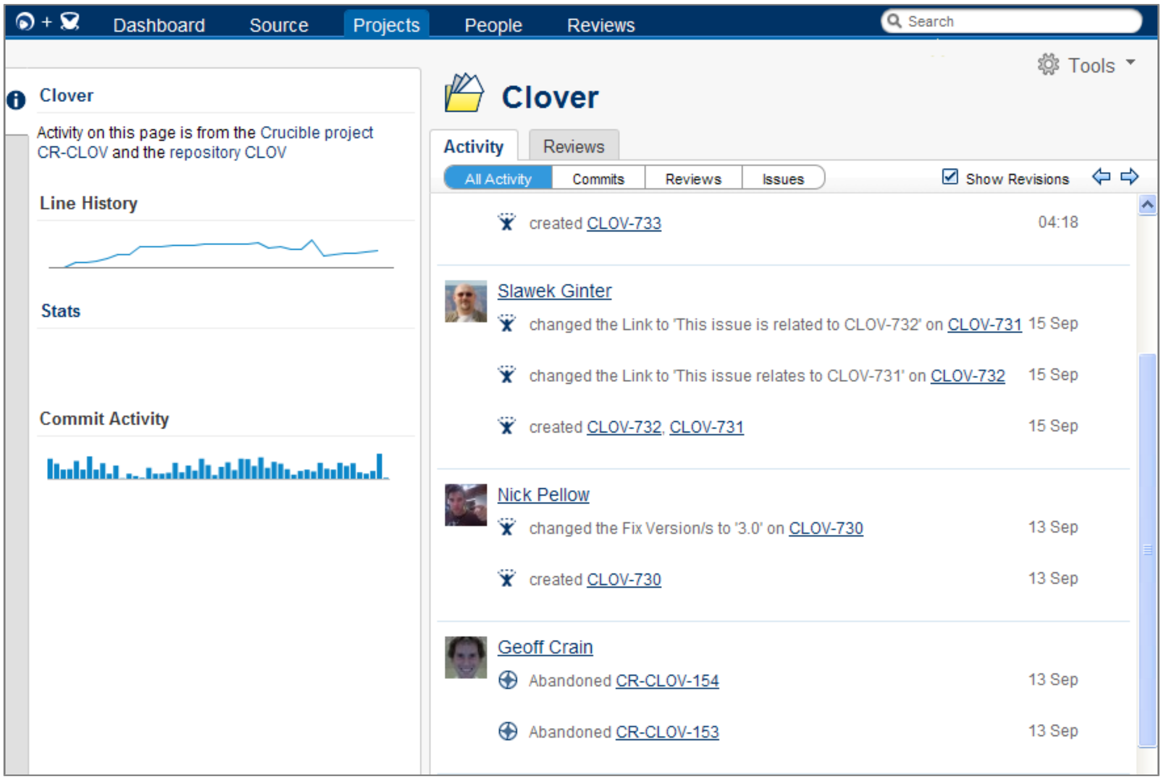
This page explains the layout of the Project Summary page.

On this page:

- Project Name Panel
- Project Line History Panel
- Project Stats Panel
- Project Commit Activity Chart

When you click through to a Crucible Project from the Projects Tab, the '**Project Summary**' screen opens.

Screenshot: The Crucible Project Summary Page



Clover

Activity on this page is from the Crucible project CR-CLOV and the repository CLOV

Clover

created CLOV-733 04:18

Slawek Ginter

changed the Link to 'This issue is related to CLOV-732' on CLOV-731 15 Sep

changed the Link to 'This issue relates to CLOV-731' on CLOV-732 15 Sep

created CLOV-732, CLOV-731 15 Sep

Nick Pellow

changed the Fix Version/s to '3.0' on CLOV-730 13 Sep

created CLOV-730 13 Sep

Geoff Crain

Abandoned CR-CLOV-154 13 Sep

Abandoned CR-CLOV-153 13 Sep

In the right hand pane, you can see an activity stream relating to this project. In the left hand pane, you can see various statistics charts relating to the project in context. These appear in a reduced size until you click them, when they will expand to show more information.

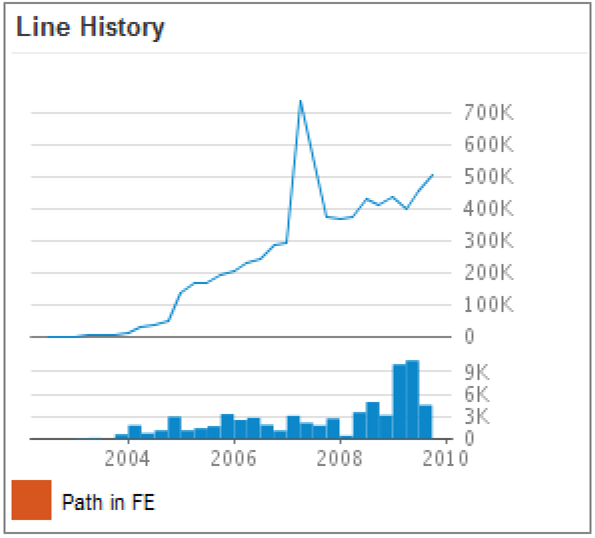
Project Name Panel

This contains a short message explaining which Crucible Project and FishEye repositories are being accessed to show the activity stream on the page.

Project Line History Panel

This panel contains a chart showing the lines of code added to the repository, graphed over time.

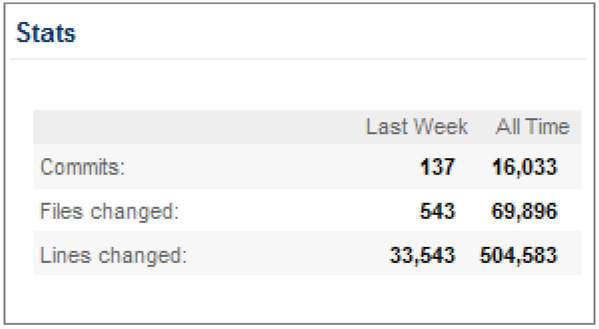
Screenshot: The Project Line History Panel



Project Stats Panel

This panel contains a chart showing numerical data for commits, files changed and lines change, graphed over time.

Screenshot: The Project Stats Panel

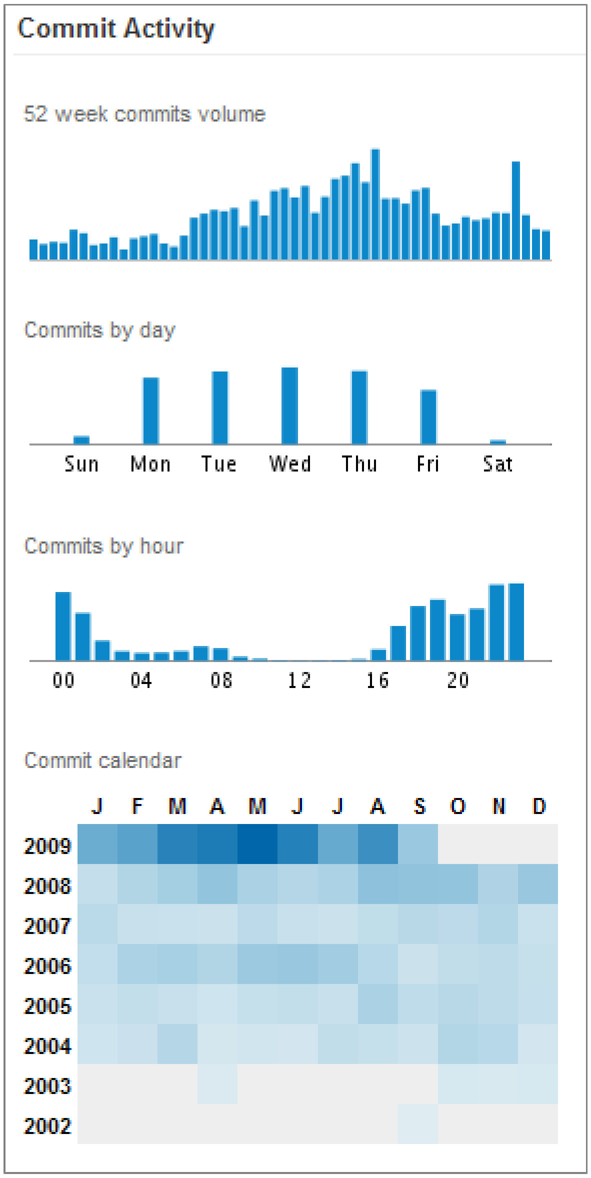


Stats

	Last Week	All Time
Commits:	**137**	**16,033**
Files changed:	**543**	**69,896**
Lines changed:	**33,543**	**504,583**

Project Commit Activity Chart

This panel contains a number of charts:

Chart	Description
 Commit Activity 52 week commits volume Commits by day Sun Mon Tue Wed Thu Fri Sat Commits by hour 00 04 08 12 16 20 Commit calendar J F M A M J J A S O N D 2009 2008 2007 2006 2005 2004 2003 2002	**52 week commits volume** This chart shows the amount of commits, shown by week over a one year period. **Commits by day** This chart shows the amount of commits, graphed by day over the past week. **Commits by hour** This chart shows the amount of commits, graphed by hours over the past day. **Commit calendar** This chart shows the amount of commits (shown as darker colours to indicate more commits) graphed by month, over years that the repository has been running.

Viewing People's Statistics

This page contains instructions on how to use the People tab in Crucible to see charts and activity from people with accounts on the system.

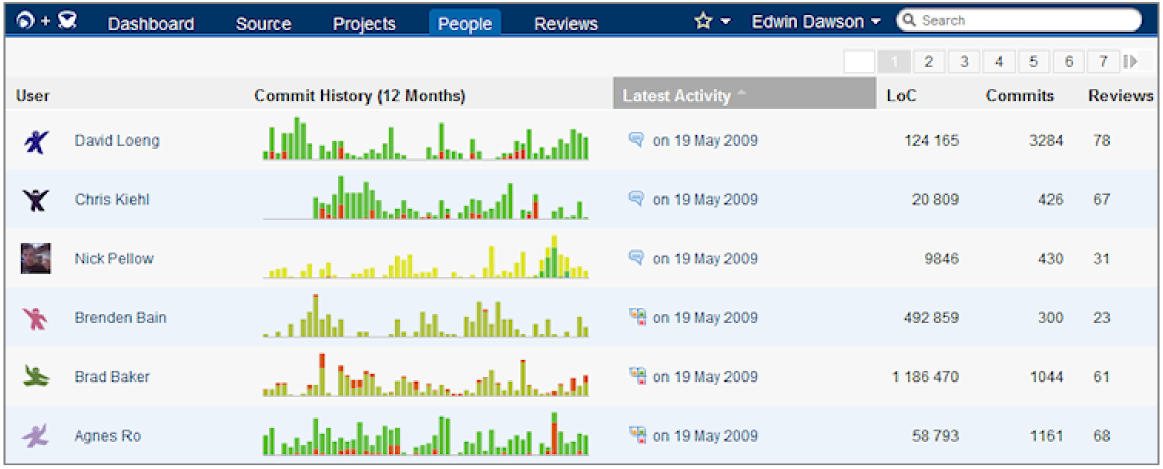
On this page:

- Opening the List of People
- Viewing a Person's Activity Screen
- Viewing Charts on a Person's Activity

Opening the List of People

To view statistics on People in Crucible, (that is, code authors, committers and reviewers) click the **People** tab at the top of the page. The list of all People appears.

Screenshot: List of all People in Crucible (when using FishEye with Crucible)



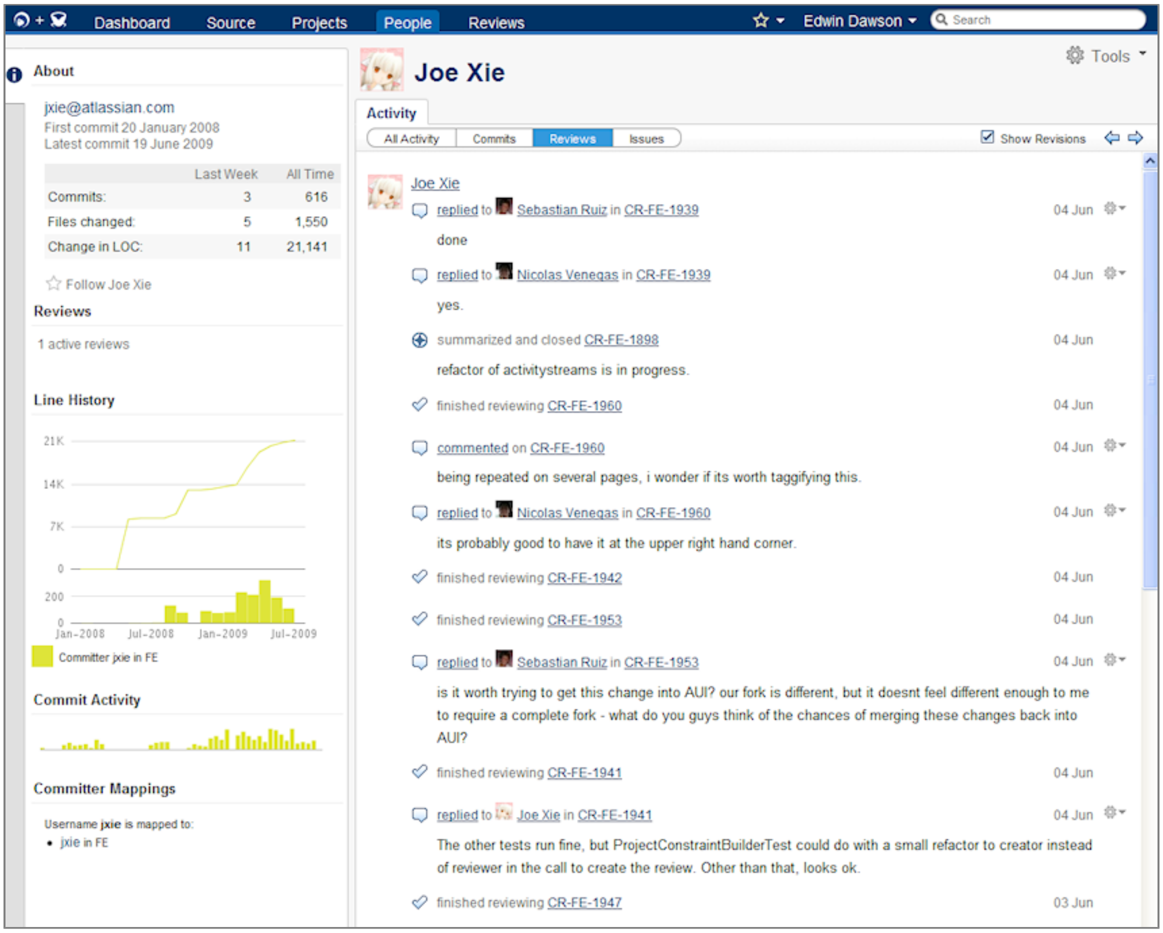
User	Commit History (12 Months)	Latest Activity	LoC	Commits	Reviews
David Loeng		on 19 May 2009	124 165	3284	78
Chris Kiehl		on 19 May 2009	20 809	426	67
Nick Pellow		on 19 May 2009	9846	430	31
Brenden Bain		on 19 May 2009	492 859	300	23
Brad Baker		on 19 May 2009	1 186 470	1044	61
Agnes Ro		on 19 May 2009	58 793	1161	68

The list of all people shows all users that have accounts on the system. By default, each user has a unique avatar that is randomly formed from the text in their email address. Users can choose to upload their own avatar image by uploading an image to an external service such as Gravatar, which Crucible supports. See the page on Changing your User Profile.

Viewing a Person's Activity Screen

Click on a user to see a listing of activity from them as well as charts showing statistics for their activity. The People Activity screen opens.

Screenshot: The People Activity Screen in Crucible



Joe Xie

About

jxie@atlassian.com
First commit 20 January 2008
Latest commit 19 June 2009

	Last Week	All Time
Commits:	3	616
Files changed:	5	1,550
Change in LOC:	11	21,141

☆ Follow Joe Xie

Reviews

1 active reviews

Line History

Commit Activity

Committer Mappings

Username jxie is mapped to:
- jxie in FE

Joe Xie

replied to Sebastian Ruiz in CR-FE-1939 — 04 Jun
done

replied to Nicolas Venegas in CR-FE-1939 — 04 Jun
yes.

summarized and closed CR-FE-1898 — 04 Jun
refactor of activitystreams is in progress.

finished reviewing CR-FE-1960 — 04 Jun

commented on CR-FE-1960 — 04 Jun
being repeated on several pages, i wonder if its worth taggifying this.

replied to Nicolas Venegas in CR-FE-1960 — 04 Jun
its probably good to have it at the upper right hand corner.

finished reviewing CR-FE-1942 — 04 Jun

finished reviewing CR-FE-1953 — 04 Jun

replied to Sebastian Ruiz in CR-FE-1953 — 04 Jun
is it worth trying to get this change into AUI? our fork is different, but it doesnt feel different enough to me to require a complete fork - what do you guys think of the chances of merging these changes back into AUI?

finished reviewing CR-FE-1941 — 04 Jun

replied to Joe Xie in CR-FE-1941 — 04 Jun
The other tests run fine, but ProjectConstraintBuilderTest could do with a small refactor to creator instead of reviewer in the call to create the review. Other than that, looks ok.

finished reviewing CR-FE-1947 — 03 Jun

In the right hand pane, we can see a list of all activity that relates to this user. You can click the icons to view full commit information in FishEye, click JIRA issue names to open the work ticket on an item, click the long button to see the list of files in context or click the star icon to add an item to your favourites.

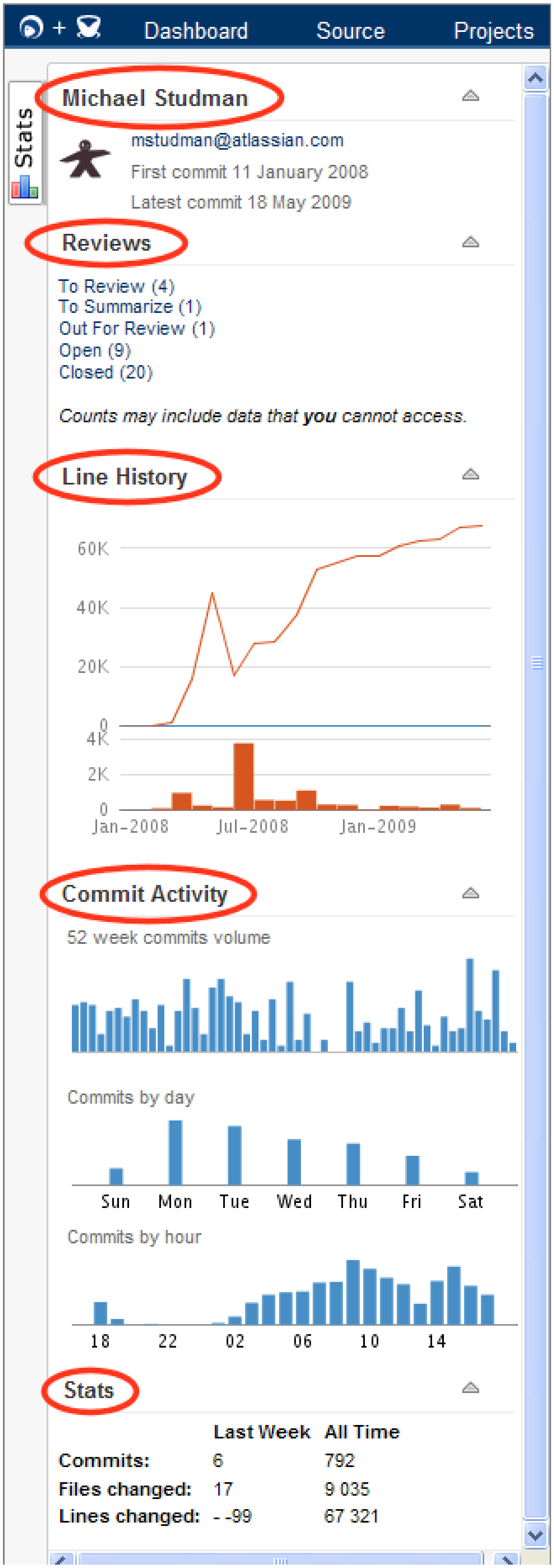
In the left hand pane, we can see charts around this activity, such as the following: number of active reviews; charted history of lines of code; code committing activity and general statistics.

Viewing Charts on a Person's Activity

To see information on a person's activity charted in detail, click the headings in the left-hand pane. Each heading will show more information on demand, when clicked. The information available and what it means is listed below.

⚠ The Charts in this section are only available when using FishEye.

Screenshot: People Activity Charts in Crucible



Michael Studman

mstudman@atlassian.com
First commit 11 January 2008
Latest commit 18 May 2009

Reviews

To Review (4)
To Summarize (1)
Out For Review (1)
Open (9)
Closed (20)

Counts may include data that you cannot access.

Line History

Commit Activity

52 week commits volume

Commits by day

Commits by hour

Stats

	Last Week	All Time
Commits:	6	792
Files changed:	17	9 035
Lines changed:	- -99	67 321

- **Username** heading:
  The username section shows the email address, then the first and latest commit dates for the person in context. It also shows username mappings from various systems if they have several usernames in play.

- **Reviews** heading:
  The Reviews section shows several filters that you can click to constrain the review items shown in the right-hand pane. The options are **To Review**, **To Summarize**, **Out For Review**, **Open** and **Closed**.

- **Line History** heading:
  The Line History section shows a graph with the number of lines committed to the repository, charted over time.

- **Commit Activity** heading:
  The Commit Activity section shows three smaller charts; the first showing the volume of commits over a 52 week period; the second showing the relative number of commits on days of the week; the third showing the relative number of commits by the hour of the day when they were lodged.

- **Stats** heading:
  The Stats section shows data points for the previous week and all-time. It shows number of commits, number of files changed and number of lines changed.

Viewing Reports

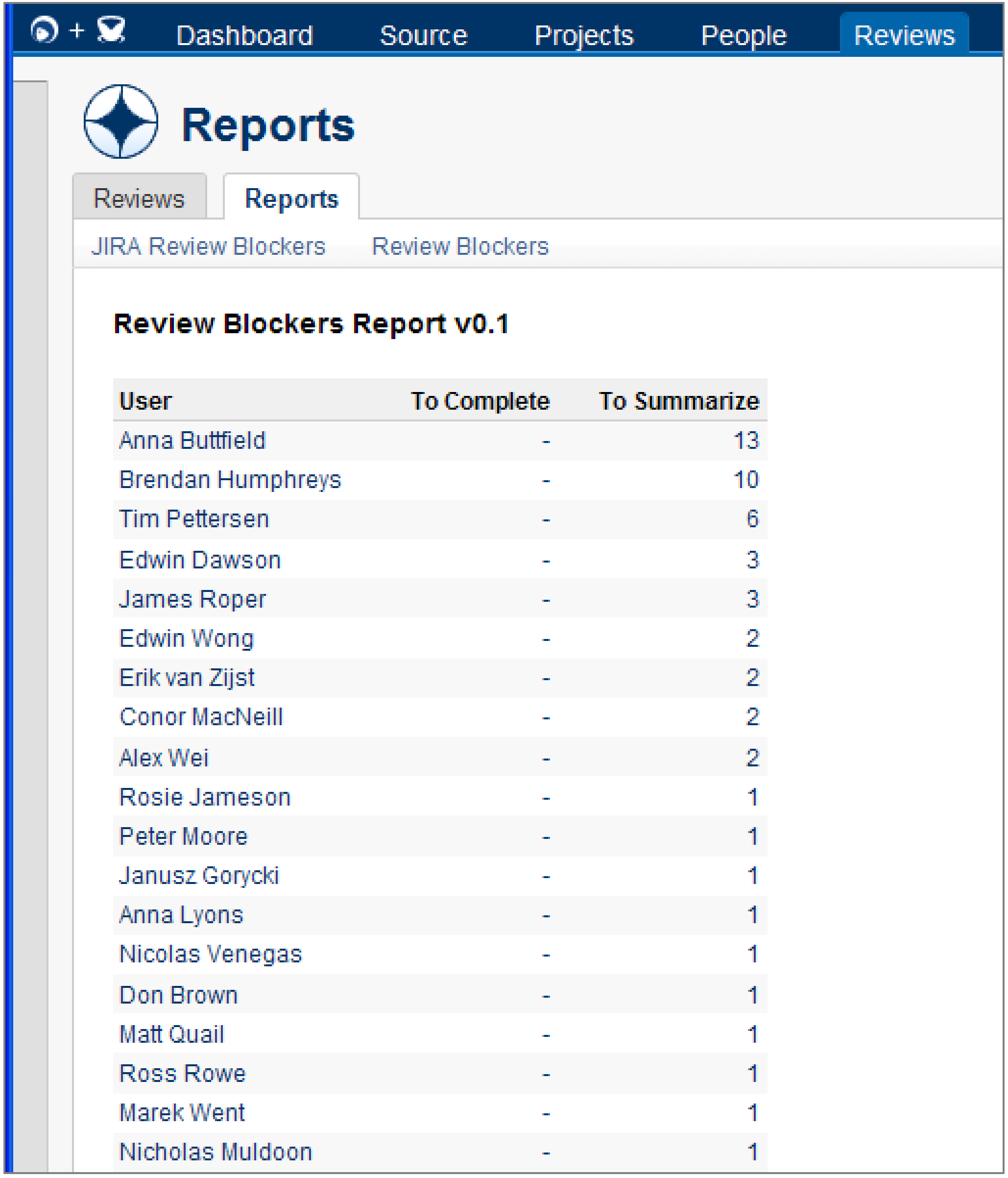
This page contains instructions on how to use the Reports tab in Crucible to see lists of people whose action is required on open reviews. These people are known as 'blockers'.
On this page:

- Viewing the 'Review Blockers' report
- Viewing the 'JIRA Review Blockers' report

Viewing the 'Review Blockers' report

To view a list of people who have open reviews assigned to them, click the '**Reports**' tab at the top of the page. The 'Review Blockers' report will appear.

Screenshot: 'Review Blockers' Report



Review Blockers Report v0.1

User	To Complete	To Summarize
Anna Buttfield	-	13
Brendan Humphreys	-	10
Tim Pettersen	-	6
Edwin Dawson	-	3
James Roper	-	3
Edwin Wong	-	2
Erik van Zijst	-	2
Conor MacNeill	-	2
Alex Wei	-	2
Rosie Jameson	-	1
Peter Moore	-	1
Janusz Gorycki	-	1
Anna Lyons	-	1
Nicolas Venegas	-	1
Don Brown	-	1
Matt Quail	-	1
Ross Rowe	-	1
Marek Went	-	1
Nicholas Muldoon	-	1

You can:

- click a **user's name** to go to their 'Activity' screen.
- click a number in the '**To Complete**' or '**To Summarize**' column to go to the list of reviews waiting to be completed/summarized by the user.

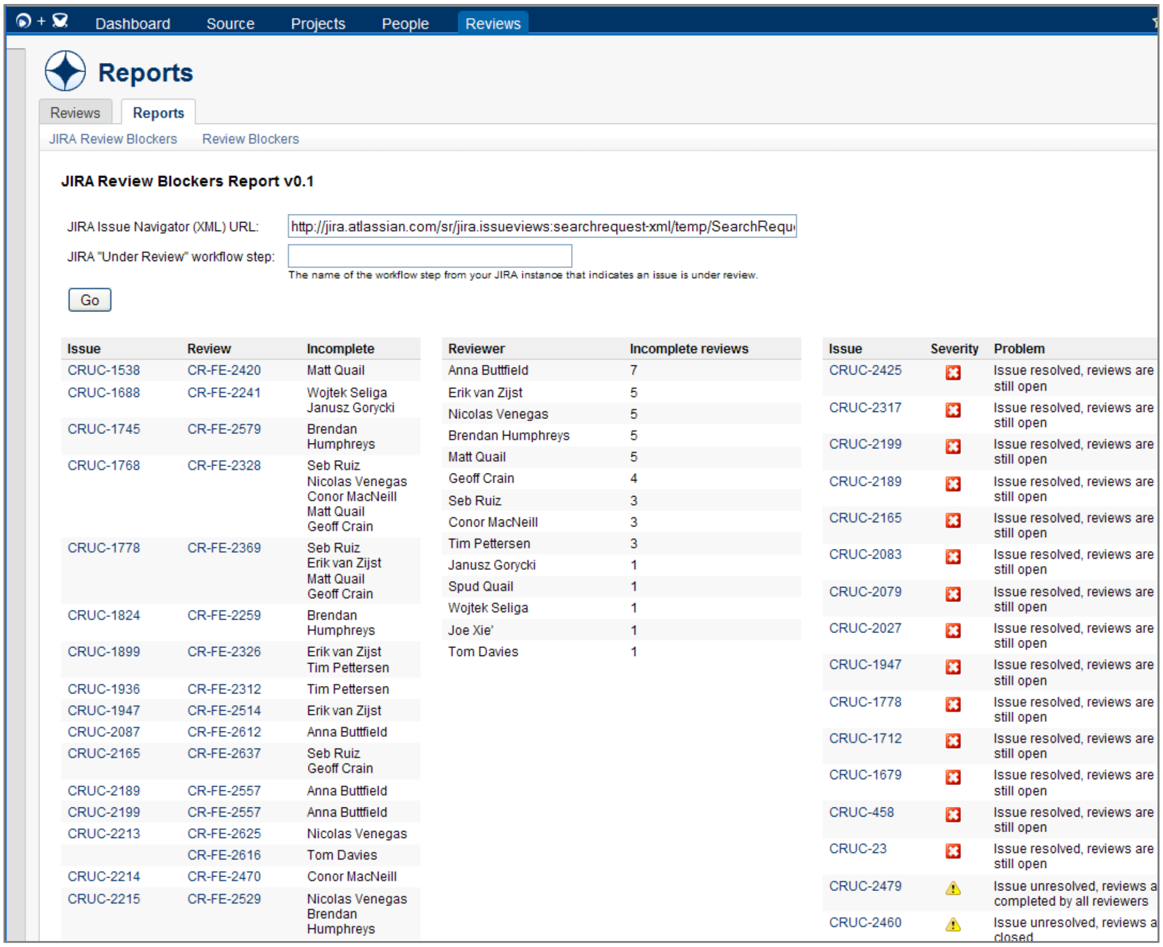
Viewing the 'JIRA Review Blockers' report

The 'JIRA Review Blockers' report shows you a list of users whose action is required on open reviews, for a particular set of JIRA issues.

To view the 'JIRA Review Blockers' report:

1. In JIRA,
    a. Click the '**Issues**' tab to search for the issues in which you are interested.
    ℹ For more information, see the JIRA documentation on Searching for Issues and Advanced Searching.
    b. On the JIRA '**Issue Navigator**' screen, with your search results displayed, click the '**Views**' menu and select '**XML**'.
    ℹ For more information, see the JIRA documentation on Displaying Search Results in XML.
    c. When the XML appears on the screen, copy the URL.
2. In Crucible,
    a. Click the '**Reports**' tab at the top of the page. The 'Review Blockers' report will appear.
    b. Click the '**JIRA Review Blockers**' tab at the top of the report. The 'JIRA Review Blockers' report will appear.
    c. In the '**JIRA Issue Navigator (XML) URL**' field, paste the URL you copied previously.
    d. *(Optional)* If your practice is to set your JIRA issue to a particular Status while its Crucible review is in progress, type the name of the Status in the '**JIRA "Under Review" workflow step**' field.
    e. Click the '**Go**' button. The 'JIRA Review Blockers' report will appear.
    ⚠ If the report doesn't contain the data you expected, you may need to authenticate (i.e. login) to the JIRA server. For more information, please see the JIRA documentation on Accessing protected data.

Screenshot: 'Review Blockers' Report



JIRA Review Blockers Report v0.1

JIRA Issue Navigator (XML) URL: http://jira.atlassian.com/sr/jira.issueviews:searchrequest-xml/temp/SearchRequ

JIRA "Under Review" workflow step:

The name of the workflow step from your JIRA instance that indicates an issue is under review.

Go

Issue	Review	Incomplete
CRUC-1538	CR-FE-2420	Matt Quail
CRUC-1688	CR-FE-2241	Wojtek Seliga Janusz Gorycki
CRUC-1745	CR-FE-2579	Brendan Humphreys
CRUC-1768	CR-FE-2328	Seb Ruiz Nicolas Venegas Conor MacNeill Matt Quail Geoff Crain
CRUC-1778	CR-FE-2369	Seb Ruiz Erik van Zijst Matt Quail Geoff Crain
CRUC-1824	CR-FE-2259	Brendan Humphreys
CRUC-1899	CR-FE-2326	Erik van Zijst Tim Pettersen
CRUC-1936	CR-FE-2312	Tim Pettersen
CRUC-1947	CR-FE-2514	Erik van Zijst
CRUC-2087	CR-FE-2612	Anna Buttfield
CRUC-2165	CR-FE-2637	Seb Ruiz Geoff Crain
CRUC-2189	CR-FE-2557	Anna Buttfield
CRUC-2199	CR-FE-2557	Anna Buttfield
CRUC-2213	CR-FE-2625	Nicolas Venegas
	CR-FE-2616	Tom Davies
CRUC-2214	CR-FE-2470	Conor MacNeill
CRUC-2215	CR-FE-2529	Nicolas Venegas Brendan Humphreys

Reviewer	Incomplete reviews
Anna Buttfield	7
Erik van Zijst	5
Nicolas Venegas	5
Brendan Humphreys	5
Matt Quail	5
Geoff Crain	4
Seb Ruiz	3
Conor MacNeill	3
Tim Pettersen	3
Janusz Gorycki	1
Spud Quail	1
Wojtek Seliga	1
Joe Xie'	1
Tom Davies	1

Issue	Severity	Problem
CRUC-2425	✖	Issue resolved, reviews are still open
CRUC-2317	✖	Issue resolved, reviews are still open
CRUC-2199	✖	Issue resolved, reviews are still open
CRUC-2189	✖	Issue resolved, reviews are still open
CRUC-2165	✖	Issue resolved, reviews are still open
CRUC-2083	✖	Issue resolved, reviews are still open
CRUC-2079	✖	Issue resolved, reviews are still open
CRUC-2027	✖	Issue resolved, reviews are still open
CRUC-1947	✖	Issue resolved, reviews are still open
CRUC-1778	✖	Issue resolved, reviews are still open
CRUC-1712	✖	Issue resolved, reviews are still open
CRUC-1679	✖	Issue resolved, reviews are still open
CRUC-458	✖	Issue resolved, reviews are still open
CRUC-23	✖	Issue resolved, reviews are still open
CRUC-2479	⚠	Issue unresolved, reviews a completed by all reviewers
CRUC-2460	⚠	Issue unresolved, reviews a closed

The columns in the 'JIRA Review Blockers' report show the following, for the given set of JIRA issues:

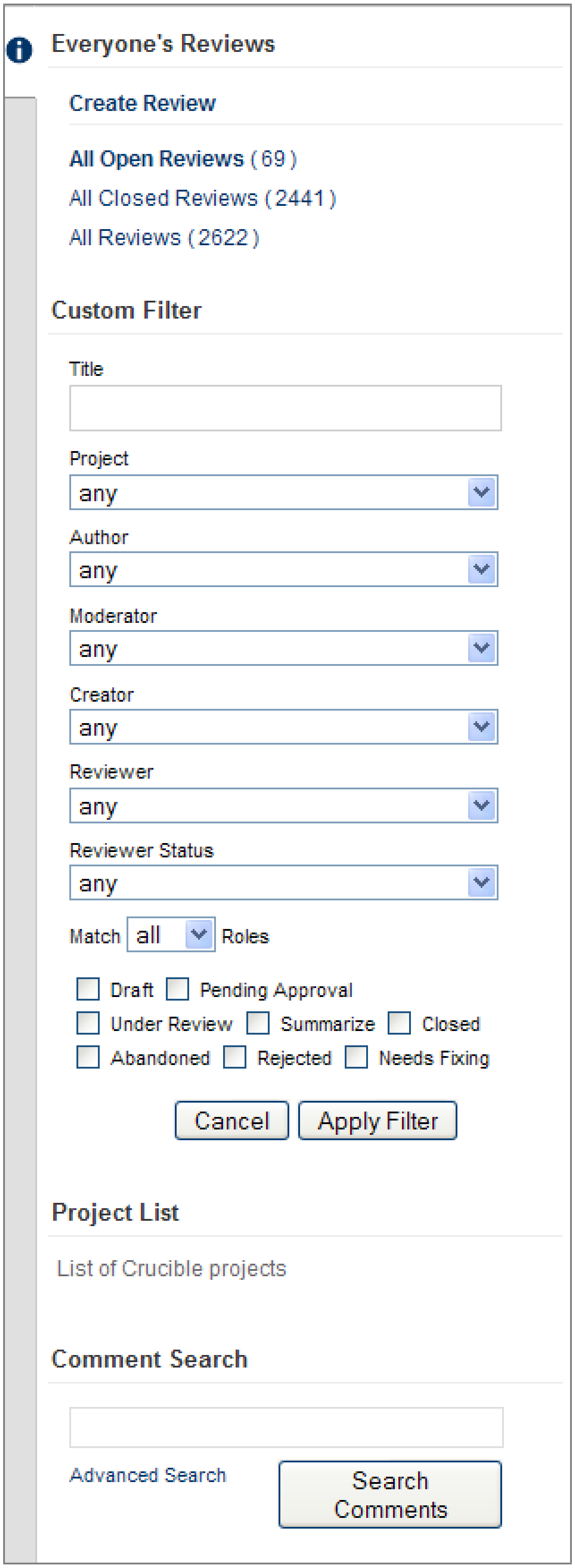
- A list of JIRA issues for which one or more Crucible reviewers has not completed their review.
- A list of users who have an incomplete Crucible review that relates to a JIRA issue.
- A list of open JIRA issues for which a Crucible review is closed, and vice versa.

Searching in Crucible

This page contains instructions on how to use the **Reviews** tab in Crucible to perform searches.

On this page:

Screenshot: Search Filter Options



Everyone's Reviews

Create Review

All Open Reviews (69)
All Closed Reviews (2441)
All Reviews (2622)

Custom Filter

Title

Project: any

Author: any

Moderator: any

Creator: any

Reviewer: any

Reviewer Status: any

Match all Roles

□ Draft □ Pending Approval
□ Under Review □ Summarize □ Closed
□ Abandoned □ Rejected □ Needs Fixing

Project List

List of Crucible projects

Comment Search

Using the Filters Navigation Bar

'Everyone's Reviews' Filters

Any reviews (even if the user has not participated in them) that have been created can be viewed, under the '**Everyone's Reviews**' section. The following options are available:

- All open reviews.
- All closed reviews.
- All reviews.

Custom Filters

To find a specific review, use the '**Custom Filter**':

Title	Find reviews by searching for words within the title.
Project	Find reviews under a particular project.
Author	Find reviews moderated by a particular authors.
Moderator	Find reviews moderated by a particular moderators.
Creator	Find reviews created by a particular creator.
Reviewer	Find reviews that are reviewed by a particular reviewer. This will default to the user logged in.
Reviewer Status	This is reliant on the above filter and is used to show reviews that have either been completed by the reviewer, not completed or all reviews.
Match Roles	To use all the above filters, choose '**all**'. To use any of the filters, choose '**any**'.

Review State Filter

You can use the checkboxes below with the above filters or on their own.

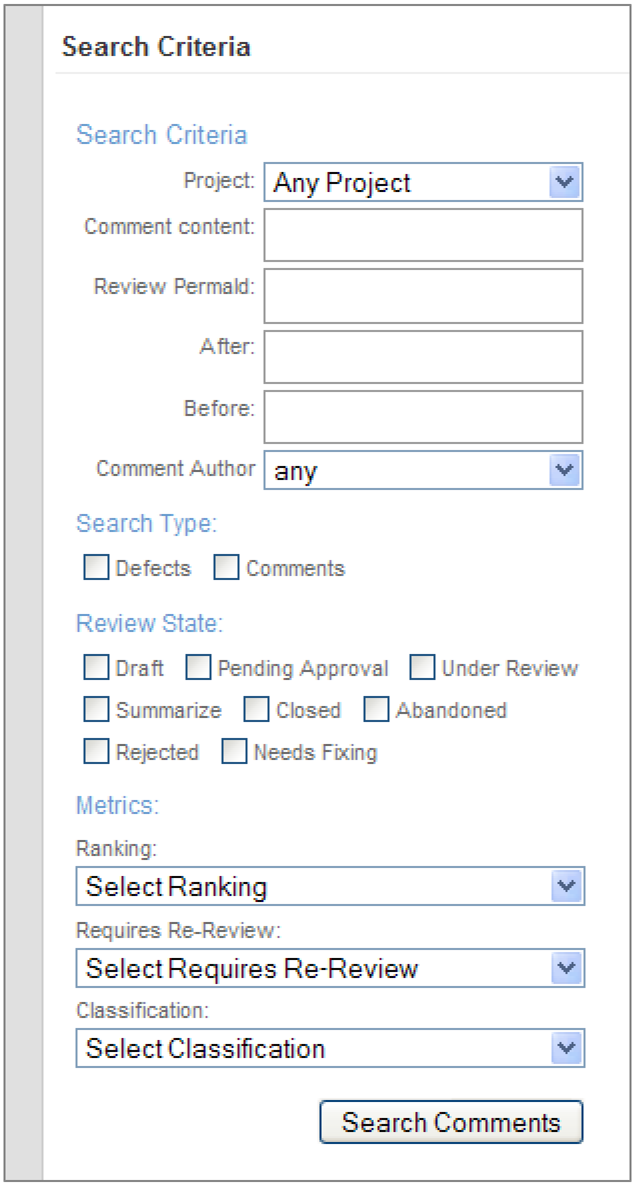
Draft	Reviews that are still in 'Draft' state.
Pending Approval	Reviews that have been moved out of 'Draft' state and are now waiting for the moderator to approve .
Under Review	The review is now 'Under Review'.
Summarize	The review is now in 'Summarize' state.
Closed	Reviews that are now 'Closed'.
Abandoned	Any reviews that are no longer relevant and can be deleted.
Rejected	Any reviews that a moderator has rejected.
Review needs fixing	A review will match this filter if the review enters into an undefined state because something went wrong with storing the review state. A moderator can use this filter to find the review and then change the state to something sensible.

Comment Search

To find review comments that contain particular text, use the '**Comment Search**' box at the bottom of the filters bar. You can also click the '**Advanced Search**' link to display the following comment-filtering options:

Project	Find comments on reviews under a particular project.
Comment content	Find comments that contains the specified text.
Review Permald	Find comments made on the specified review.
After	Find comments made after after a particular date.
Before	Find comments made after before a particular date.
Comment Author	Find comments made by a particular user.
Search Type	You can: - Tick the '**Defects**' check-box to find comments that are flagged as Defects. - Tick the '**Comments**' check-box to find comments that are not flagged as Defects. - Tick neither check-box (or both of them) to find all .
Review State	Find comments on reviews that are in a particular state. See *Review State Filter* (above).
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').
Requires Re-Review	Find defects that have been marked as requiring re-review (or not).
Classification	Find defects that have been given a particular classification (e.g. 'Missing', 'Ambiguous').

Screenshot: Search Comment Filter Options



Search Criteria

Search Criteria

Project: Any Project

Comment content:

Review Permald:

After:

Before:

Comment Author any

Search Type:

□ Defects □ Comments

Review State:

□ Draft □ Pending Approval □ Under Review
□ Summarize □ Closed □ Abandoned
□ Rejected □ Needs Fixing

Metrics:

Ranking: Select Ranking

Requires Re-Review: Select Requires Re-Review

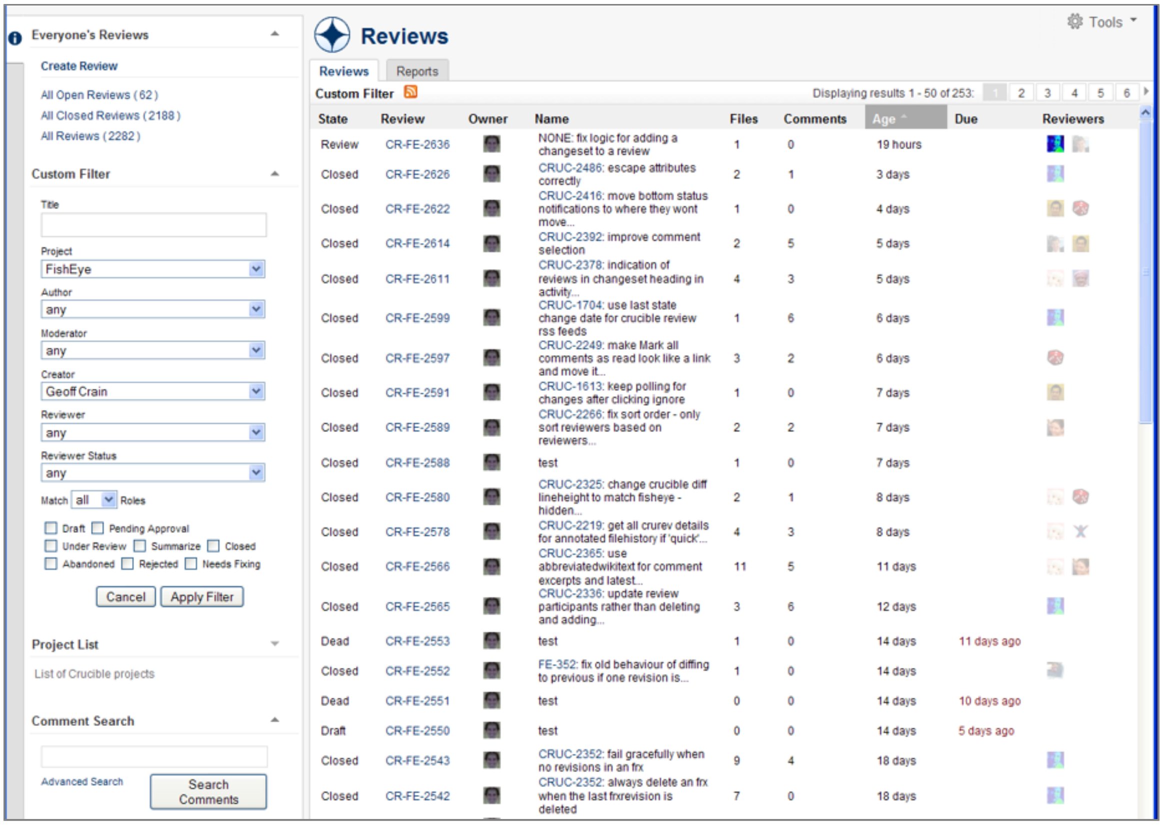
Classification: Select Classification

Viewing Search Results

Review Search

The reviews are displayed in descending order of age. You can click any column heading to re-sort the search results.

Screenshot: Search Results



Title:

Project: FishEye

Author: any

Moderator: any

Creator: Geoff Crain

Reviewer: any

Reviewer Status: any

Match all Roles

□ Draft □ Pending Approval
□ Under Review □ Summarize □ Closed
□ Abandoned □ Rejected □ Needs Fixing

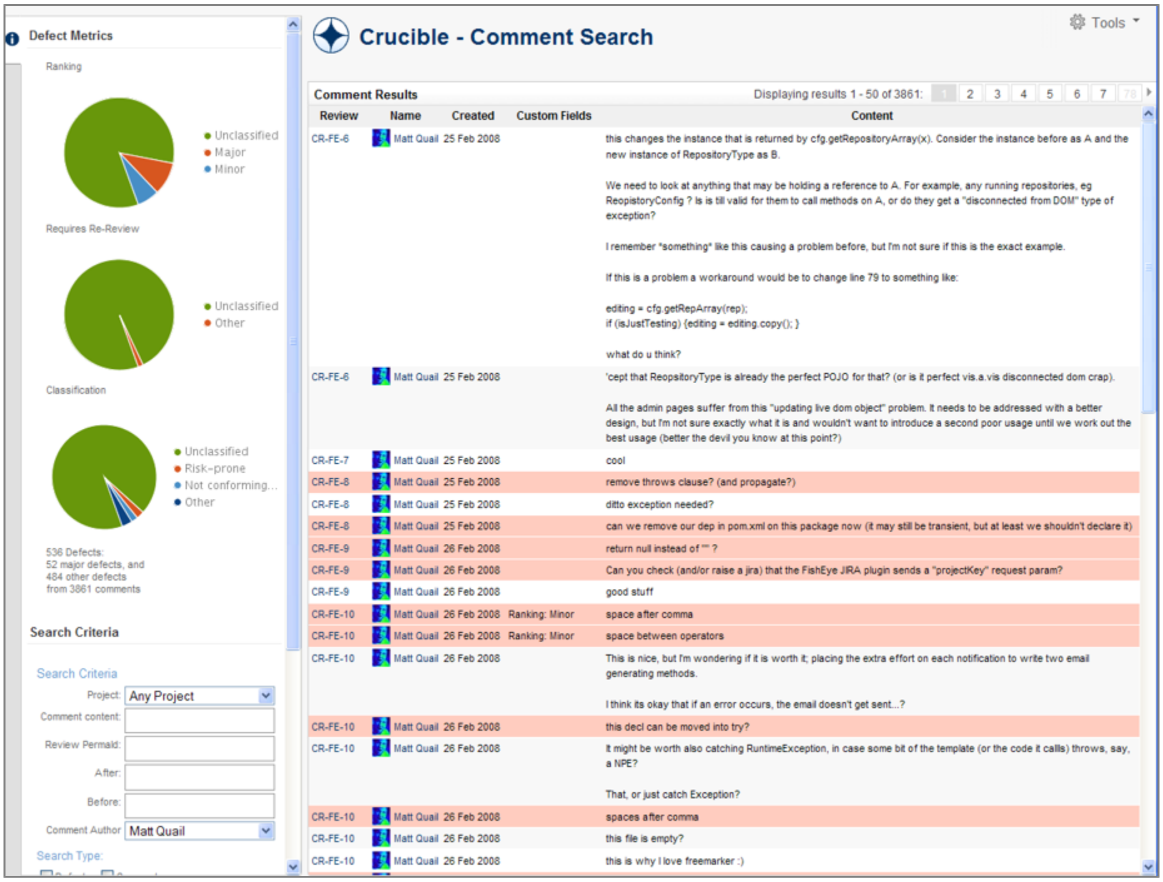
Displaying results 1 - 50 of 253: 1 2 3 4 5 6

State	Review	Owner	Name	Files	Comments	Age	Due	Reviewers
Review	CR-FE-2636		NONE: fix logic for adding a changeset to a review	1	0	19 hours		
Closed	CR-FE-2626		CRUC-2486: escape attributes correctly	2	1	3 days		
Closed	CR-FE-2622		CRUC-2416: move bottom status notifications to where they wont move...	1	0	4 days		
Closed	CR-FE-2614		CRUC-2392: improve comment selection	2	5	5 days		
Closed	CR-FE-2611		CRUC-2378: indication of reviews in changeset heading in activity...	4	3	5 days		
Closed	CR-FE-2599		CRUC-1704: use last state change date for crucible review rss feeds	1	6	6 days		
Closed	CR-FE-2597		CRUC-2249: make Mark all comments as read look like a link and move it...	3	2	6 days		
Closed	CR-FE-2591		CRUC-1613: keep polling for changes after clicking ignore	1	0	7 days		
Closed	CR-FE-2589		CRUC-2266: fix sort order - only sort reviewers based on reviewers...	2	2	7 days		
Closed	CR-FE-2588		test	1	0	7 days		
Closed	CR-FE-2580		CRUC-2325: change crucible diff lineheight to match fisheye - hidden...	2	1	8 days		
Closed	CR-FE-2578		CRUC-2219: get all crurev details for annotated filehistory if 'quick'...	4	3	8 days		
Closed	CR-FE-2566		CRUC-2365: use abbreviatedwikitext for comment excerpts and latest...	11	5	11 days		
Closed	CR-FE-2565		CRUC-2336: update review participants rather than deleting and adding...	3	6	12 days		
Dead	CR-FE-2553		test	1	0	14 days	11 days ago	
Closed	CR-FE-2552		FE-352: fix old behaviour of diffing to previous if one revision is...	1	0	14 days		
Dead	CR-FE-2551		test	0	0	14 days	10 days ago	
Draft	CR-FE-2550		test	0	0	14 days	5 days ago	
Closed	CR-FE-2543		CRUC-2352: fail gracefully when no revisions in an frx	9	4	18 days		
Closed	CR-FE-2542		CRUC-2352: always delete an frx when the last frxrevision is deleted	7	0	18 days		

Comment Search

The comment rearch results include defect classification charts in the left navigation pane.

Screenshot: Comment Search Results



Search Criteria

Project: Any Project

Comment content:

Review Permald:

After:

Before:

Comment Author: Matt Quail

Search Type:

Crucible - Comment Search

Comment Results — Displaying results 1 - 50 of 3861: 1 2 3 4 5 6 7 78

Review	Name	Created	Custom Fields	Content
CR-FE-6	Matt Quail	25 Feb 2008		this changes the instance that is returned by cfg.getRepositoryArray(x). Consider the instance before as A and the new instance of RepositoryType as B. We need to look at anything that may be holding a reference to A. For example, any running repositories, eg ReopistoryConfig ? Is is till valid for them to call methods on A, or do they get a "disconnected from DOM" type of exception? I remember "something" like this causing a problem before, but I'm not sure if this is the exact example. If this is a problem a workaround would be to change line 79 to something like: editing = cfg.getRepArray(rep); if (isJustTesting) {editing = editing.copy(); } what do u think?
CR-FE-6	Matt Quail	25 Feb 2008		'cept that ReopsitoryType is already the perfect POJO for that? (or is it perfect vis.a.vis disconnected dom crap). All the admin pages suffer from this "updating live dom object" problem. It needs to be addressed with a better design, but I'm not sure exactly what it is and wouldn't want to introduce a second poor usage until we work out the best usage (better the devil you know at this point?)
CR-FE-7	Matt Quail	25 Feb 2008		cool
CR-FE-8	Matt Quail	25 Feb 2008		remove throws clause? (and propagate?)
CR-FE-8	Matt Quail	25 Feb 2008		ditto exception needed?
CR-FE-8	Matt Quail	25 Feb 2008		can we remove our dep in pom.xml on this package now (it may still be transient, but at least we shouldn't declare it)
CR-FE-9	Matt Quail	26 Feb 2008		return null instead of "" ?
CR-FE-9	Matt Quail	26 Feb 2008		Can you check (and/or raise a jira) that the FishEye JIRA plugin sends a "projectKey" request param?
CR-FE-9	Matt Quail	26 Feb 2008		good stuff
CR-FE-10	Matt Quail	26 Feb 2008	Ranking: Minor	space after comma
CR-FE-10	Matt Quail	26 Feb 2008	Ranking: Minor	space between operators
CR-FE-10	Matt Quail	26 Feb 2008		This is nice, but I'm wondering if it is worth it; placing the extra effort on each notification to write two email generating methods. I think its okay that if an error occurs, the email doesn't get sent...?
CR-FE-10	Matt Quail	26 Feb 2008		this decl can be moved into try?
CR-FE-10	Matt Quail	26 Feb 2008		It might be worth also catching RuntimeException, in case some bit of the template (or the code it calls) throws, say, a NPE? That, or just catch Exception?
CR-FE-10	Matt Quail	26 Feb 2008		spaces after comma
CR-FE-10	Matt Quail	26 Feb 2008		this file is empty?
CR-FE-10	Matt Quail	26 Feb 2008		this is why I love freemarker :)

Crucible Icons

This page contains a list of Crucible icons and an explanation what each one represents in the user interface.

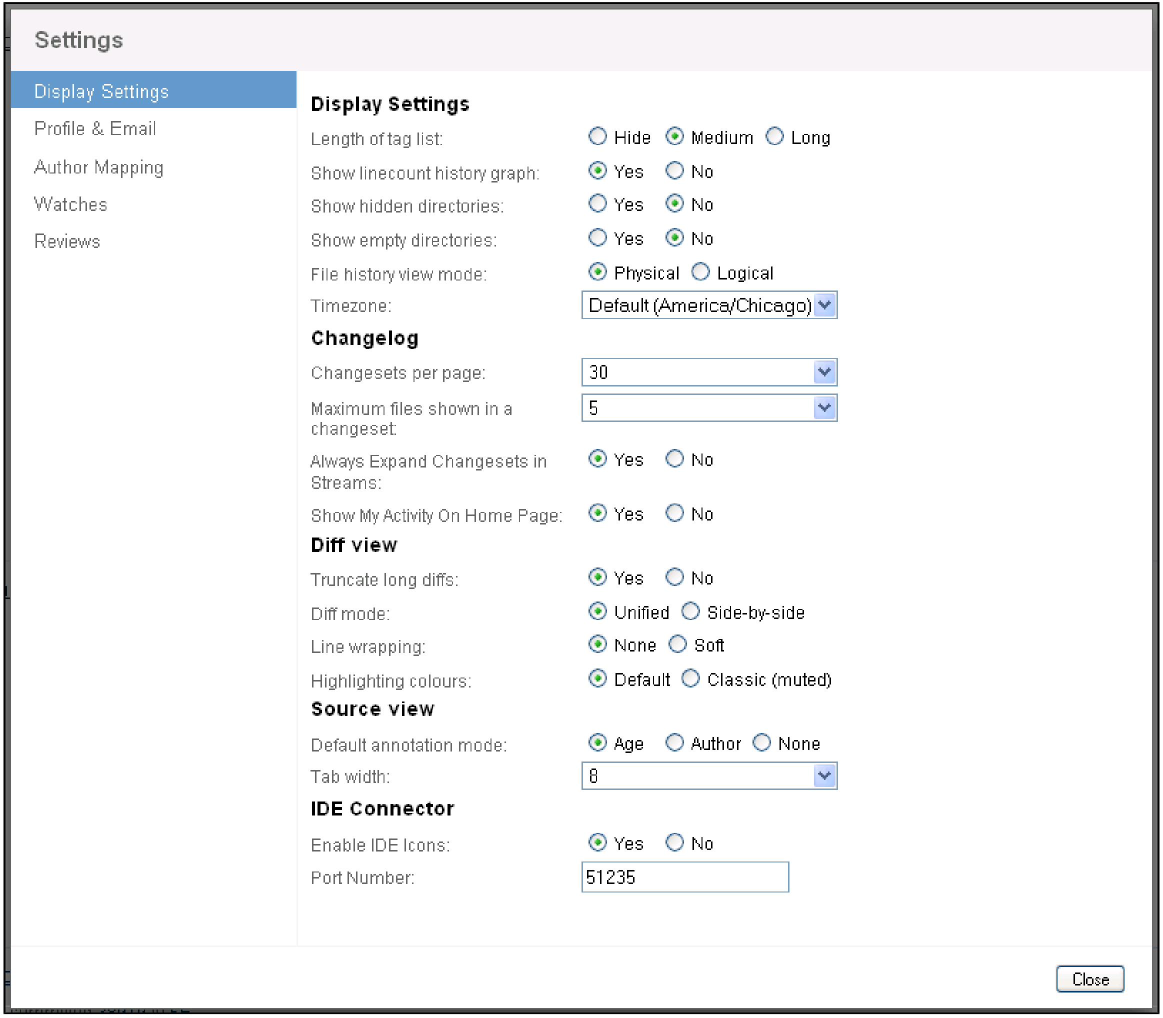
Icon	Description
🗨	View review-level comments
⇦	Go to the previous comment
⇨	Go to the next comment
⊕	Add a comment
←	Go to the previous file in this review
→	Go to the next file in this review
⇕	Expand all files
⧗	Collapse all files
<>	A file included in this review
📁	A directory included in this review

Changing your User Profile

Users can change their own account settings such as passwords, watches and display settings. These include the user avatar and profile image.

To view your user profile, log into Crucible, and click the User Menu (labelled with your username) at the top of the screen, then select '**Settings**'.

Screenshot: Settings



Display Settings

Length of tag list: ○ Hide ● Medium ○ Long

Show linecount history graph: ● Yes ○ No

Show hidden directories: ○ Yes ● No

Show empty directories: ○ Yes ● No

File history view mode: ● Physical ○ Logical

Timezone: Default (America/Chicago)

Changelog

Changesets per page: 30

Maximum files shown in a changeset: 5

Always Expand Changesets in Streams: ● Yes ○ No

Show My Activity On Home Page: ● Yes ○ No

Diff view

Truncate long diffs: ● Yes ○ No

Diff mode: ● Unified ○ Side-by-side

Line wrapping: ● None ○ Soft

Highlighting colours: ● Default ○ Classic (muted)

Source view

Default annotation mode: ● Age ○ Author ○ None

Tab width: 8

IDE Connector

Enable IDE Icons: ● Yes ○ No

Port Number: 51235

Refer to the FishEye documentation for information about the tabs labelled 'Display settings', 'Profile and Email' and 'Watches'.

> **Click 'Save'**
> Always click the 'Save' button after making any changes.

Reviews Tab

If the SMTP server is set up, then you will receive emails when different actions occur within Crucible.

You can change the options described below, to specify the stages at which emails will be sent.

State change	Default is 'Yes'. A Crucible review moves through different states e.g: 'Draft', 'Under Review'. An email is sent when the state changes.

Comment added	Default is 'Yes'. An email is sent when a comment is added to a review.
Participant finished	Default is 'Yes'. An email is sent (to the Moderator only) when any reviewer has completed their review.
General message	Default is 'Yes'. An email is sent when a reviewer is added or removed from a review, after it has gone into the 'Under Review' state.
My actions	Default is 'No'. If set to 'Yes', an email is sent every time you perform an action on a review.

Author Mapping Tab

The 'Author mapping' tab allows you to make an association between you (as a logged-in user in Crucible) and an author in each repository.

This is only necessary if the name of the user within Crucible is different to the name within the repository. Crucible will by default check to see whether the usernames match.

Customising Your User Avatar

If your administrator has enabled an external avatar server (e.g. Gravatar), you can upload an avatar image of your choice. Note that your login name to the external server must be the email address that is specified in your User Profile.

Roles and Status Classifications

This page explains the roles & status classifications in Crucible.

- Roles in Crucible
  - Author
  - Creator/Moderator
  - Reviewer
  - User
- Status Classifications in Crucible
  - Draft
  - Under Review
  - Summarized
  - Closed
  - Abandoned

Roles in Crucible

Author

The *author* is the person primarily responsible for acting on the outcomes of the review. In the vast majority of cases the author will be the person who made the code change under review.

Creator/Moderator

The *creator* is the person who creates the review. In most cases this person will also act as moderator.
The *moderator* is the person responsible for creating the review, approving the review, determining when reviewing is finished, summarising the outcomes and closing the review. By default, the moderator is the creator.

Reviewer

A *reviewer* is a person assigned to review the change. Reviewers can make comments and indicate when they have completed their review. The moderator and author are implicitly considered reviewers.

User

A *user* is a person using Crucible.

Status Classifications in Crucible

Draft

Draft Reviews are not yet completed or released to the reviewers.

Under Review

Reviews Under Review are either waiting for attention by reviewers or waiting to be summarized.

Summarized

Summarized reviews are past the reviewing phase. The moderator can still add conclusions or comments.

Closed

Closed reviews are complete.

Abandoned

Abandoned reviews are 'in the trash'. Reviews must be Abandoned before they can be deleted.

See also the *Glossary of terms* used in Crucible.

Conducting a Review

This page contains links to instructions how to create a review and manage the workflow through its various states to completion. Click on the desired topic to see more information.

- Creating a Review
- Adding Reviewers
- Issuing a Review
- Performing the Review
- Summarising and Closing the Review
- Deleting an Abandoned Review
- Moving a Review to Another Project

For an overview of how to apply a workflow to Crucible, see Defining your Workflow.

For an explanation of the different roles that people play in a review, see Roles and Status Classifications.

Creating a Review

This page explains how to create a Crucible review.

There are a number of ways to create a review. Choose from the list below:

- Creating a Review from FishEye
- Creating a Review from JIRA
- Creating a Review within Crucible

- Creating a Review from a URL
- Creating a Patch Review
- Selecting the Files for the Review

ℹ Note that only people with the '**Create**' permission can create a review.

Creating a Review from FishEye

This page explains how to create a Crucible review from FishEye.

On this page:

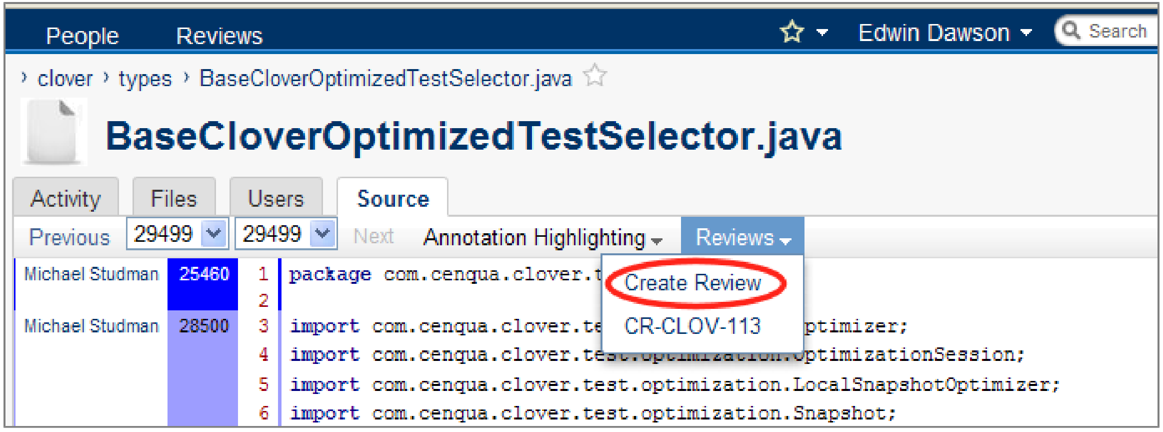
- Opening the FishEye Source View
- Starting the Review
- Choosing a Project
- Selecting Files for Review

Opening the FishEye Source View

To create a review from within FishEye,

To begin, the code author sets up the review. There are a number of ways to do this, but for this example, the author starts from the FishEye Source view of the file he wants to review:

Screenshot: Opening a review from the FishEye Source view

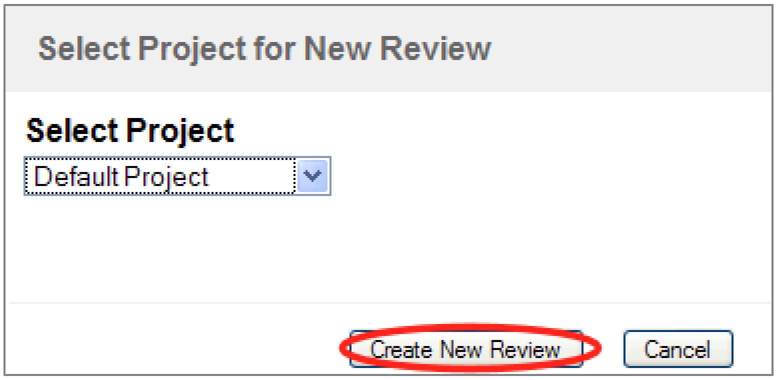


```
package com.cenqua.clover.t

import com.cenqua.clover.te                     ptimizer;
import com.cenqua.clover.test.optimization.OptimizationSession;
import com.cenqua.clover.test.optimization.LocalSnapshotOptimizer;
import com.cenqua.clover.test.optimization.Snapshot;
```

Starting the Review

From the FishEye Source view, the author clicks the '**Reviews**' drop-down menu above the source view, then selects '**Create New Review**'. If there are multiple projects, the Project Selection dialog opens.

Screenshot: The Project Selection dialog

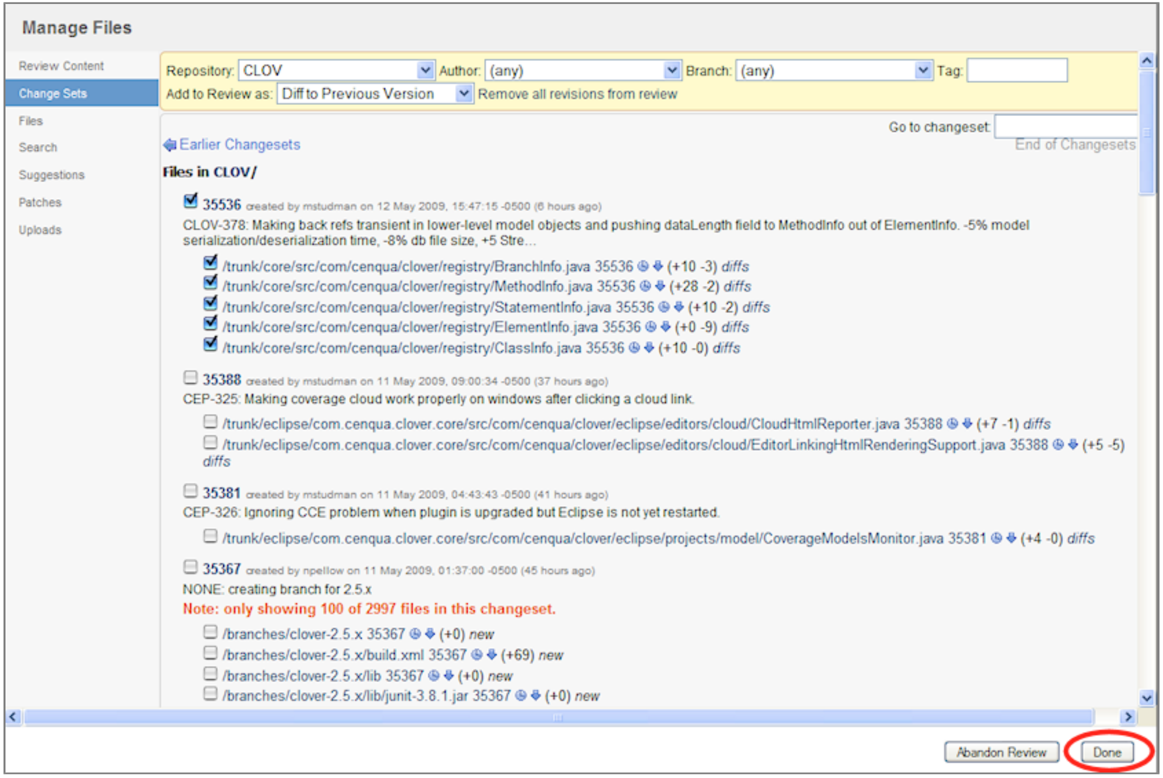


Select Project: Default Project

Choosing a Project

In the Project Selection dialog, you are prompted to choose a project for this review from the drop-down list. Once the selection is made, the author clicks the '**Create New Review**' button. The Manage Files dialog opens.

Screenshot: The Crucible Manage Files dialog



Repository: CLOV Author: (any) Branch: (any) Tag:

Add to Review as: Diff to Previous Version

Go to changeset:

Files in CLOV/

☑ 35536 created by mstudman on 12 May 2009, 15:47:15 -0500 (6 hours ago)

CLOV-378: Making back refs transient in lower-level model objects and pushing dataLength field to MethodInfo out of ElementInfo. -5% model serialization/deserialization time, -8% db file size, +5 Stre...

- ☑ /trunk/core/src/com/cenqua/clover/registry/BranchInfo.java 35536 (+10 -3) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/MethodInfo.java 35536 (+28 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/StatementInfo.java 35536 (+10 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ElementInfo.java 35536 (+0 -9) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ClassInfo.java 35536 (+10 -0) *diffs*

□ 35388 created by mstudman on 11 May 2009, 09:00:34 -0500 (37 hours ago)

CEP-325: Making coverage cloud work properly on windows after clicking a cloud link.

- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/CloudHtmlReporter.java 35388 (+7 -1) *diffs*
- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/EditorLinkingHtmlRenderingSupport.java 35388 (+5 -5) *diffs*

□ 35381 created by mstudman on 11 May 2009, 04:43:43 -0500 (41 hours ago)

CEP-326: Ignoring CCE problem when plugin is upgraded but Eclipse is not yet restarted.

- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/projects/model/CoverageModelsMonitor.java 35381 (+4 -0) *diffs*

□ 35367 created by npellow on 11 May 2009, 01:37:00 -0500 (45 hours ago)

NONE: creating branch for 2.5.x

Note: only showing 100 of 2997 files in this changeset.

- □ /branches/clover-2.5.x 35367 (+0) *new*
- □ /branches/clover-2.5.x/build.xml 35367 (+69) *new*
- □ /branches/clover-2.5.x/lib 35367 (+0) *new*
- □ /branches/clover-2.5.x/lib/junit-3.8.1.jar 35367 (+0) *new*

Selecting Files for Review

In the Manage Files dialog, the author selects the source files they want to include in the review, by clicking the checkboxes next to the desired files. Once finished, the author clicks '**Done**'. The Edit Review dialog appears, where the author can create and issue the review.

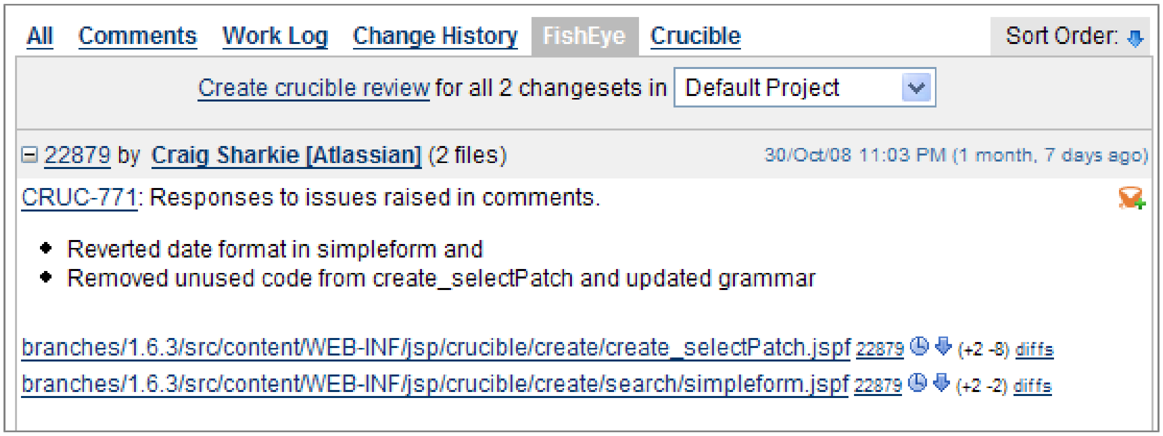
The next step is to add reviewers.

Creating a Review from JIRA

This page explains how to create a Crucible review directly from JIRA, the Atlassian issue-tracker application.

To create a review from within JIRA, click the small Crucible icon next to the required changeset on the '**FishEye**' tab.

Screenshot: Adding a Review from within JIRA



Create crucible review for all 2 changesets in Default Project

22879 by Craig Sharkie [Atlassian] (2 files) 30/Oct/08 11:03 PM (1 month, 7 days ago)

CRUC-771: Responses to issues raised in comments.

- Reverted date format in simpleform and
- Removed unused code from create_selectPatch and updated grammar

branches/1.6.3/src/content/WEB-INF/jsp/crucible/create/create_selectPatch.jspf 22879 (+2 -8) diffs
branches/1.6.3/src/content/WEB-INF/jsp/crucible/create/search/simpleform.jspf 22879 (+2 -2) diffs

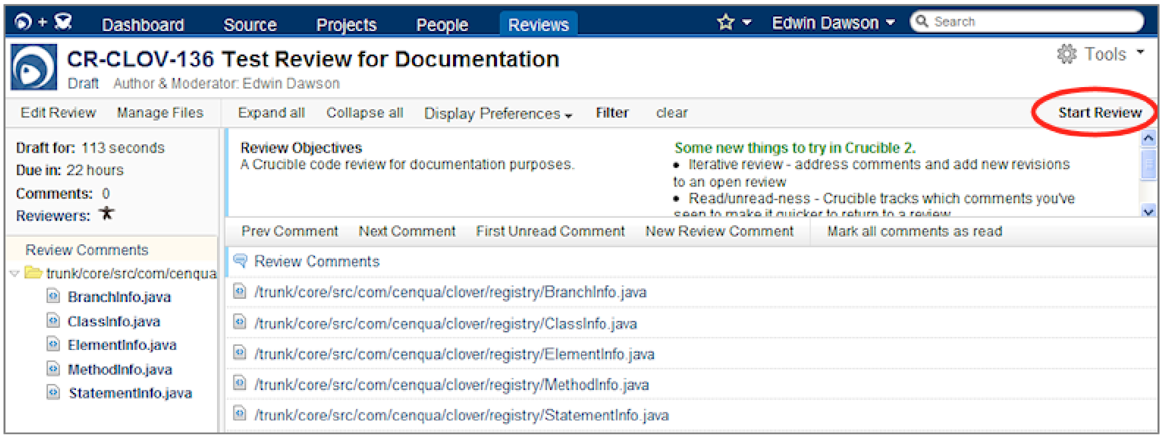
When you click the icon, you will be prompted to select the relevant project(if more than one project exists) in which to create your review. A new draft review will then be created, including the following information pre-filled:

- The content of the changeset becomes the content (i.e. files) to be reviewed.
- The author of the changeset becomes the author of the review, if Crucible is aware of this user. Otherwise the creator of the review becomes the author.
- The creator of the review becomes the moderator.
- The commit log message is used as both the Title and Statement of Objective.

All aspects of the review can be changed. To edit any of the above settings, click the title to see the '**Edit details**' screen. Or you can click the Manage Files tab.

If you click the Crucible icon, you will see the '**Review**' screen below:

Screenshot: Review Screen in Crucible



CR-CLOV-136 Test Review for Documentation

Draft Author & Moderator: Edwin Dawson

Draft for: 113 seconds
Due in: 22 hours
Comments: 0
Reviewers:

Review Objectives
A Crucible code review for documentation purposes.

Some new things to try in Crucible 2.
- Iterative review - address comments and add new revisions to an open review
- Read/unread-ness - Crucible tracks which comments you've seen to make it quicker to return to a review

Review Comments
trunk/core/src/com/cenqua
BranchInfo.java
ClassInfo.java
ElementInfo.java
MethodInfo.java
StatementInfo.java

Review Comments
/trunk/core/src/com/cenqua/clover/registry/BranchInfo.java
/trunk/core/src/com/cenqua/clover/registry/ClassInfo.java
/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java
/trunk/core/src/com/cenqua/clover/registry/MethodInfo.java
/trunk/core/src/com/cenqua/clover/registry/StatementInfo.java

The next step is to add reviewers.

Creating a Review within Crucible

This page explains how to create a review from the Crucible interface.

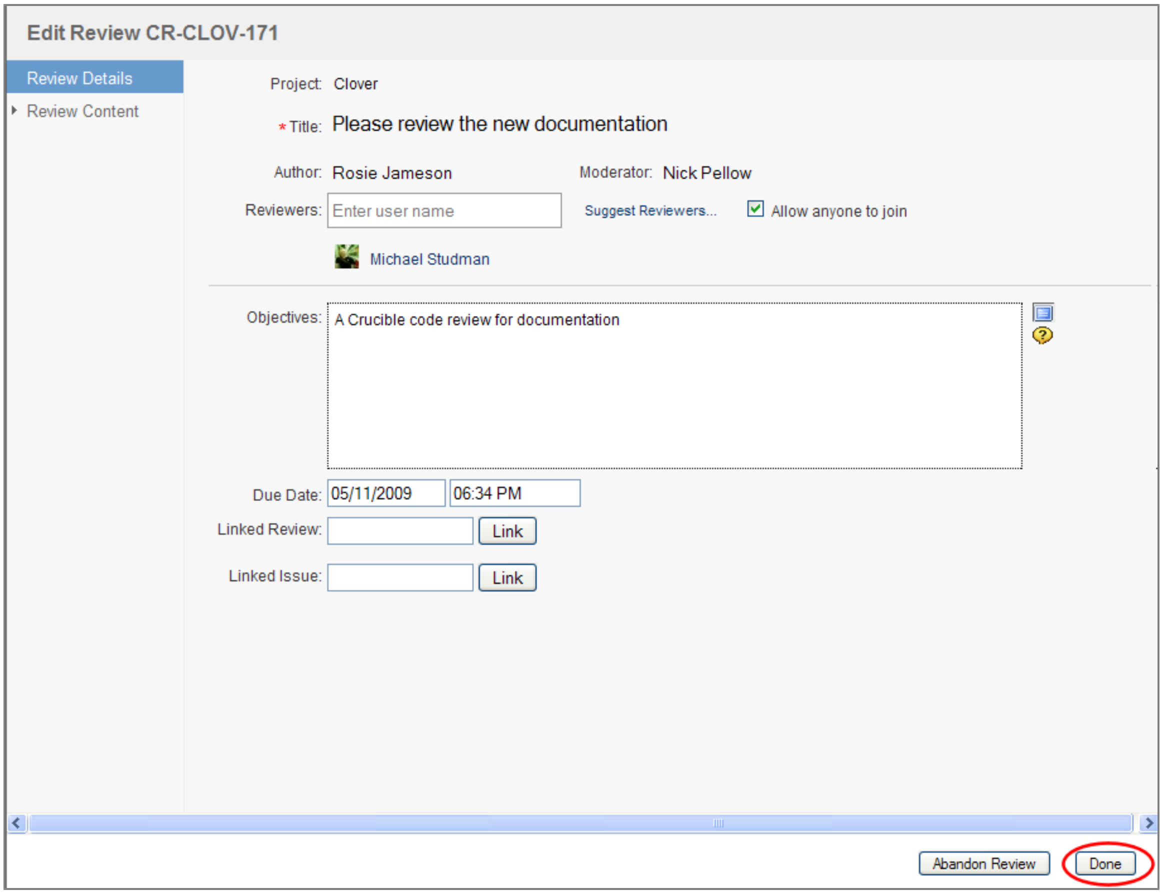
On this page:

- Open the Reviews Screen in Crucible
- Next Steps

Open the Reviews Screen in Crucible

Within Crucible, open the Reviews Screen by clicking the **Reviews** Tab. You can now create a review by opening the **Tools** menu at the top right of the Reviews screen, then selecting **Create Blank Review**. You will be prompted to select the Project to which the review will belong. The **Changesets** screen then appears, allowing you to select the files to be reviewed. Once you have selected the files, click the **Review Details** tab to go to the **Edit details** screen, as shown below.

Screenshot: Adding a review in Crucible



Edit Review CR-CLOV-171

Project: Clover

* Title: Please review the new documentation

Author: Rosie Jameson

Moderator: Nick Pellow

Reviewers: Enter user name

☑ Allow anyone to join

Michael Studman

Objectives: A Crucible code review for documentation

Due Date: 05/11/2009 06:34 PM

Linked Review:

Linked Issue:

Next Steps

- Use the '**Manage Files**' tab to select the files to be reviewed. Read more details about selecting files for review.
- Use the '**Changesets**', '**Files**' or '**Search**' tab to find the files to be added.
- Once you click **Save**, you're ready to Add Reviewers.

Creating a Review from a URL

You can set up a URL which you can then click to create a Crucible review.

The format of your URL is as follows:

```
http:
   //localhost:8060/cru/create?csid=%2F%2F&repo=
```

The parameters are as follows:

Parameter	Description	Required?
csid	The changeset ID. You can specify one or more, of the form //repo/csid (where '%2F' is the URL-encoded form of is '/')	Yes
repo	The name of your repository.	Yes (unless supplied in the csid)
title	The title of your new Crucible review.	No
description	The description of your new Crucible review.	No

When you click the URL, you will be prompted to select the relevant project(if more than one project exists) in which to create your review. A new draft review will then be created, including the following information:

- The content of the changeset becomes the content (i.e. files) to be reviewed.
- The author of the changeset becomes the author of the review, if Crucible is aware of this user. Otherwise the creator of the review becomes the author.
- The creator of the review becomes the moderator.
- The commit log message is used as both the Title (unless you have explicitly defined a title in your URL) and Statement of Objective.

All aspects of the review can be changed. To edit any of the above settings, click the title to see the '**Edit details**' screen. Or you can click the Manage Files tab.

The next step is to add reviewers.

Creating a Patch Review

This page includes instructions on generating patch files from your repository, and how to load them into Crucible to be reviewed.

On this page:

- Creating a Patch in Subversion With All Lines of Code
- Creating a Patch in Perforce With All Lines of Code

Crucible allows you to review a change before it has been committed. To do this, you upload a patch file to the '**Patch**' tab (or paste it in as text) when creating a review. You must first generate this patch file from your repository, using either commands built into your IDE, or via the repository command-line tools.

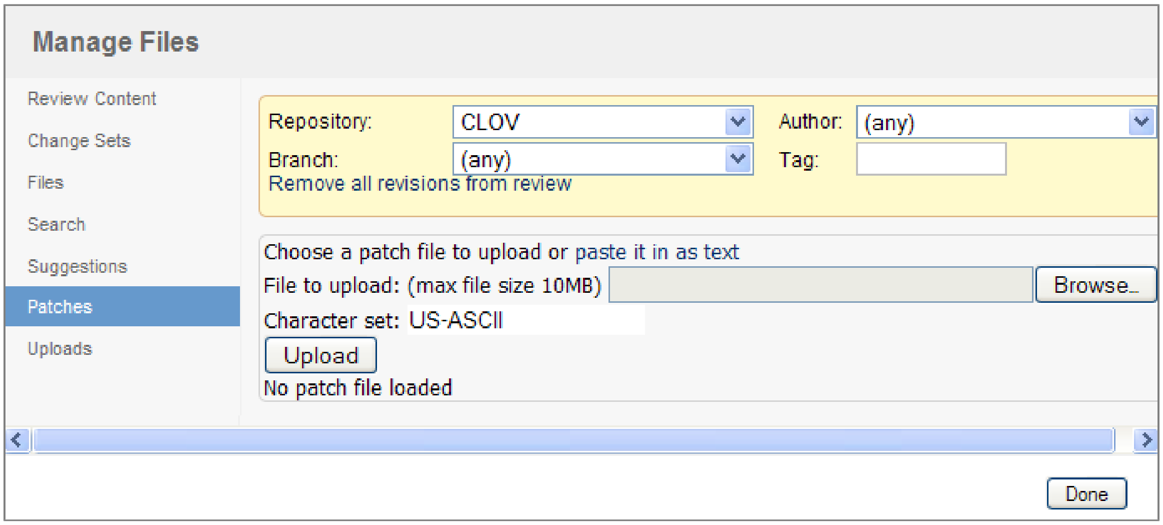
By default, patch files will only show a few lines of code surrounding each change, rather than the entire file and its changes. However, you can create a patch that includes all of the original files' code by using specific parameters (listed on this page) when creating a patch with the command-line tools for your repository.

Crucible can accept a patch file created from any version control system, but instructions on this page cover only Subversion, CVS and Perforce repositories.

Loading a Patch Into Crucible

Once you have generated the patch file, loading patch files into Crucible is easy.

Screenshot: The Crucible Patch tab



Repository: CLOV
Author: (any)
Branch: (any)
Tag:
Remove all revisions from review

Choose a patch file to upload or paste it in as text
File to upload: (max file size 10MB)
Character set: US-ASCII
No patch file loaded

If you already have the patch file you want to review, open the Patches View by clicking '**Manage Files**' on the review toolbar, then click '**Patches**' on the left navigation bar of the Manage Files view. From there, simply click '**Browse**' to select the file and click '**Upload**' to add it to your review.

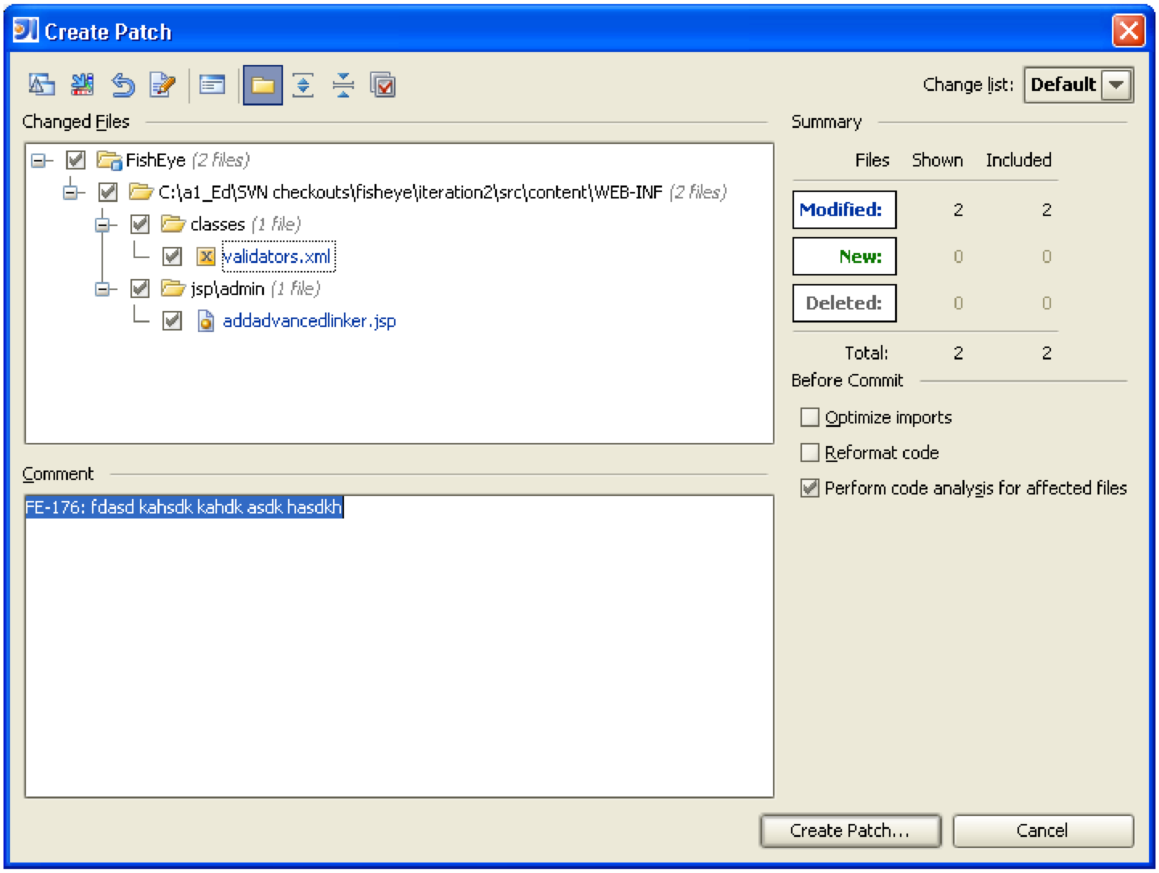
Creating a Patch File From Your IDE

Creating a Patch File in IntelliJ IDEA 7.0

To create a Patch File under IntelliJ IDEA, do the following:

Select a parent folder, sub-folder or file that you have altered in the Project tool window. Select '**Version Control**' > '**Create Patch**'. The following window appears:

Screenshot: The IDEA Create Patch window


Create Patch

Change list: Default

Changed Files

- ☑ FishEye *(2 files)*
  - ☑ C:\a1_Ed\SVN checkouts\fisheye\iteration2\src\content\WEB-INF *(2 files)*
    - ☑ classes *(1 file)*
      - ☑ validators.xml
    - ☑ jsp\admin *(1 file)*
      - ☑ addadvancedlinker.jsp

Summary

	Files Shown	Included
Modified:	2	2
New:	0	0
Deleted:	0	0
Total:	2	2

Before Commit

☐ Optimize imports
☐ Reformat code
☑ Perform code analysis for affected files

Comment

FE-176: fdasd kahsdk kahdk asdk hasdkh

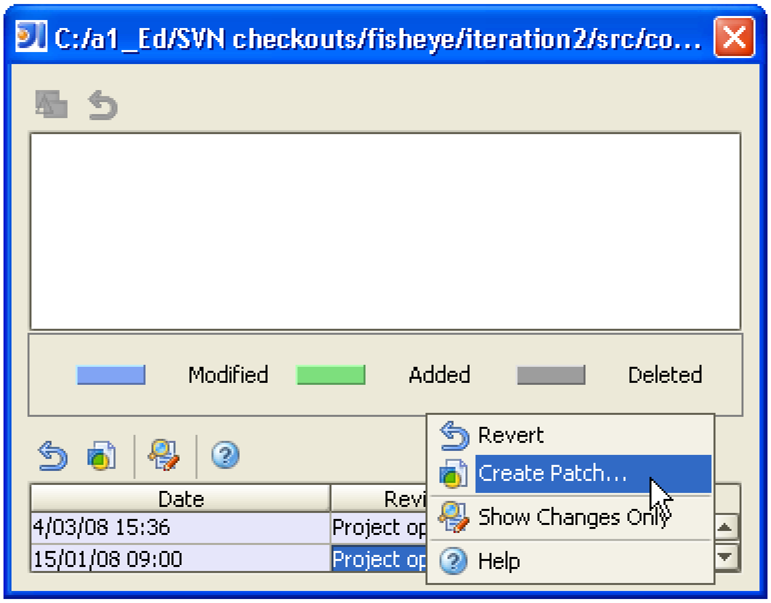
Click '**Create Patch**'. Choose a location to save the patch file and click '**Ok**'.

If You Do Not Have the Create Patch Command Available Under IDEA

If you have not configured version control in IDEA, you may not have the '**Create Patch**' option available. If so, use the following steps to create a patch file in IDEA:

1. Select a parent folder, sub-folder or file that you have altered in the Project tool window, right-click it and choose '**Local History**' > '**Show History**'.

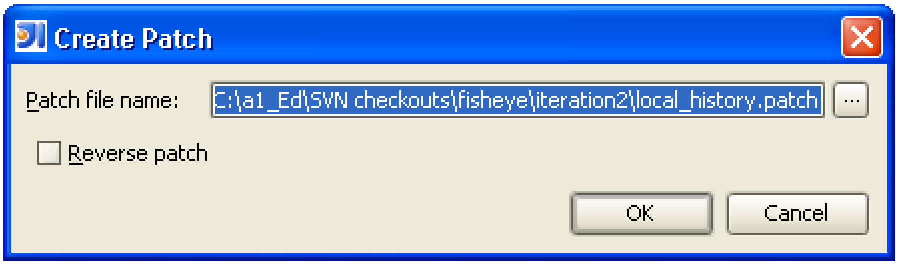
Screenshot: The IDEA Show History dialog


C:/a1_Ed/SVN checkouts/fisheye/iteration2/src/co...

Date	Revi
4/03/08 15:36	Project op
15/01/08 09:00	Project op

2. In the Local History view, right-click the revision number, and choose '**Create Patch**'.

Screenshot: The IDEA Create Patch dialog



Create Patch

Patch file name: C:\a1_Ed\SVN checkouts\fisheye\iteration2\local_history.patch

☐ Reverse patch

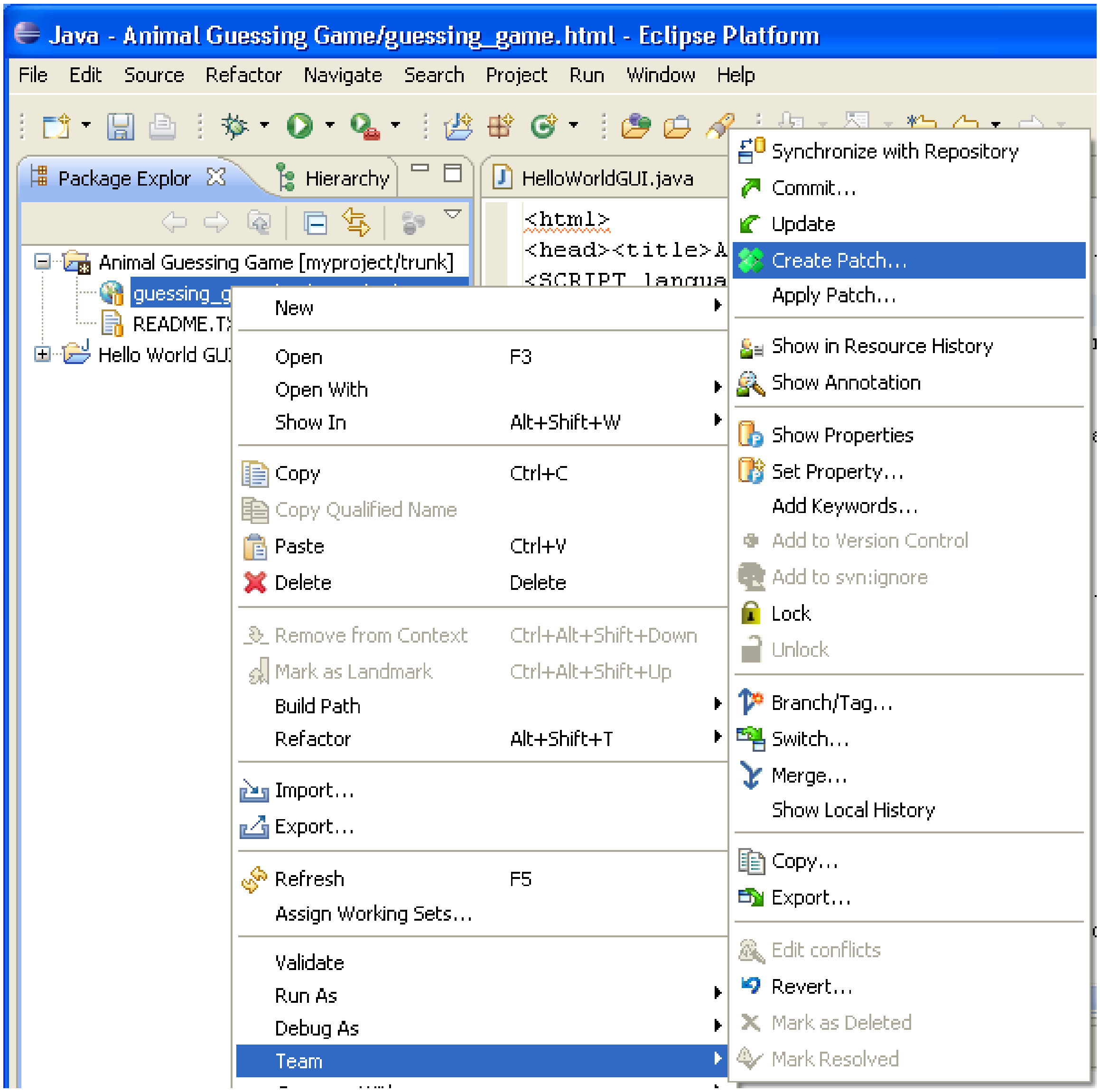
3. In the Create Patch dialog, choose a location for the patch file and a file name, then click '**OK**'.

Creating a Patch File in Eclipse 3.3.1.1

To create a patch file under Eclipse, do the following:

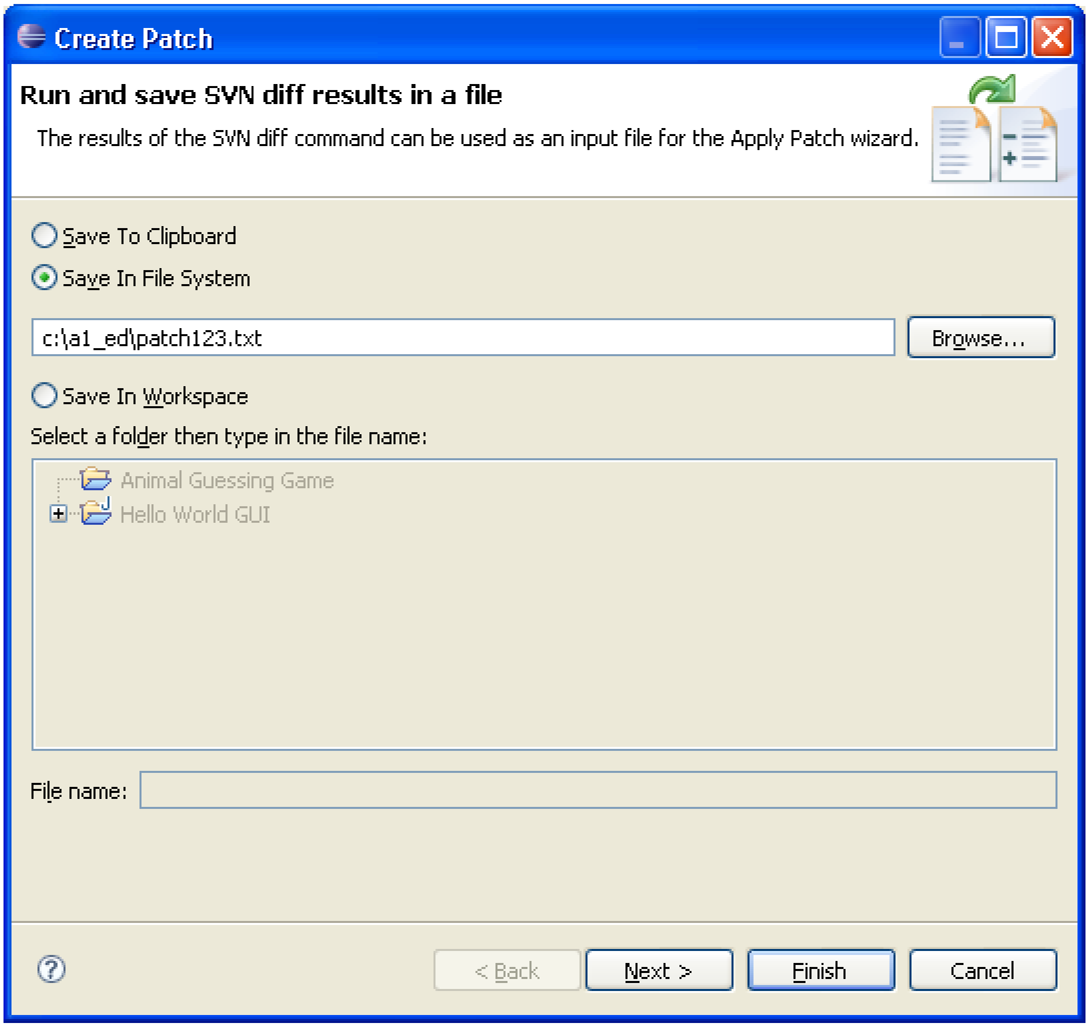
Find the parent folder, sub-folder or file that you have altered, right-click it and choose '**Team**' > '**Create Patch**'.

Screenshot: Instigating a Patch in Eclipse



In the Create Patch window, choose a location on your computer and type an appropriate file name (the file format is plain text).

Screenshot: The Eclipse Create Patch dialog



Create Patch

Run and save SVN diff results in a file

The results of the SVN diff command can be used as an input file for the Apply Patch wizard.

○ Save To Clipboard

● Save In File System

c:\a1_ed\patch123.txt

○ Save In Workspace

Select a folder then type in the file name:

- Animal Guessing Game
- Hello World GUI

File name:

Creating a Basic Patch File From The Repository Command Line

CVS Patch Creation On The Command Line

To create a patch in CVS, use the `cvs diff -Nu` command from your workspace. For example:

```
cvs diff -Nu > patch.txt
```

Note that patch files created with this command will only include around three lines of code, before and after each change.

Subversion Patch Creation Via The Command Line

To create a patch in Subversion, use the `svn diff` command from your workspace. For example:

```
svn diff > patch.txt
```

⚠ `svn diff` does not print any information about files copied in the workspace.

Note that patch files created with this command will only include around three lines of code, before and after each change.

Perforce Patch Creation Via The Command Line

> ⚠ **Set P4DIFF**
> To create a patch in Perforce, you must ensure you have set P4DIFF to point to a GNU-compatible diff program.

Then use `p4 diff -du` to generate a patch for changed files. For example:

```
p4 diff -du > patch.txt
```

Since Perforce diffs do not include added and deleted files so you should then do a `p4 opened` to find such files. For added files, append them individually to the patch using GNU diff:

```
diff -u path_to_added_file /dev/null >>
   patch.txt
```

In the example above, replace `path_to_added_file` with the actual path of your added file. You can follow a similar procedure with deleted files using `p4 print` to extract the previous version of the file.

Note that patch files created with this command will only include around three lines of code, before and after each change.

Creating Patches That Include All Lines of Code

To create a patch file that shows all lines of code as well as the changes, use the following commands from your repository command-line tools.

Creating a Patch in CVS With All Lines of code

To create a CVS patch that shows all code (not just the changes and surrounding code), use this command:

```
cvs diff -N -U 10000 > patch.txt
```

ℹ The '10000' number refers to the number of code lines included in the patch, before and after each change. '**Patch.txt**' represents your desired name for the new patch file.

Creating a Patch in Subversion With All Lines of Code

To create a patch in Subversion that shows all code (not just the changes and surrounding code), use this command:

```
svn diff --diff-cmd diff -x "-U 10000" >
   patch.txt
```

- The in-built diff feature in `svn diff` does not support specifying lines of context, so you must tell Subversion to use an external diff command.
- The second "diff" in the command above needs to be the name of your external diff command. You might need to specify the full path to that command, such as `/usr/bin/diff`.
- On the Windows platform, you may need a Unix-like emulator such as Cygwin, and install the optional diff command for that.

Creating a Patch in Perforce With All Lines of Code

To create a Perforce patch that shows all code (not just the changes and surrounding code), use this command:

```
p4 diff -dU10000 > patch.txt
```

ℹ The '10000' number refers to the number of code lines included in the patch, before and after each change. '**Patch.txt**' represents your desired name for the new patch file.

Selecting the Files for the Review

This page explains how to select files that will be included in a Crucible review.

The **Manage Files** dialog allows you to select and modify which files make up the review. In the Manage Files dialog, the author selects the source files they want to include in the review, by clicking the checkboxes next to the desired files. Once finished, the author clicks '**Done**'.

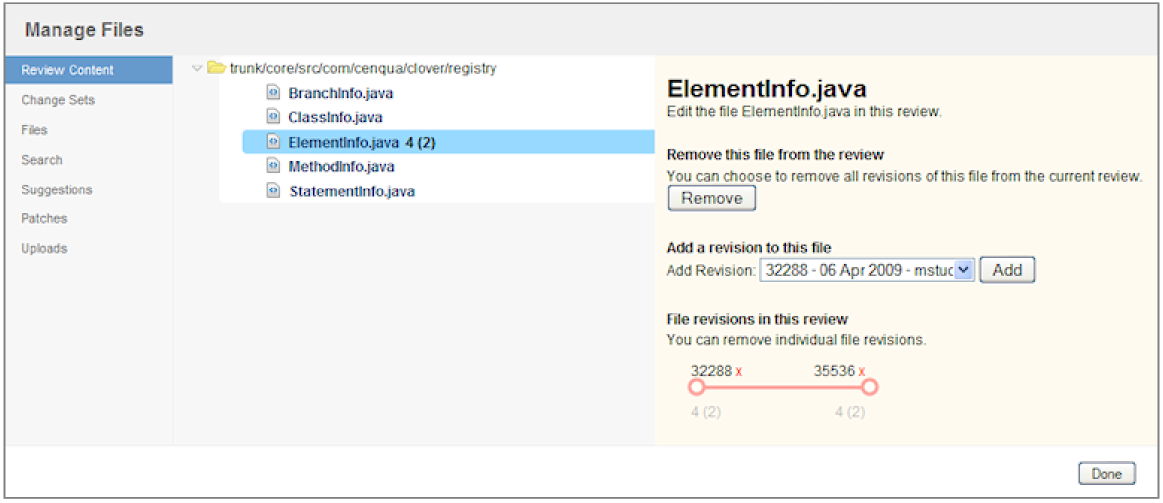
On this page:

Overview of the Manage Files Dialog

The '**Manage Files**' dialog is first displayed when you are creating a review. Once a review has started, (as the creator or moderator) you can reopen the dialog by clicking the **Manage Files** button on the Review toolbar.

The left navigation bar allows you to select and locate different files for the review including files from repositories, patch files and uploaded files. The screenshot below shows the view shown when checking the list of currently selected files; the **'Review Content'** view.

Screenshot: The Review Content View Under the Manage Files Dialog



ElementInfo.java

Edit the file ElementInfo.java in this review.

Remove this file from the review

You can choose to remove all revisions of this file from the current review.

Add a revision to this file

Add Revision: 32288 - 06 Apr 2009 - mstuc

File revisions in this review

You can remove individual file revisions.

See the table below for detailed instructions on the various links on the left navigation bar of the Manage Files dialog.

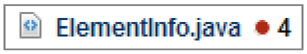
Left Navigation Bar Links In The Manage Files Dialog

Left Nav Title	Explanation	Special Conditions
Review Content	Shows the list of files selected for review and allows you to manage them.	
Change Sets	Allows you to choose change sets from a Source Code Management (SCM) repository and add them to a review.	Only appears when FishEye is installed.
Files	Allows you to choose files from a Source Code Management (SCM) repository and add them to a review.	
Search	Allows you to search a Source Code Management (SCM) repository for files or change sets and add them to a review.	Only appears when FishEye is installed.

Suggestions	Analyses the list of files in the current review and makes suggestions based on certain logic (for example, suggesting a newer version of a file if one exists).	
Patches	Allows you to upload patch files to a review.	
Uploads	Allows you to upload any file to a review, including binary files and files outside of a Source Code Management (SCM) repository.	

When a red dot is shown next to a file name or changeset name, it indicates that for the file you have selected, a newer version exists. When a number is shown next to the file or changeset name, it indicates the number of comments that have been made. Numbers in brackets show unread comments.

Screenshot: Red Dot and Comment Counter in the Manage Files Dialog

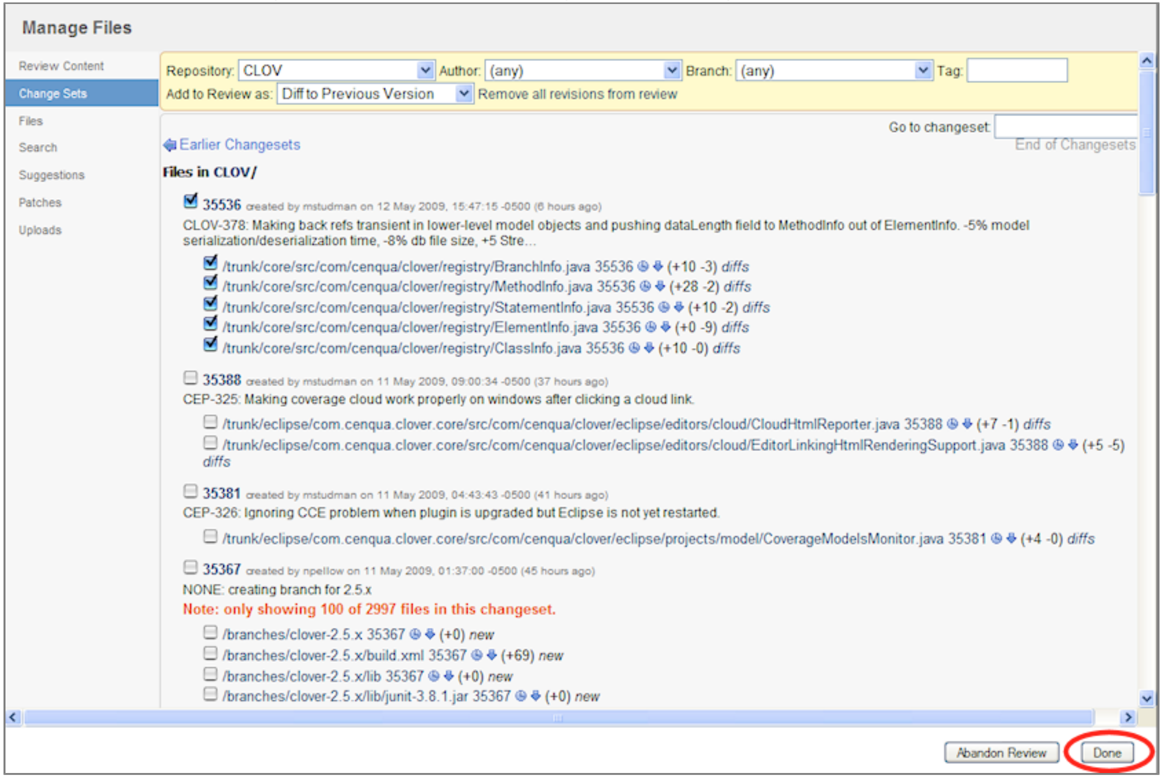


ElementInfo.java ● 4

Selecting Changesets for Review

To add changesets to a review, open the **'Changesets'** view by clicking the "Changesets" option on the left navigation bar in the Manage Files dialog. The Changesets view opens.

Screenshot: The Changesets View in the Manage Files Dialog



Repository: CLOV Author: (any) Branch: (any) Tag:

Add to Review as: Diff to Previous Version Remove all revisions from review

Go to changeset:

Files in CLOV/

☑ 35536 created by mstudman on 12 May 2009, 15:47:15 -0500 (6 hours ago)

CLOV-378: Making back refs transient in lower-level model objects and pushing dataLength field to MethodInfo out of ElementInfo. -5% model serialization/deserialization time, -8% db file size, +5 Stre...

- ☑ /trunk/core/src/com/cenqua/clover/registry/BranchInfo.java 35536 (+10 -3) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/MethodInfo.java 35536 (+28 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/StatementInfo.java 35536 (+10 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ElementInfo.java 35536 (+0 -9) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ClassInfo.java 35536 (+10 -0) *diffs*

□ 35388 created by mstudman on 11 May 2009, 09:00:34 -0500 (37 hours ago)

CEP-325: Making coverage cloud work properly on windows after clicking a cloud link.

- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/CloudHtmlReporter.java 35388 (+7 -1) *diffs*
- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/editors/cloud/EditorLinkingHtmlRenderingSupport.java 35388 (+5 -5) *diffs*

□ 35381 created by mstudman on 11 May 2009, 04:43:43 -0500 (41 hours ago)

CEP-326: Ignoring CCE problem when plugin is upgraded but Eclipse is not yet restarted.

- □ /trunk/eclipse/com.cenqua.clover.core/src/com/cenqua/clover/eclipse/projects/model/CoverageModelsMonitor.java 35381 (+4 -0) *diffs*

□ 35367 created by npellow on 11 May 2009, 01:37:00 -0500 (45 hours ago)

NONE: creating branch for 2.5.x

Note: only showing 100 of 2997 files in this changeset.

- □ /branches/clover-2.5.x 35367 (+0) *new*
- □ /branches/clover-2.5.x/build.xml 35367 (+69) *new*
- □ /branches/clover-2.5.x/lib 35367 (+0) *new*
- □ /branches/clover-2.5.x/lib/junit-3.8.1.jar 35367 (+0) *new*

By default, Crucible presents a list of the author's changesets in reverse chronological order. You can see other changesets by changing the options at the top of the screen.

Click the checkbox next to a changeset ID to add the entire changeset, or the checkboxes next to file names to add or remove individual files. Clicking the checkbox of a parent item will select or de-select all of its children. Click '**Remove all revisions from review**' to remove all.

Options for Adding Changesets

Repository	This is a list of the repositories that contain the files that can be reviewed. If the repository you require is not in the list then it has not been added to FishEye. Please contact your Crucible/FishEye administrator.
Author	This contains a list of all the authors who have made changes within the repository. When creating a review, this will default if possible to the username of the user authoring this review and will therefore show their changesets.
Branch	This will only show files and recent changes on that branch from the repository set above.
Tag	This will only show files and recent changes tagged.
Add to Review As	Choose the form of the review; see full details.
Go to Changeset	Allows you to jump to a particular change set by entering its title and pressing Enter.

Understanding the Icons Shown with Changesets

To help decide what files are to be placed under review, you can click the icons next to the files to gain further information about them before they go out for review:

Clock icon or the file URL	A view detailing the history of this particular file.
Changeset ID next to the file URL	A view of the complete file. Amended lines are highlighted on the left in yellow.
Down arrow icon	Option to download the file.
Change Indicator (+n -n)	This shows how many lines have been amended (e.g. +3 -2) and also what type of change has been made. If it says '*diffs*' then you can click this to see the differences in the file between the revisions.

After you have chosen the files, click the '**Review Content**' option on the left navigation bar to view the files that are included in the review.

Selecting Files for Review

To add files to a review, click the **'Files'** option on the left navigation bar of the Manage Files dialog. The Files view opens.

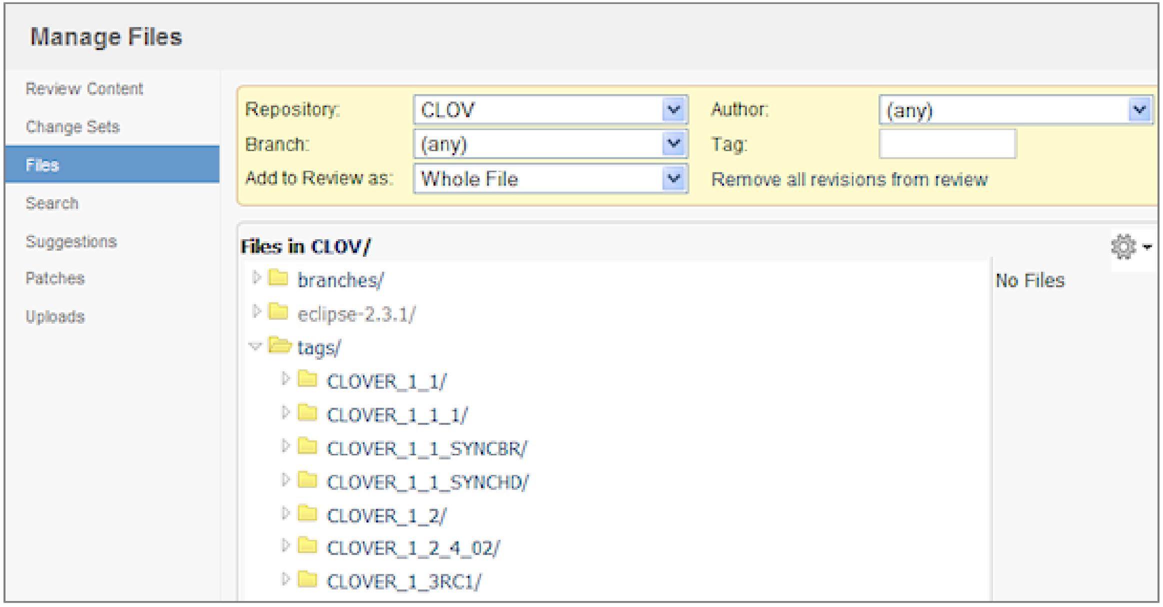
To find a file, browse the folders by clicking the relevant folder. The folders by default are sorted by path name but can sorted by last-commit or first-commit. To choose a file for reviewing, click the checkbox to the left of the filename and if required the revision number. To select a particular revision of a file, open the revision number list and select the option "Load Full History...". This will refresh the available options in the list.

Caveats:

- Empty folders will be greyed out.
- If the folders contain empty folders then a toggle option called '**Hide Empty**' will appear under the 'Sort' options.
- To see or ignore **deleted** files, you can click the '**Hide**' and '**Show**' options located above the file names on the left.

After you have chosen the files, click the '**Review Content**' view to see the files that are included in the review.

Screenshot: Files View in the Manage Files Dialog



Manage Files

Repository: CLOV
Branch: (any)
Add to Review as: Whole File
Author: (any)
Tag:

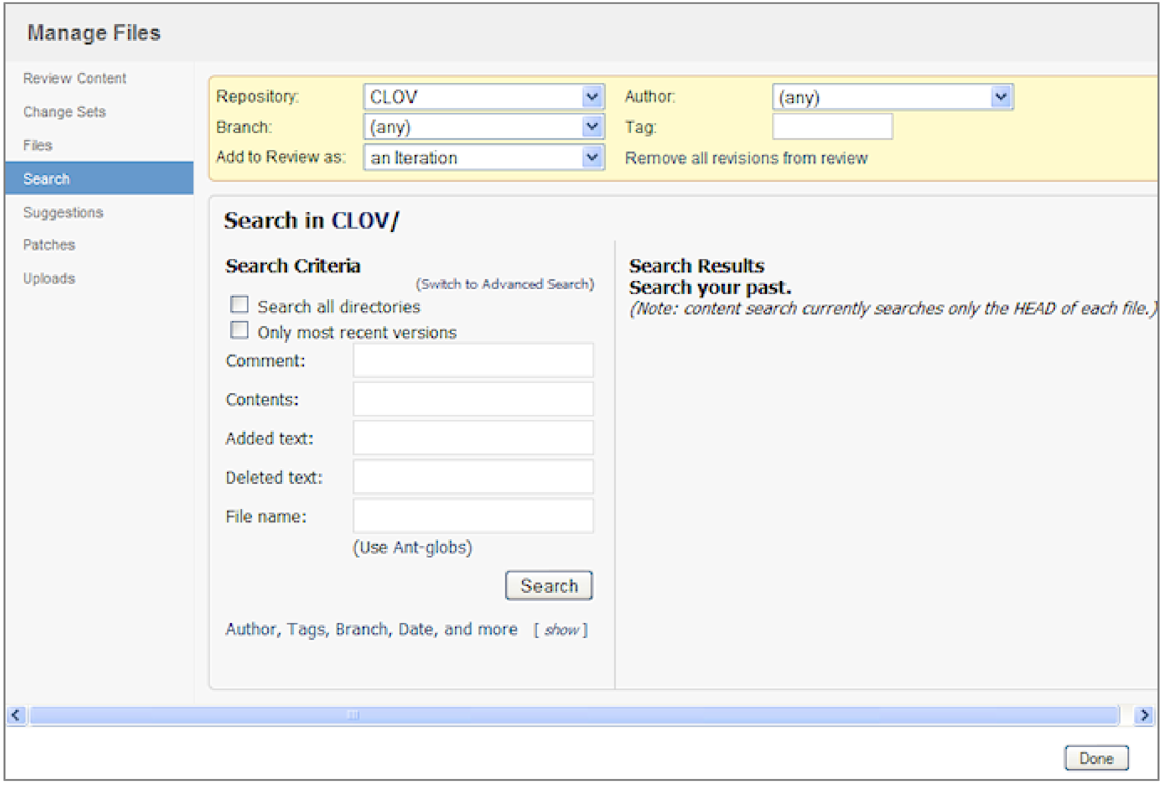
Files in CLOV/

No Files

Searching in the Manage Files View

ⓘ The 'Search' view is available when using FishEye with Crucible.

Screenshot: The Search View in the Manage Files Dialog



Manage Files

Repository: CLOV
Branch: (any)
Add to Review as: an Iteration
Author: (any)
Tag:

Search in CLOV/

Search Criteria

☐ Search all directories
☐ Only most recent versions

Comment:
Contents:
Added text:
Deleted text:
File name:
(Use Ant-globs)

Search Results
Search your past.
(Note: content search currently searches only the HEAD of each file.)

Review Content
Change Sets
Files
Search
Suggestions
Patches
Uploads
Remove all revisions from review
(Switch to Advanced Search)
Search
Author, Tags, Branch, Date, and more [show]
Done

If you are not certain about which changesets/revisions/files to include in a review, use the Search view to find them.

Adjust the search filters to find the files you need. If the simple filters are not enough, read about EyeQL queries in the FishEye documentation.

After you have chosen the files, click the '**Review Content**' view to see the files that are included in the review.

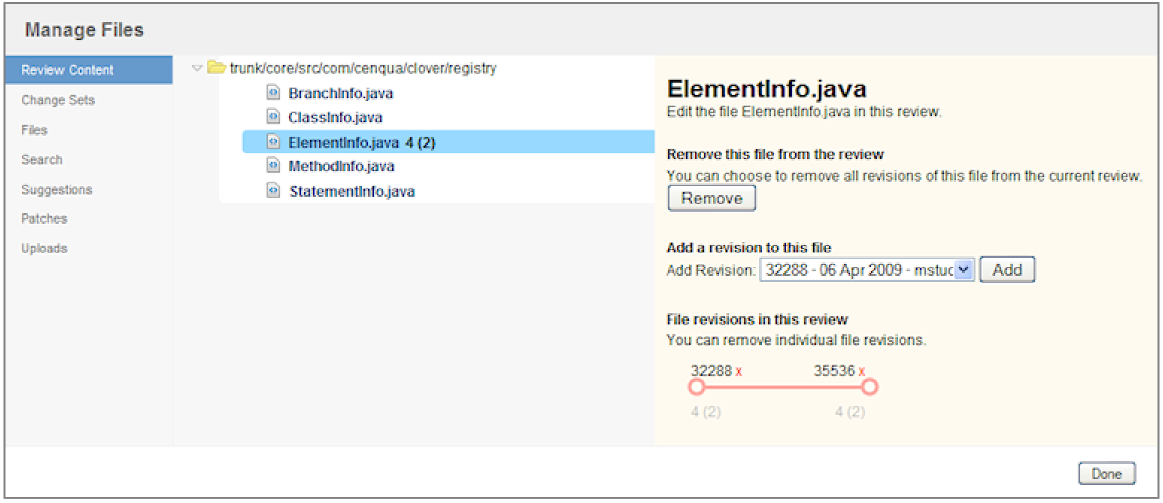
Read the FishEye documentation for more information about the searching your repository.

Viewing the Content Selected for Review

To look at the list of content selected for the review, click the **'Review Content'** option on the left navigation bar. The Review Content view opens.

In this view, you can look over the list of items selected for review. For each item, you can click its name to see which revisions are included in the review. In that view, you can also remove individual files or content items completely (by clicking the small red '**X**' icons over each item), as well as adding or removing individual revisions.
Note that you cannot removed items when they already have comments added to them.

Screenshot: The Review Content View in the Manage Files Dialog



Manage Files

trunk/core/src/com/cenqua/clover/registry
- BranchInfo.java
- ClassInfo.java
- ElementInfo.java 4 (2)
- MethodInfo.java
- StatementInfo.java

ElementInfo.java
Edit the file ElementInfo.java in this review.

Remove this file from the review
You can choose to remove all revisions of this file from the current review.

Add a revision to this file
Add Revision: 32288 - 06 Apr 2009 - mstuc

File revisions in this review
You can remove individual file revisions.

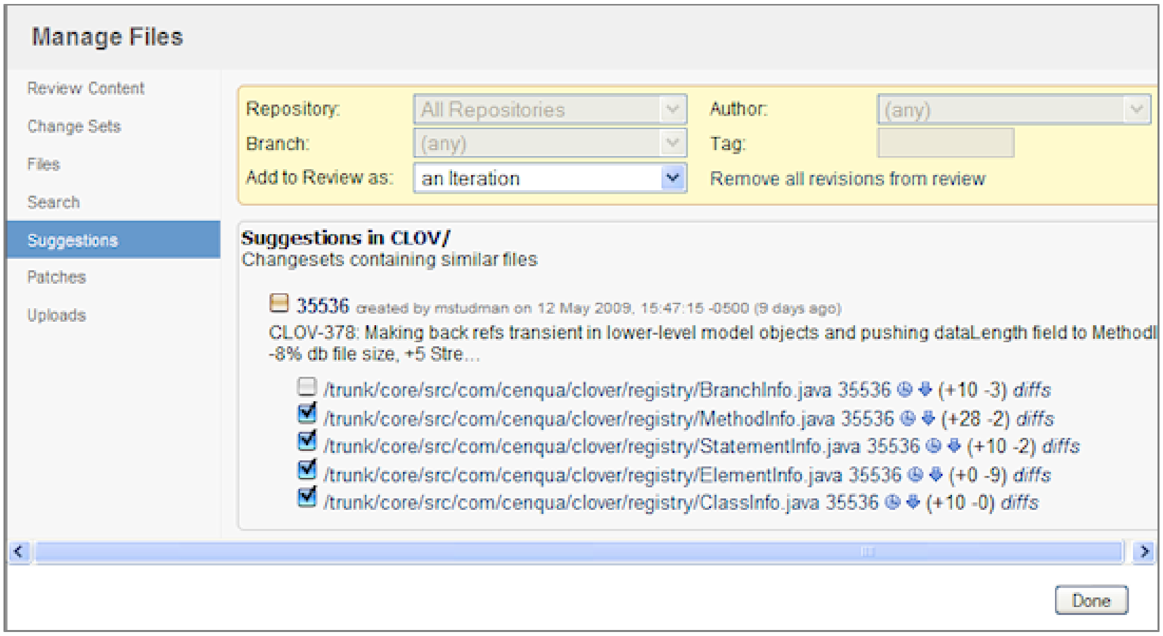
Using the Suggestions Feature When Adding Files to a Review

Crucible can make intelligent suggestions when you are creating a review. Once you have selected some files, click '**Suggestions**' in the left navigation bar to let Crucible analyse your selection and make some recommendations to you.

The Suggestions feature logic is based around the following:

- Most recent versions: If a newer version of a file exists, Crucible will suggest that you add it to the review.
- Similar files: Files with a similar filename may be of relevance to your review; Crucible will show them to you.

Screenshot: The Suggestions View in the Manage Files Dialog



Repository: All Repositories
Branch: (any)
Add to Review as: an Iteration
Author: (any)
Tag:

Suggestions in CLOV/
Changesets containing similar files

35536 created by mstudman on 12 May 2009, 15:47:15 -0500 (9 days ago)
CLOV-378: Making back refs transient in lower-level model objects and pushing dataLength field to Methodl
-8% db file size, +5 Stre...

- □ /trunk/core/src/com/cenqua/clover/registry/BranchInfo.java 35536 (+10 -3) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/MethodInfo.java 35536 (+28 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/StatementInfo.java 35536 (+10 -2) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ElementInfo.java 35536 (+0 -9) *diffs*
- ☑ /trunk/core/src/com/cenqua/clover/registry/ClassInfo.java 35536 (+10 -0) *diffs*

Adding Patch Files to a Review

For a full explanation of the '**Patch View**' functions, read about creating a patch review.

Uploading Files to a Review

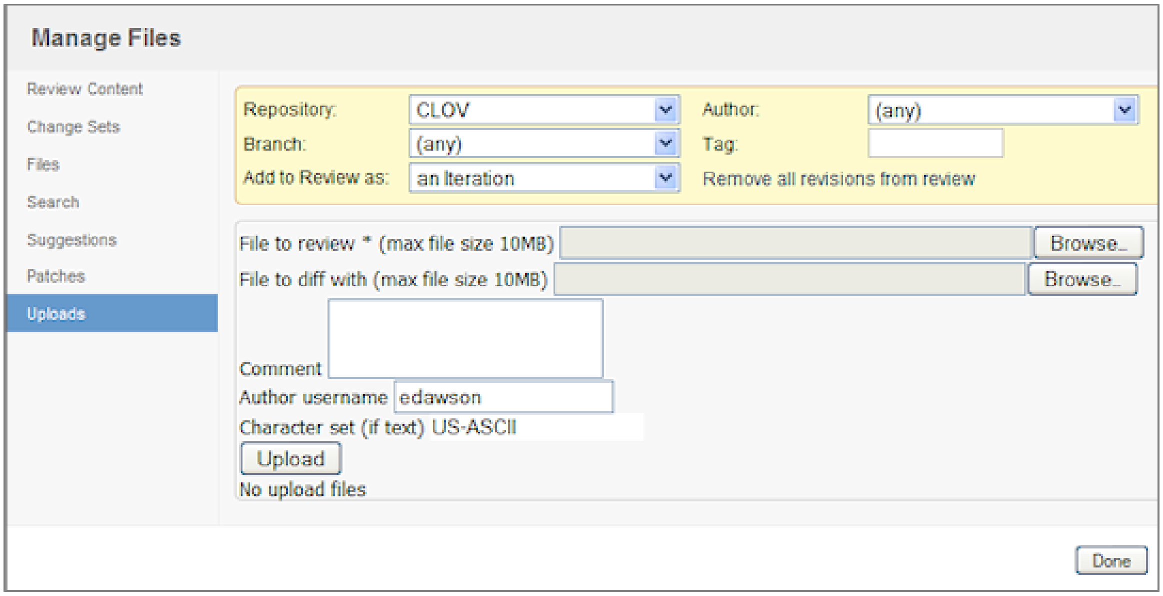
You can upload additional files to be used in the review, including binary files, images or code files that are not stored in a version control repository.

The '**Upload**' view contains various controls to help you do this. These are listed in the table below.

File to review	By clicking '**Browse**', you can find the file you want to review.
File to diff with	By clicking '**Browse**', you can find the file you want to compare against the first file.
Comment	Here, you can put in an identifying comment about this particular piece of work.
Author username	Enter the author username into this field.
Character Set (if any)	In the drop-down list, you can choose the character set being used. '**US-ASCII**' is the default.

When you've made your selection, simply click the '**Upload**' button. Once complete, a list of uploaded files is displayed at the bottom of the screen.

Screenshot: Upload View in the Manage Files Dialog

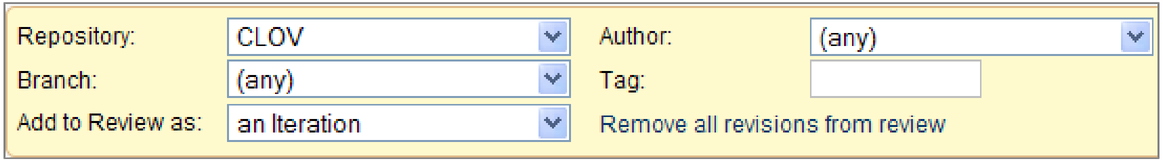


Repository: CLOV
Branch: (any)
Add to Review as: an Iteration
Author: (any)
Tag:
Remove all revisions from review
File to review * (max file size 10MB)
File to diff with (max file size 10MB)
Comment
Author username edawson
Character set (if text) US-ASCII
No upload files

Choosing the Form of the Review

When adding files to a review, you can set the form of review taking place. Several options are listed along the top of the 'Manage Files' dialog, in the **Add to Review as:** drop-down menu:

Screenshot: File Addition Options in the Manage Files Dialog



Repository: CLOV
Branch: (any)
Add to Review as: an Iteration
Author: (any)
Tag:

Whole File	Adds the entire file with all content, rather than just a diff with context.
Diff to Previous Version	This is the default behaviour, and how previous versions of Crucible worked by default.
An Iteration	This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
Diff to Last Reviewed Version	This creates a review with a diff to the last reviewed changeset.
Diff to... (a particular revision)	This allows you to specify the review to show the differences between two specific versions of a file. This is useful in various situations. For example, you may want to review all changes since you branched your project.
Whole file	The review content is not limited by changesets, showing the entire file contents.

Adding Reviewers

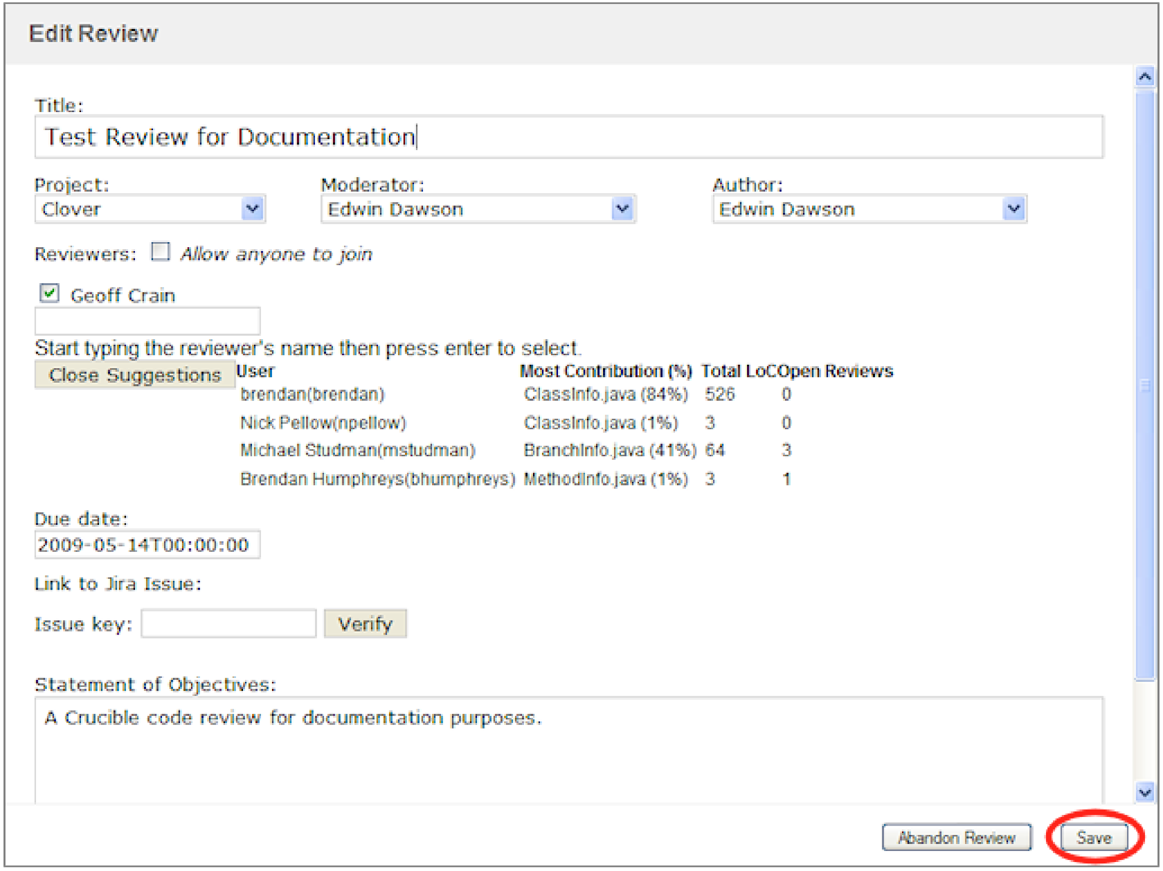
This page explains how to add reviewers to a new review, after it has been created.
See Creating a Review for information about creating reviews.

On this page:

Entering Basic Information

Once a review has been created, the Edit Review dialog opens.

Screenshot: The Edit Review dialog



Edit Review

Title: Test Review for Documentation

Project: Clover

Moderator: Edwin Dawson

Author: Edwin Dawson

Reviewers: □ *Allow anyone to join*

☑ Geoff Crain

Start typing the reviewer's name then press enter to select.

Close Suggestions

User	Most Contribution (%)	Total LoC	Open Reviews
brendan(brendan)	ClassInfo.java (84%)	526	0
Nick Pellow(npellow)	ClassInfo.java (1%)	3	0
Michael Studman(mstudman)	BranchInfo.java (41%)	64	3
Brendan Humphreys(bhumphreys)	MethodInfo.java (1%)	3	1

Due date: 2009-05-14T00:00:00

Link to Jira Issue:

Issue key: Verify

Statement of Objectives:

A Crucible code review for documentation purposes.

In the Edit Review dialog, the author enters information needed for the review. This includes entering a title and description for the review, a due date and the key for a related JIRA issue (if any). The project, moderator and author are pre-selected (for this example, the author should select himself as a moderator.

You must also select reviewers.

Adding Reviewers

Before a review can be issued to reviewers, you must decide who can review it. When adding reviewers, you can add registered users immediately. The usernames will auto-complete, showing partial matches before you finish typing. You can quickly select one of the matches shown with the keyboard arrow keys, pressing Enter or Tab to add them to the review.

In addition, you can easily invite external users who do not yet have accounts in Crucible to take part by typing their email address into the **Reviewers** field.

Adding Users to a Review

Select users by typing names into the text field under **Reviewers**. Crucible will show a list of matches. Press Enter to select one after each entry.

ℹ Crucible will suggest users that have contributed to the files you've selected and also don't have a lot of open reviews.

Clicking the 'Save' button will save the review as a draft for later issue.

You can also decide to allow any registered user to add themselves as a reviewer in the review. To enable this option, put a check next to 'Allow anyone to join'.

Inviting Non-Registered Users to the Review

You can invite users who don't have a Crucible account to join a review.

⚠ There are two prerequisites:

1. FishEye's SMTP server must be configured and capable of sending email.
2. The setting '**Built-in Public Sign-up**' must be set to '**ON**'. This setting can be accessed by opening the '**Admin Menu**', then clicking '**Security**' under '**Global Settings**' on the left navigation bar.

To invite an external user to a review,

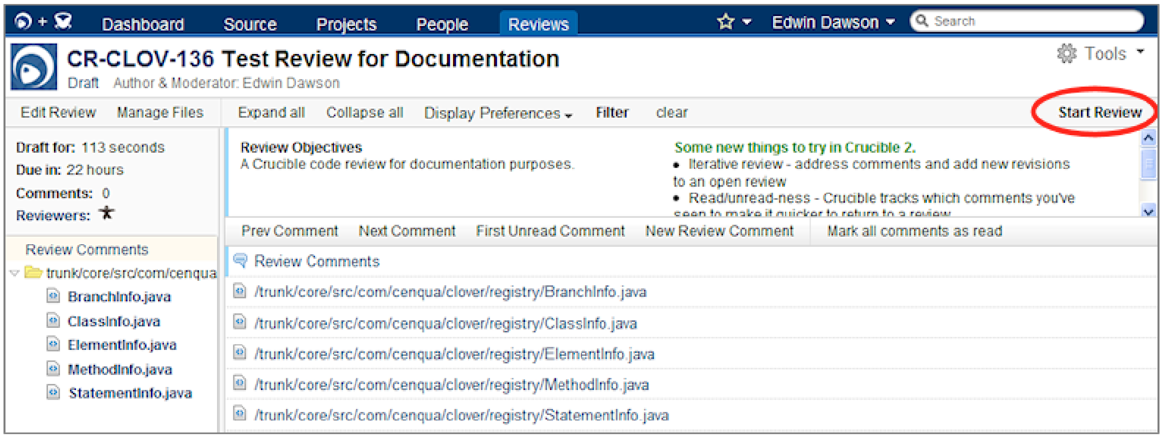
- a.
  - i.
    1. Create a new review.
    2. On the '**Create New Review**' screen, simply type the user's email address into the '**Reviewers**' field, then press Enter to select.
    3. Click Save to save the draft review. The users are not sent any information at this time.
    4. When you click '**Start Review**', this is when all email invites and notifications are sent out.
    5. The external user will receive an email address from the Crucible server, containing a special URL that they can visit.
    6. When the user loads the URL they received via email, they are taken to a special Crucible log in screen. On this screen, the user can create a new account that will be linked to the current email address. (If they already have a Crucible account under another address, they can simply sign-in with that username and password.)
    7. When the user has successfully created a Crucible account, they will be able to access the review(s) associated with their email address and take part.

ℹ You can enter multiple addresses separated by commas, allowing you to paste in a list of email addresses from your favourite email application.

When finished, the author clicks '**Save**'. The review will now be created in a draft form.

Checking the Draft and Starting the Review

Screenshot: A newly created Crucible review



CR-CLOV-136 Test Review for Documentation

Draft Author & Moderator: Edwin Dawson

Draft for: 113 seconds
Due in: 22 hours
Comments: 0
Reviewers:

Review Objectives
A Crucible code review for documentation purposes.

Some new things to try in Crucible 2.
- Iterative review - address comments and add new revisions to an open review
- Read/unread-ness - Crucible tracks which comments you've seen to make it quicker to return to a review

Review Comments
- trunk/core/src/com/cenqua
  - BranchInfo.java
  - ClassInfo.java
  - ElementInfo.java
  - MethodInfo.java
  - StatementInfo.java

Review Comments
/trunk/core/src/com/cenqua/clover/registry/BranchInfo.java
/trunk/core/src/com/cenqua/clover/registry/ClassInfo.java
/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java
/trunk/core/src/com/cenqua/clover/registry/MethodInfo.java
/trunk/core/src/com/cenqua/clover/registry/StatementInfo.java

The draft review opens. In the draft stage, the author can check the contents of the review files to ensure they are correct and put in any notes for reviewers as comments. During the draft phase, no notification email is sent out to reviewers. Once the author is finished with the draft phase, he clicks '**Start Review**'.

The review will now be started and notification email will go out to all participants. Crucible will now send out an email notification to all the participants. This lets them know that the review is under way and prompts them to take action, providing a URL for direct access to the review. (You can also subscribe to an RSS feed.)

Next Steps

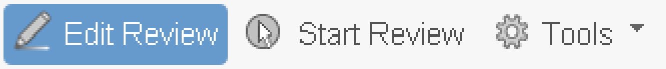
You can now begin Performing the Review.
If you have a moderator controlling your review process, you can move onto Issuing a Review.

Removing Reviewers From An Active Review

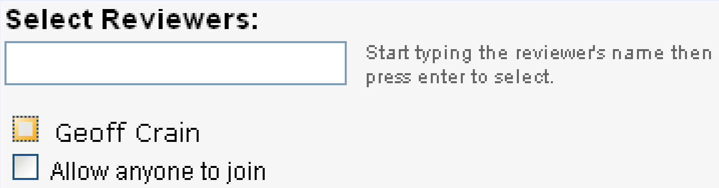
Reviewers can be removed from an active review at any time.

To remove a reviewer from an active review,

1. Log in to Crucible as the review creator or moderator.
2. Open the review in question.
3. Click the '**Edit Review**' button:



4. Find the user you want to remove and click the checkbox next to their name to remove them (so that the checkbox becomes empty):



Select Reviewers:

Start typing the reviewer's name then press enter to select.

□ Geoff Crain
□ Allow anyone to join

5. Click '**Save**'.
6. The user will be removed from the review and notified by email.

ℹ Reviewers can be only removed by the review creator or moderator.

⚠ You cannot remove the review creator or moderator from

the review.

Issuing a Review

Issuing a review in Crucible is known as **approving the review**. This simply means formally starting the review and inviting people to take part.

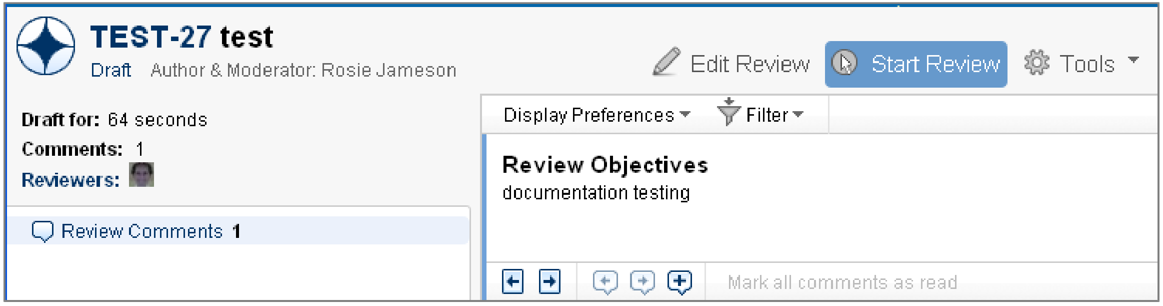
Once you have selected the reviewers, the next stage is to notify the reviewers and the author (if different to the moderator) that they can start reviewing. The review has been in 'Draft' state until this point. Only the moderator has the permission to approve a review.

To issue the review:

- If you are the moderator of the review, click the '**Start Review**' button. Or;
- If you are not the moderator of your review, click '**Send to Moderator**'. This changes the state to 'Require Approval' and notifies the moderator. The moderator can change any aspect of the review before approving it.

Once the review has been approved, the review state becomes '**Under Review**'.

Screenshot: Starting a Review



TEST-27 test

Draft Author & Moderator: Rosie Jameson

Draft for: 64 seconds

Comments: 1

Reviewers:

Review Objectives

documentation testing

Note that only people with the '**Approve**' permission can issue a review.

Performing the Review

This page describes how to find and manage the Crucible reviews that relate to you.

On this page:

- Browse Your Reviews Under the 'Dashboard' Tab
- Browse All Reviews Under the 'Reviews' Tab
- When files change during a review
- Next steps

> **Deciding what needs to be reviewed**
> Crucible does not dictate how or what to review. It simply provides a mechanism to record comments. The '**Statement of Objective**' is a brief description of what the review is intended to achieve.

Browse Your Reviews Under the 'Dashboard' Tab

When you first load Crucible, the '**Dashboard**' screen will load, which shows your current reviews and other activity related to you.

Use the Crucible '**Dashboard**' to manage your reviews. Read the overview on filtering your view.

Active reviews are listed on each reviewer's dashboard under the default '**To Review**' filter. Reviews are listed under '**Out for Review**' until all reviewers indicate they are complete. Then the reviews move to the '**To Summarize**' list.

Read more about using the Dashboard tab.

Browse All Reviews Under the 'Reviews' Tab

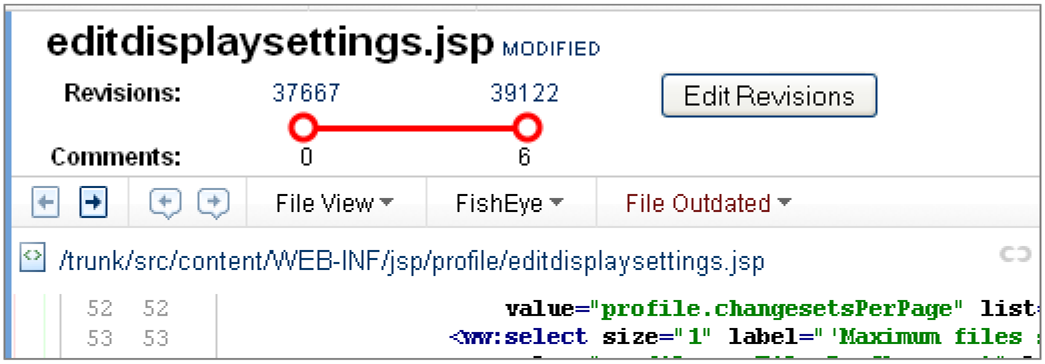
All reviews that involve you in any role are listed when you click '**Open**' or '**Closed**' in the left navigation bar. For instance, use the 'My Reviews' --> 'Open' filter to locate a review that doesn't require further action from you, but is still under way.

If email notifications are enabled (see SMTP settings in the FishEye documentation), reviewers will receive an email with information about the review. Click the link within the email to go directly to the review.

When files change during a review

If a file in the repository changes during a review, Crucible will visually alert you by showing the '**File Outdated**' menu:

Screenshot: Visual Cue for Updated Repository Files

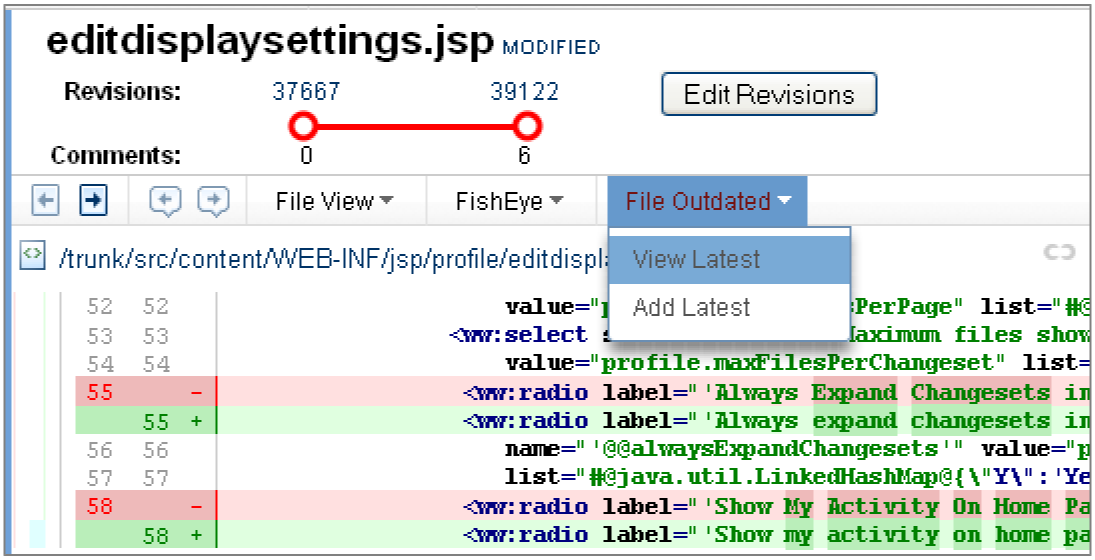


/trunk/src/content/WEB-INF/jsp/profile/editdisplaysettings.jsp

```
                    value="profile.changesetsPerPage" list
                <ww:select size="1" label="'Maximum files
```

From the '**File Outdated**' menu, you can choose to view the latest revision of the updated file, or add the latest revision to the review:

Screenshot: 'File Outdated' menu



/trunk/src/content/WEB-INF/jsp/profile/editdispl

```
                    value="p              PerPage" list="#@
                <ww:select s              laximum files show
                    value="profile.maxFilesPerChangeset" list=
 -               <ww:radio label="'Always Expand Changesets in
 +               <ww:radio label="'Always expand changesets in
                    name="'@@alwaysExpandChangesets'" value="p
                    list="#@java.util.LinkedHashMap@{\"Y\":'Ye
 -               <ww:radio label="'Show My Activity On Home Pa
 +               <ww:radio label="'Show my activity on home pa
```

Next steps

- Adding Comments

- Flagging Defects
- Completing your Review
- Sending all of a Review's Comments via Email
- Using JIRA Integration in Crucible Reviews

Adding Comments

Comments can be added at the level of a review, revision, or line. You can also reply to a comment.

On this page:

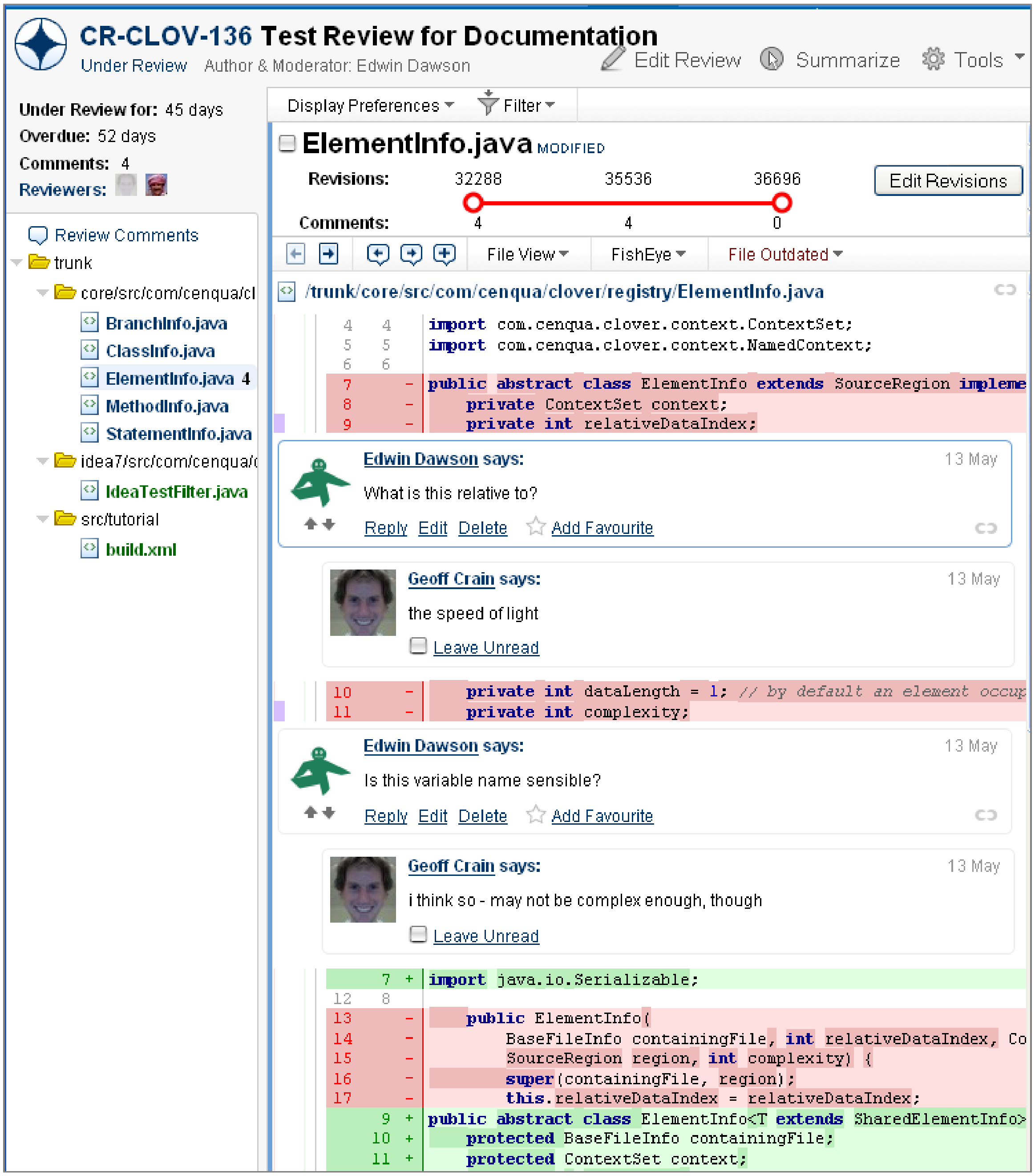
- Locating existing comments
- Adding a Comment
- Draft Comments

Locating existing comments

The number shown next to a filename, in the left-hand column of the screen, indicates the number of comments that apply to that file.

(The number of unread comments, if there are any, is shown in brackets.)

Screenshot: Comments



Under Review for: 45 days
Overdue: 52 days
Comments: 4
Reviewers:

ElementInfo.java MODIFIED

Revisions:	32288	35536	36696
Comments:	4	4	0

/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java

```
import com.cenqua.clover.context.ContextSet;
import com.cenqua.clover.context.NamedContext;

-   public abstract class ElementInfo extends SourceRegion impleme
-       private ContextSet context;
-       private int relativeDataIndex;
```

Edwin Dawson says: 13 May

What is this relative to?

Geoff Crain says: 13 May

the speed of light

```
-       private int dataLength = 1; // by default an element occup
-       private int complexity;
```

Edwin Dawson says: 13 May

Is this variable name sensible?

Geoff Crain says: 13 May

i think so - may not be complex enough, though

```
+   import java.io.Serializable;

-       public ElementInfo(
-           BaseFileInfo containingFile, int relativeDataIndex, Co
-           SourceRegion region, int complexity) {
-           super(containingFile, region);
-           this.relativeDataIndex = relativeDataIndex;
+   public abstract class ElementInfo<T extends SharedElementInfo>
+       protected BaseFileInfo containingFile;
+       protected ContextSet context;
```

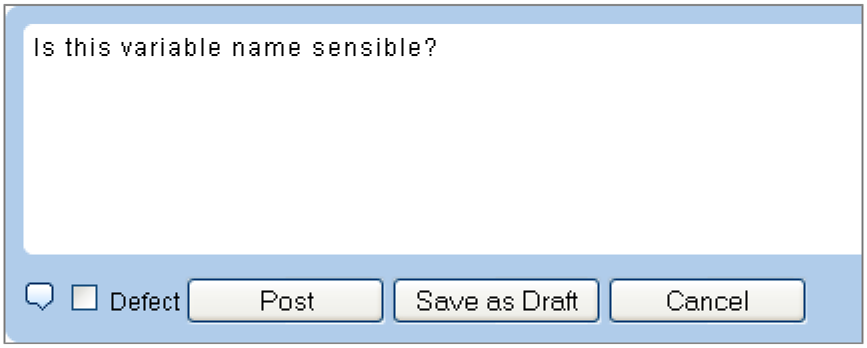
Adding a Comment

- To add a comment that applies to the whole review, select the '**Review Comments**' line in the left-hand navigation panel, then click the following icon: ⊕
- To add a comment that applies to a revision/change, select the filename in the left-hand navigation panel, then click the following icon: ⊕
- To add a source-level comment, expand the source view then click a line of code. You can click and drag to select multiple lines from one revision or diff, or click individual lines to select/deselect them. The comment will appear in the source at the last line selected. Hover over the comment to highlight the selected lines.
    - ℹ To select text on the page without adding a comment, hold down the **Alt** button while dragging the cursor.
- To reply to a comment, click the '**Reply**' link at the bottom of the comment.

⚠ Only people with the '**Comment**' permission can add comments.

ⓘ Read about flagging defects too.

Screenshot: Adding a Comment

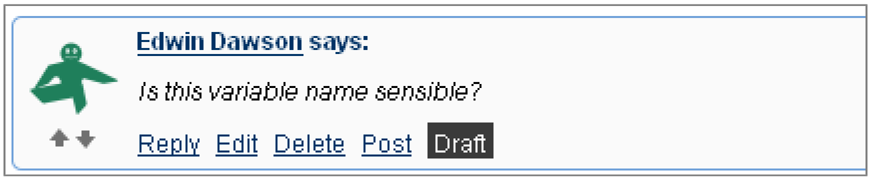


Is this variable name sensible?

Draft Comments

You can save your comments as drafts and then edit them later. When you complete the review , you will be prompted to post/discard/edit any remaining draft comments.

Screenshot: Draft comments



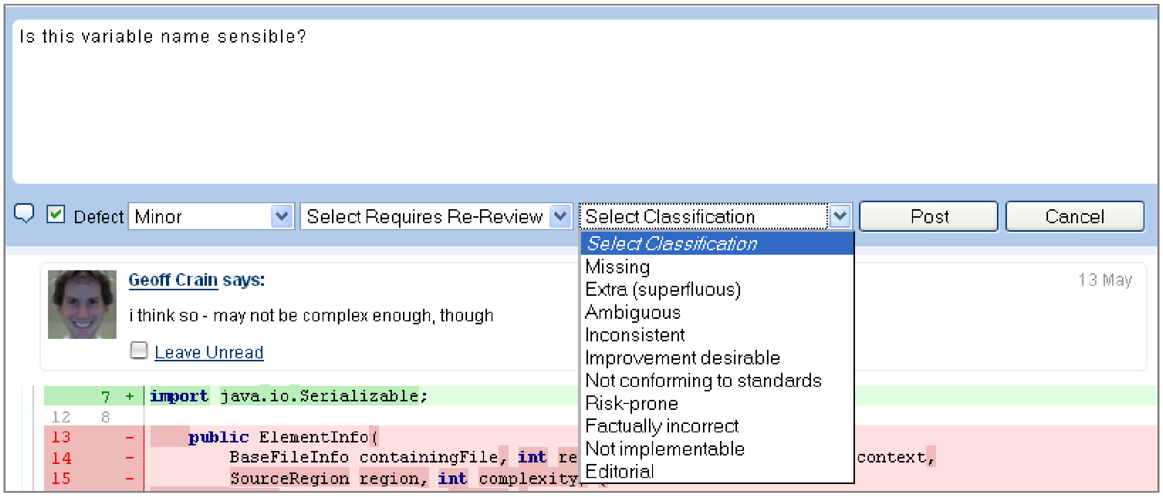
Edwin Dawson says:

Is this variable name sensible?

Flagging Defects

Any comment that is not a reply to an existing comment can be flagged as a **defect** by its author.

Screenshot: Defects



Is this variable name sensible?

☑ Defect: Minor

Select Requires Re-Review

Select Classification

Geoff Crain says:

i think so - may not be complex enough, though

13 May

□ Leave Unread

```
+ import java.io.Serializable;

-     public ElementInfo(
-         BaseFileInfo containingFile, int re                    context,
-         SourceRegion region, int complexity,
```

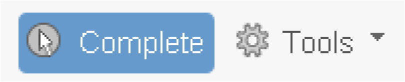
You may want to mark comments as defects to associate defect classifications, or simply to highlight to the author or moderator that the issue you raised in your comment requires attention.

ⓘ Crucible intentionally does not mandate how defects are to be used. The Crucible administrator can customise the defect classifications.

Completing your Review

Once each reviewer has added comments to the review and has nothing further to add, the next step is to **Complete** their individual review.

To complete your individual review, go to the review and click the '**Complete**' button at the right of the screen, next to the '**Tools**' menu:



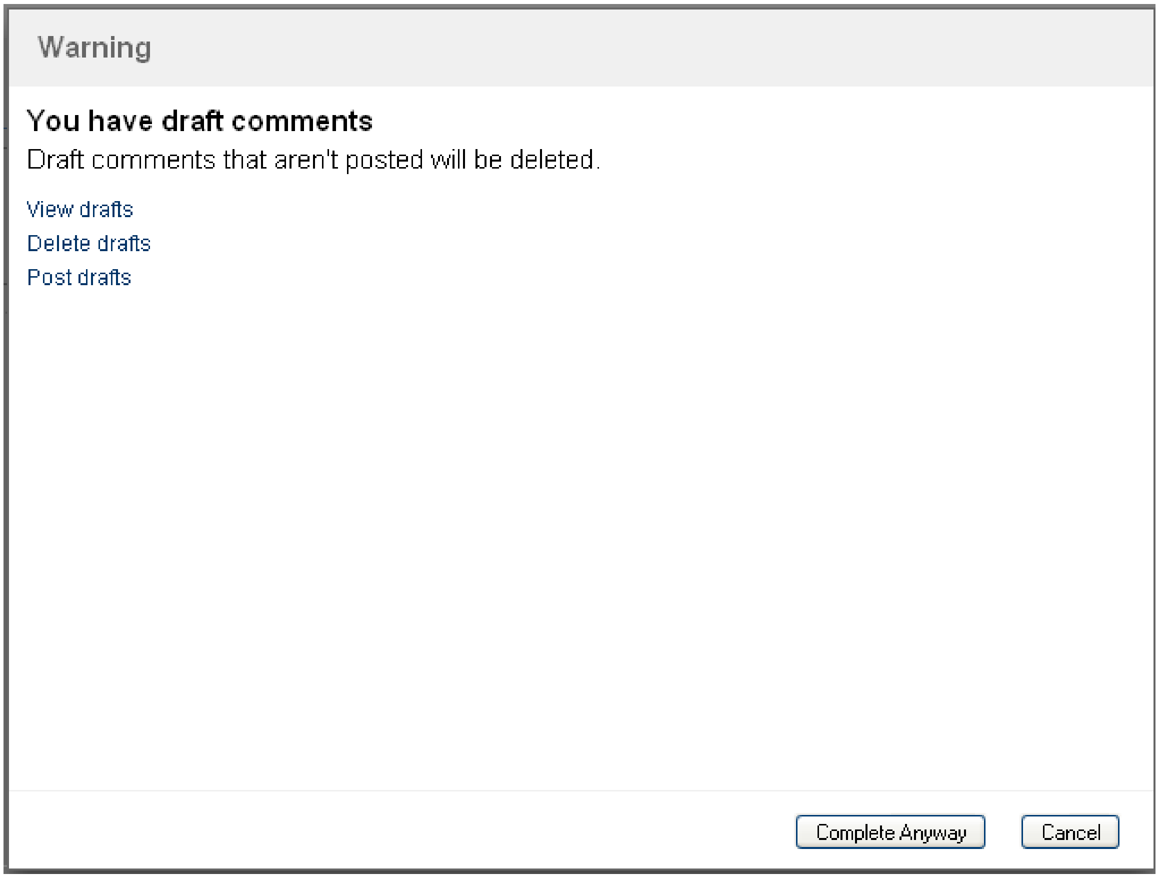
ⓘ Only people with the '**Complete**' permission can complete a review.

This notifies the moderator (via email if configured) that you have completed your review.

Reviewers can still continue to add comments until the moderator summarises the review. The moderator does **not** have to wait for all reviewers to complete their individual reviews before summarising.

If you have any draft comments, you will be prompted to post/discard/edit any comments before completing the review.

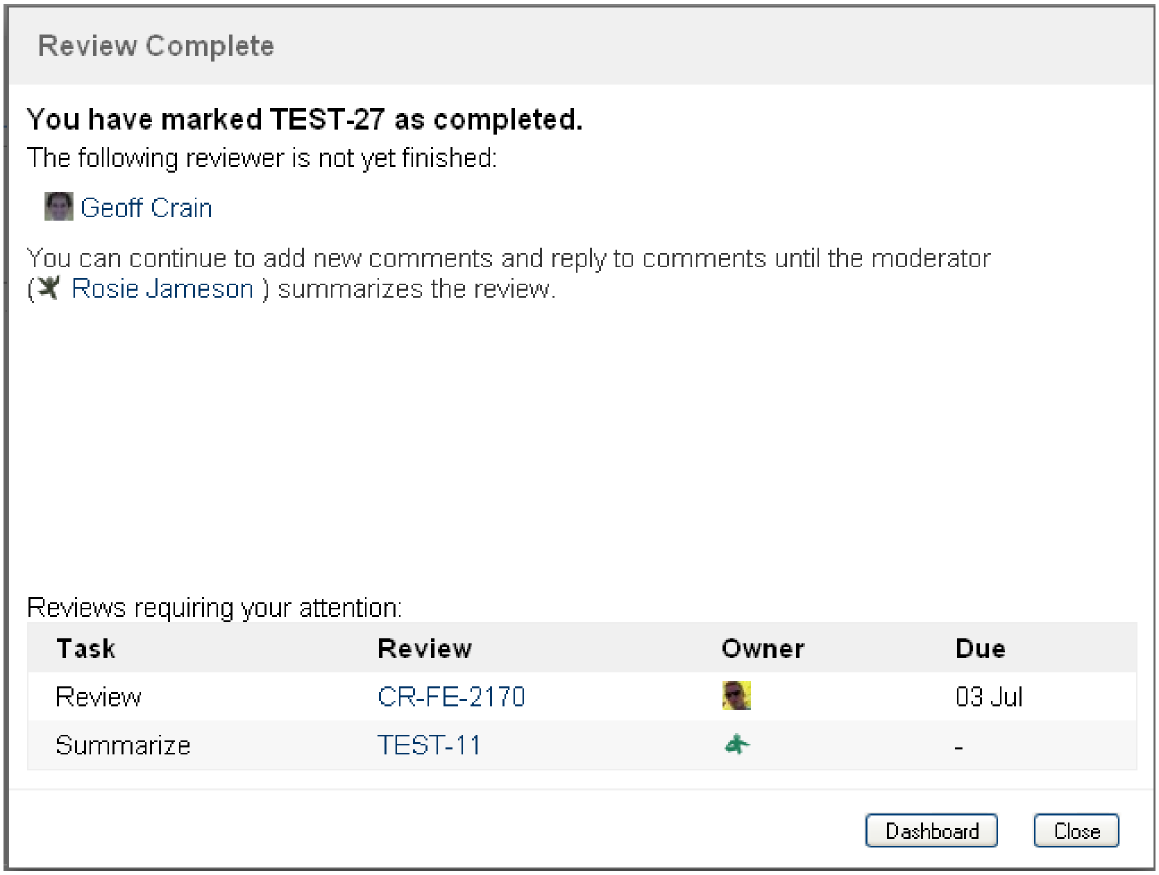
Screenshot: Draft comments


Warning

You have draft comments

Draft comments that aren't posted will be deleted.

Screenshot: Review complete



Review Complete

You have marked TEST-27 as completed.

The following reviewer is not yet finished:

Geoff Crain

You can continue to add new comments and reply to comments until the moderator (Rosie Jameson) summarizes the review.

Reviews requiring your attention:

Task	Review	Owner	Due
Review	CR-FE-2170		03 Jul
Summarize	TEST-11		-

Sending all of a Review's Comments via Email

You can send all of the comments from a review to anyone you want via email. You may wish to do this to allow a person outside the review to quickly scan the content of the comments, or oversee the review activity. Alternatively, you may wish to send all participants this information to let them read the current status of the review and its comments in full.

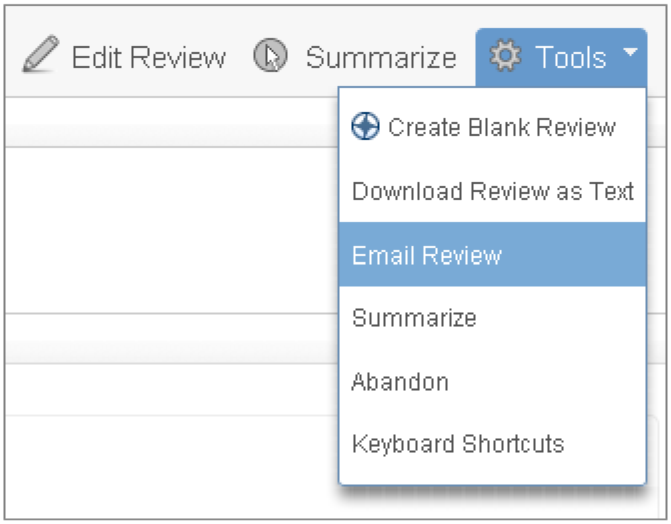
Sending a Review's Comments via Email

To send all of a review's comments via email,

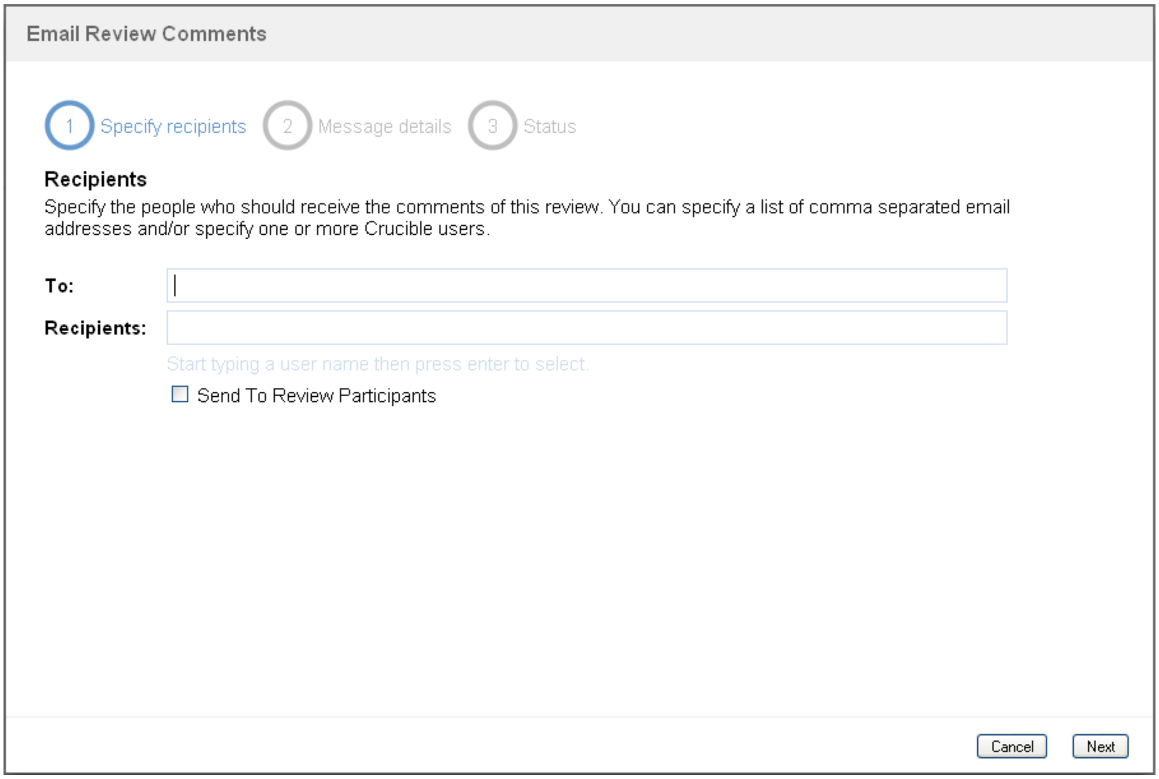
1. In Crucible, navigate to the review in question.
2. From the '**Tools**' menu, select '**Email Review'** (see *Screenshot 1* below).
3. The '**Recipients**' page appears (see *Screenshot 2* below). On that page:
4. In the '**To:**' field you can enter multiple email addresses, separated by commas.
5. In the '**Recipients:**' field, you can type usernames from your Crucible instance to add them to the distribution list. You can also simply tick the '**Send to Review Participants**' check-box to include all of the review's reviewers.
6. When you have finished the distribution list, click the '**Next**' button.
7. The '**Recipients**' page appears (see *Screenshot 3* below). On that page:
8. In the '**To:**' field you can enter multiple email addresses, separated by commas.
9. When you have finished your message, click the '**Send**' button.
10. The '**Status**' page appears (see *Screenshot 4* below), confirming that your email has been sent

Users that are not logged in cannot send email, but instead can view the text content of the review's comments by clicking the '**View Text**' option which will appear instead of '**Email Review**'.

Screenshot 1: The 'Email Review' option in Crucible



Screenshot 2: The 'Recipients' Screen in Crucible



Email Review Comments

Recipients

Specify the people who should receive the comments of this review. You can specify a list of comma separated email addresses and/or specify one or more Crucible users.

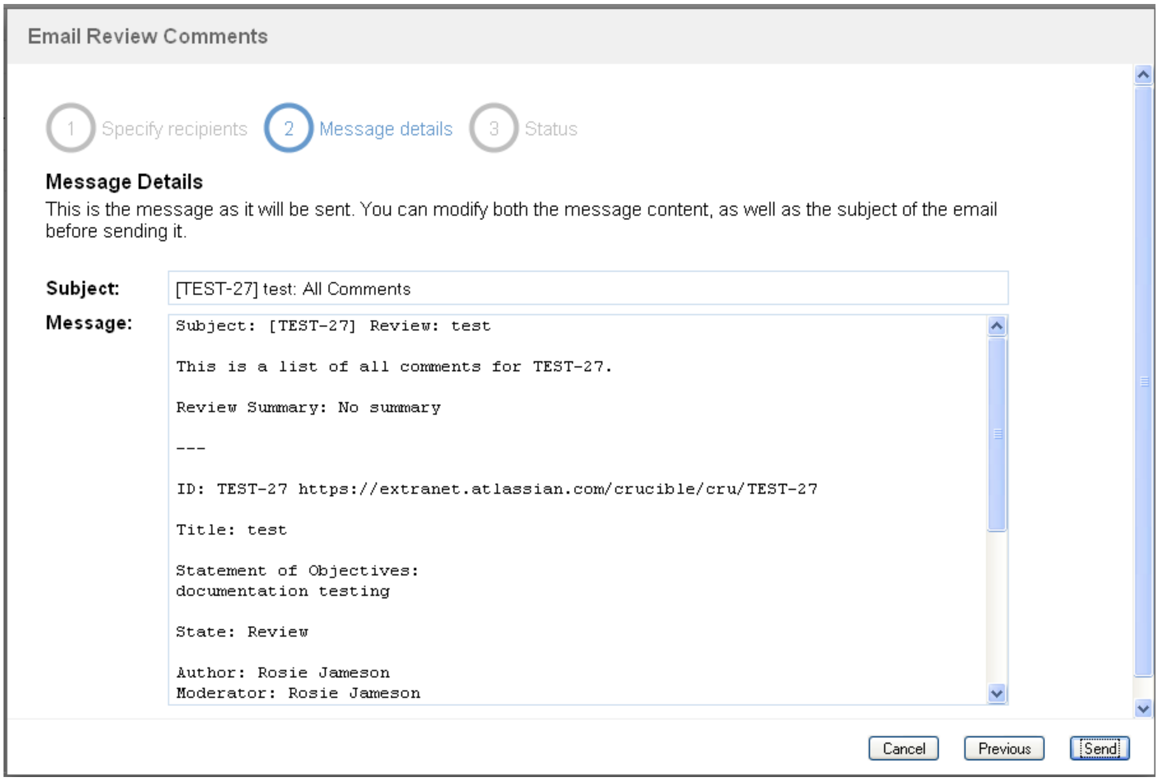
To:

Recipients:

Start typing a user name then press enter to select.

□ Send To Review Participants

Screenshot 3: The 'Message' Screen in Crucible



Message Details

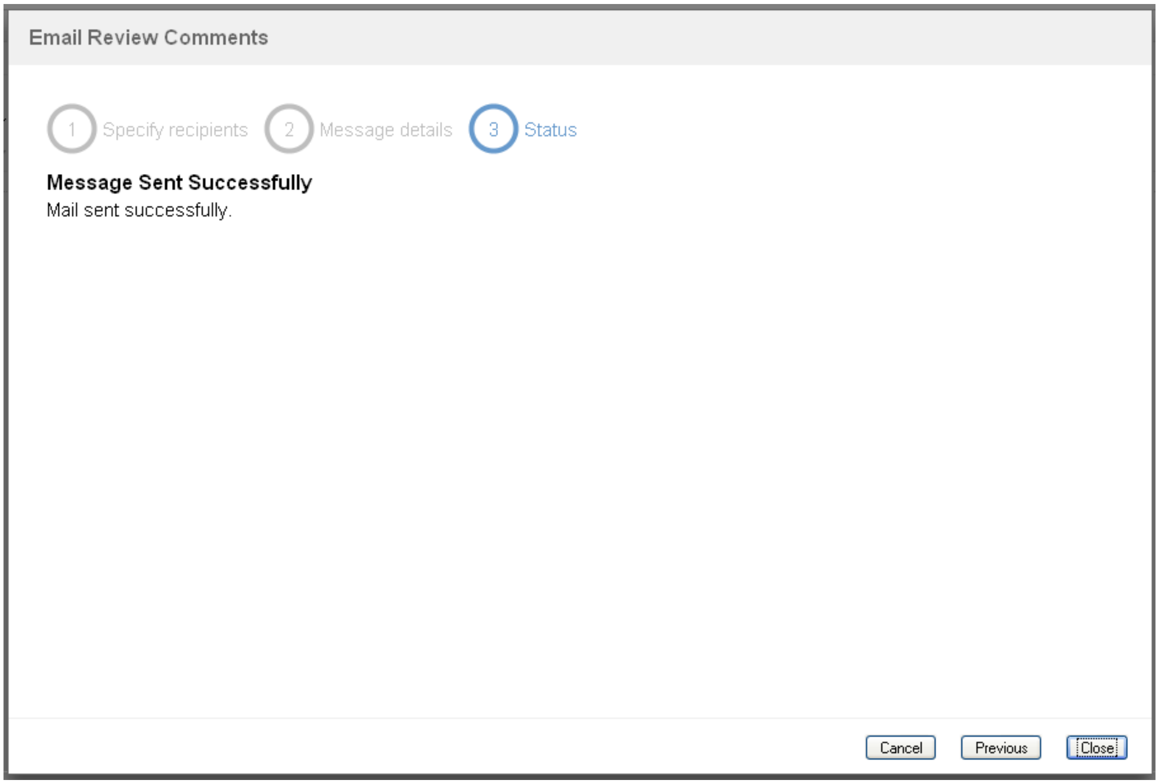
This is the message as it will be sent. You can modify both the message content, as well as the subject of the email before sending it.

Subject: [TEST-27] test: All Comments

Message:

```
Subject: [TEST-27] Review: test

This is a list of all comments for TEST-27.

Review Summary: No summary

---

ID: TEST-27 https://extranet.atlassian.com/crucible/cru/TEST-27

Title: test

Statement of Objectives:
documentation testing

State: Review

Author: Rosie Jameson
Moderator: Rosie Jameson
```

Screenshot 4: The Email Confirmation Screen in Crucible



Message Sent Successfully

Mail sent successfully.

Using JIRA Integration in Crucible Reviews

This page contains information on how to use JIRA integration with Crucible reviews.

On this page:

Before you begin, both your Crucible and JIRA instances must be configured to use make use of these JIRA integration features. Read more.

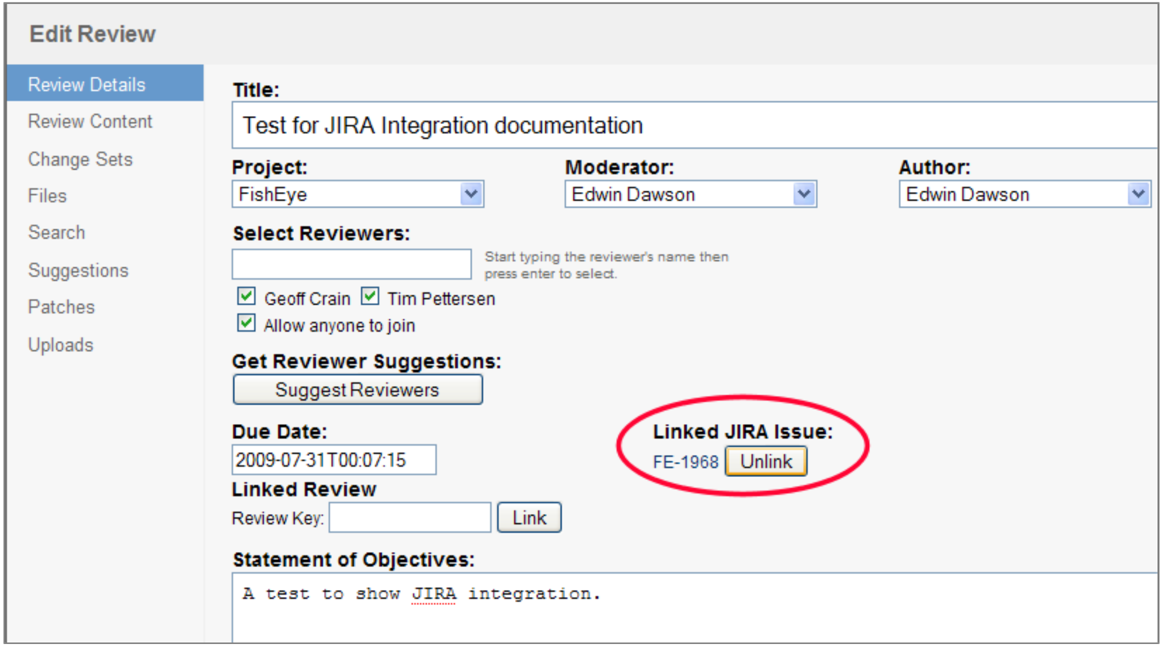
1. Create a JIRA Issue for your Review

To use JIRA integration, you must begin with a JIRA issue that you will use as the parent issue for the review. Crucible will open and close sub-tasks below this parent issue. Once your parent issue is created, make a note of its issue key, e.g. `FE-1968`.

2. Create Your Review and Link it to a JIRA Issue

When creating your review, you have an option called '**Linked JIRA Issue**'. You can specify a different issue key and click '**Link**' to save it. You can also click '**Unlink**' to clear the field and load a different issue key.

Screenshot: Selecting a JIRA Issue When Creating a Review



Title: Test for JIRA Integration documentation

Project: FishEye
Moderator: Edwin Dawson
Author: Edwin Dawson

Select Reviewers:
Start typing the reviewer's name then press enter to select.
☑ Geoff Crain ☑ Tim Pettersen
☑ Allow anyone to join

Get Reviewer Suggestions:

Due Date: 2009-07-31T00:07:15

Linked JIRA Issue: FE-1968

Linked Review
Review Key:

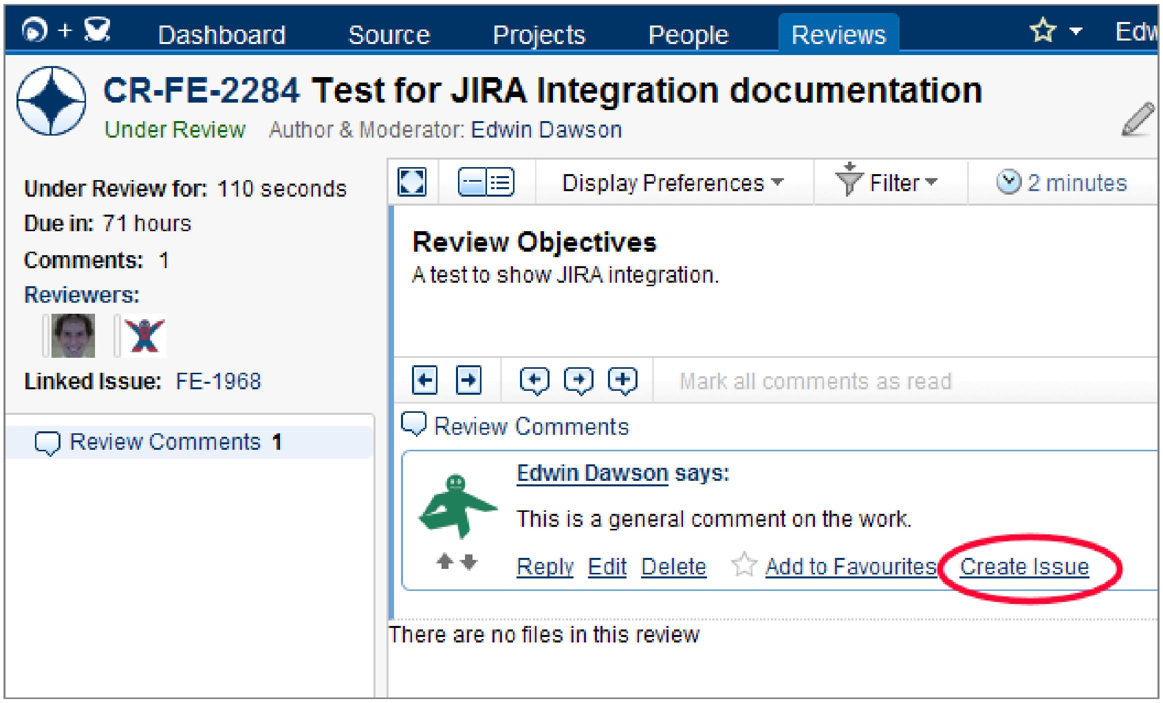
Statement of Objectives:
A test to show JIRA integration.

You can also link a JIRA issue to the review after the review is created. When viewing a review, the top-left hand corner of the screen shows meta-information. One of the links in this area is titled '**Linked JIRA Issue:**' and then a suggested JIRA issue key. Click this link to associate that JIRA issue with this review.

3. Make a Comment or Defect on the Review

Once your review has a linked JIRA issue, create a comment or defect comment anywhere on the review. Once created, the comment actions will show a link titled '**Create Issue**'. You can click that to instantly create a sub-task under the parent JIRA, which will take the content of the comment as its summary.

Screenshot: Selecting a JIRA Issue From a Comment



CR-FE-2284 Test for JIRA Integration documentation

Under Review Author & Moderator: Edwin Dawson

Under Review for: 110 seconds
Due in: 71 hours
Comments: 1
Reviewers:
Linked Issue: FE-1968

Review Objectives

A test to show JIRA integration.

Edwin Dawson says:

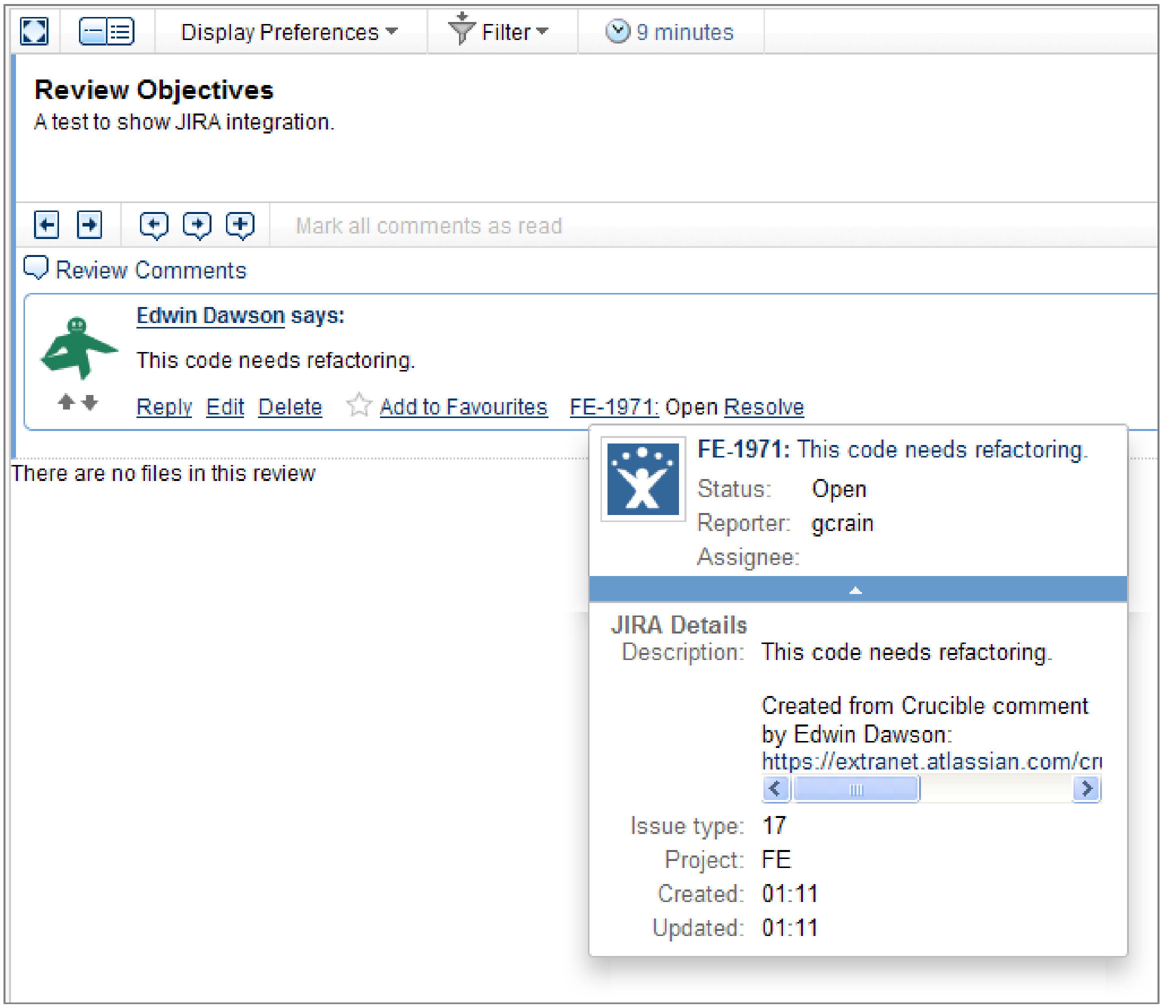
This is a general comment on the work.

There are no files in this review

4. **Click to create a JIRA Sub-Task**

Clicking the '**Create Issue**' link will allow you to create a JIRA sub-task under the parent JIRA issue. The sub-task JIRA issue key, status and default action (i.e. '**Resolve**') will be shown. If you hover your mouse over the JIRA issue key, an information window will show more information and controls relating to that JIRA issue.

Screenshot: The JIRA Hover information window



Review Objectives
A test to show JIRA integration.

Edwin Dawson says:

This code needs refactoring.

FE-1971: Open Resolve

There are no files in this review

FE-1971: This code needs refactoring.

Status: Open
Reporter: gcrain
Assignee:

Description: This code needs refactoring.

Created from Crucible comment by Edwin Dawson: https://extranet.atlassian.com/cru

Issue type: 17
Project: FE
Created: 01:11
Updated: 01:11

Users are mapped to their own accounts when using Trusted Applications. If you don't have the permissions to carry out the default action ('**Resolve**', in this case), an error will occur.

5. Resolve the JIRA through Crucible

Once the work required on your sub-task is completed, simply click the action link provided to signal that this has occurred (e.g. '**Resolve**'). The JIRA issue will be closed.

> If you encounter problems or have trouble using JIRA integration, please read the FAQ page on this topic.

(hidden draft) Using JIRA Integration in Crucible Reviews

> ⚠ This page is a hidden draft for the release of Crucible 2.1

This page contains information on how to use JIRA integration with Crucible reviews.

On this page:

- 1. Create a JIRA Issue for your Review
- 2. Create Your Review and Link it to a JIRA Issue
- 3. Make a Comment or Defect on the Review

- 4. Click to create a JIRA Sub-Task
- 5. Resolve the JIRA issue through Crucible

Before you begin, both your Crucible and JIRA instances must be configured to use make use of these JIRA integration features. Read more.

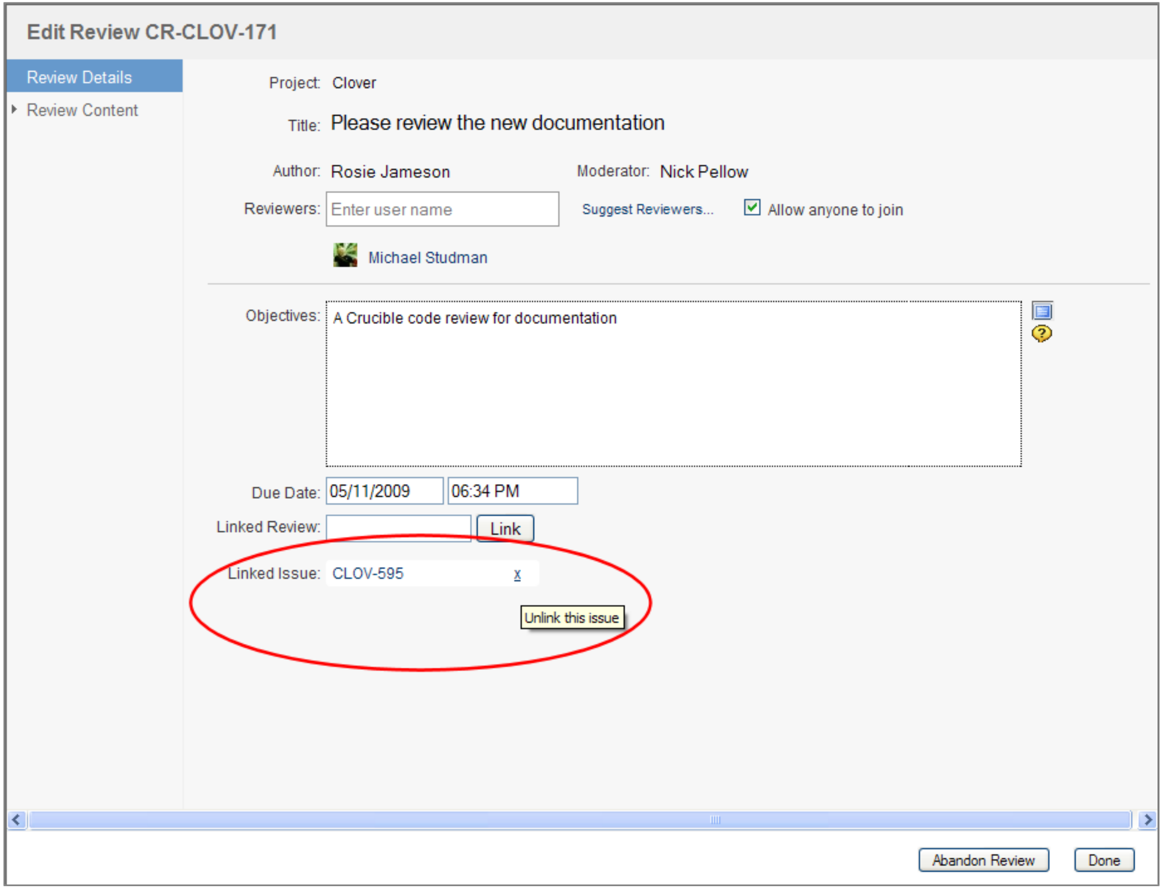
1. Create a JIRA Issue for your Review

To use JIRA integration, you must begin with a JIRA issue that you will use as the parent issue for the review. Crucible will create and resolve sub-tasks belonging to this parent issue. Once your parent issue is created, make a note of its issue key, e.g. `FE-1968`.

2. Create Your Review and Link it to a JIRA Issue

When creating your review, you have an option called '**Linked Issue**'. Crucible may put a suggested JIRA issue key into this field automatically. You can specify a different issue key and click '**Link**' to save it. You can also click the 'x' to clear the field and load a different issue key.

Screenshot: Selecting a JIRA Issue When Creating a Review



Edit Review CR-CLOV-171

Project: Clover

Title: Please review the new documentation

Author: Rosie Jameson

Moderator: Nick Pellow

Reviewers: Enter user name

☑ Allow anyone to join

Michael Studman

Objectives: A Crucible code review for documentation

Due Date: 05/11/2009 06:34 PM

Linked Review:

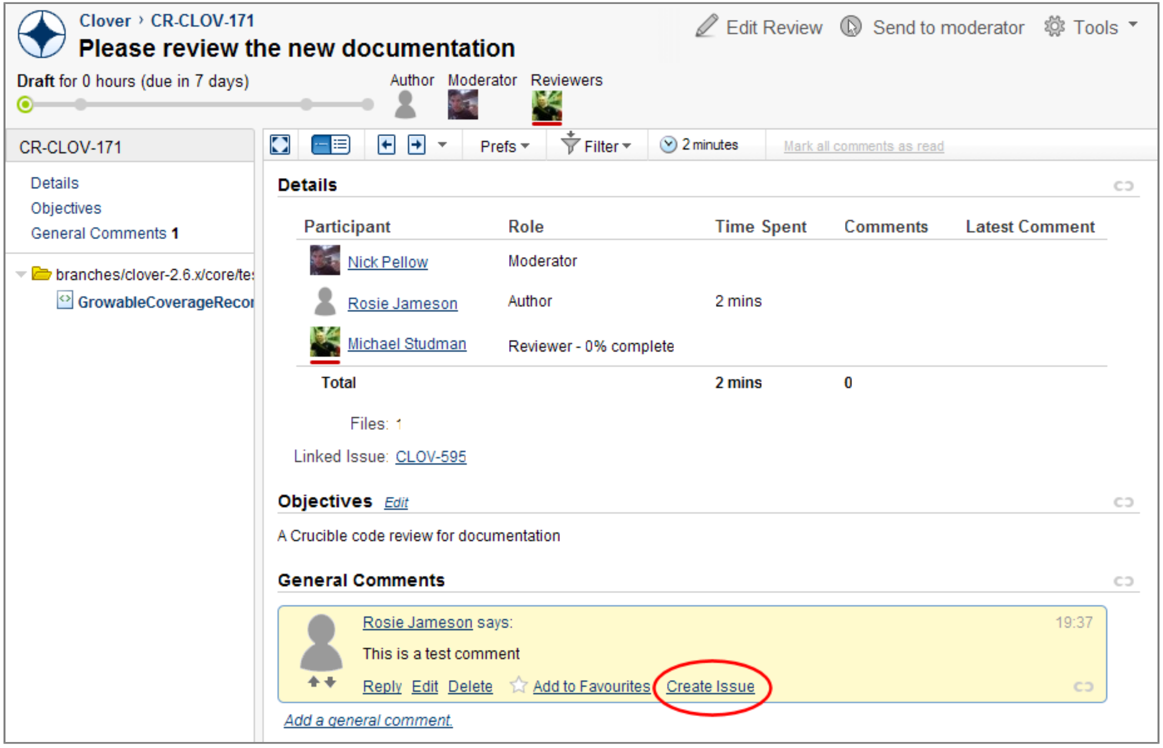
Linked Issue: CLOV-595

You can also link a JIRA issue to the review after the review is created. When viewing a review, the top-left hand corner of the screen shows meta-information. One of the links in this area is titled '**Linked JIRA Issue:**' and then a suggested JIRA issue key. Click this link to associate that JIRA issue with this review.

3. Make a Comment or Defect on the Review

Once your review has a linked JIRA issue, create a comment or defect comment anywhere on the review. Once created, the comment actions will show a link titled '**Create Issue**' (note that this link does not appear on replies — only on new comments). You can click that to instantly create a sub-task under the parent JIRA, which will take the content of the comment as its summary.

Screenshot: Selecting a JIRA Issue From a Comment



Please review the new documentation

Draft for 0 hours (due in 7 days)

Details

Participant	Role	Time Spent	Comments	Latest Comment
Nick Pellow	Moderator			
Rosie Jameson	Author	2 mins		
Michael Studman	Reviewer - 0% complete			
Total		2 mins	0	

Files: 1

Linked Issue: CLOV-595

Objectives Edit

A Crucible code review for documentation

General Comments

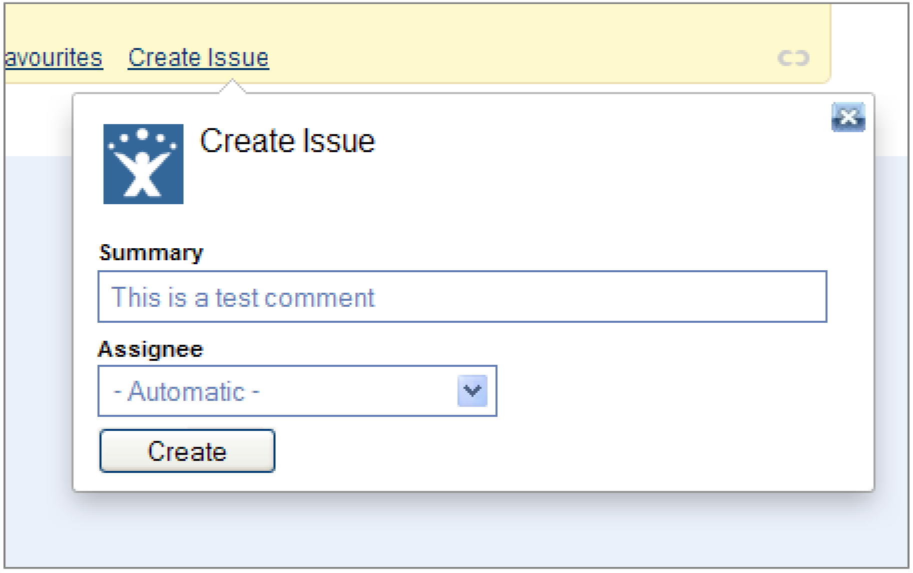
Rosie Jameson says: 19:37

This is a test comment

4. Click to create a JIRA Sub-Task

Clicking the '**Create Issue**' link will allow you to create a JIRA sub-task under the parent JIRA issue, e.g.:

Screenshot: The JIRA 'Create Issue' dialog



Summary

This is a test comment

Assignee

- Automatic -

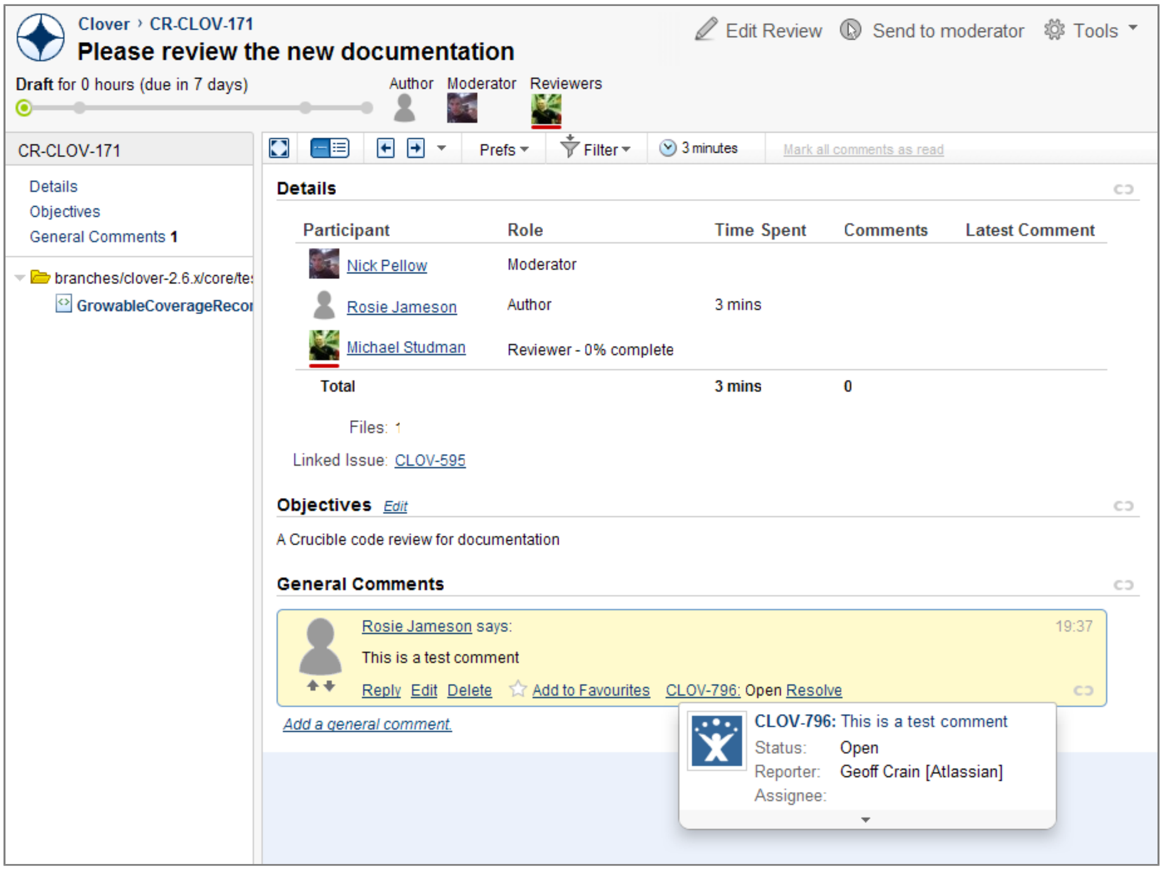
ⓘ The list of possible assignees will include:

- 'Automatic' (i.e. the default assignee for that JIRA project)
- the assignee of the subtask's parent issue
- the reporter of the subtask's parent issue
- 'Unassigned' (if your JIRA administrator has enabled **Allow Unassigned**)
- plus, if Trusted Applications have been configured between JIRA and Crucible,
  - the review author

- the review moderator
- the comment/defect author
- yourself

Once created, the sub-task JIRA issue key, status and default action (i.e. '**Resolve**') will be shown. If you hover your mouse over the JIRA issue key, an information window will show more information and controls relating to that JIRA issue.

Screenshot: The JIRA Hover information window



Please review the new documentation

Draft for 0 hours (due in 7 days)

Details

Participant	Role	Time Spent	Comments	Latest Comment
Nick Pellow	Moderator			
Rosie Jameson	Author	3 mins		
Michael Studman	Reviewer - 0% complete			
Total		3 mins	0	

Files: 1

Linked Issue: CLOV-595

Objectives Edit

A Crucible code review for documentation

General Comments

Rosie Jameson says: 19:37

This is a test comment

CLOV-796: Open Resolve

CLOV-796: This is a test comment

Status: Open

Reporter: Geoff Crain [Atlassian]

Assignee:

ⓘ Users are mapped to their own accounts when using Trusted Applications. If you don't have the permissions to carry out the default action ('**Resolve**', in this case), an error will occur.

5. Resolve the JIRA issue through Crucible

Once the work required on your sub-task is completed, simply click the action link provided to signal that this has occurred (e.g. '**Resolve**'). The JIRA issue will be closed.

> ⓘ If you encounter problems or have trouble using JIRA integration, please read the FAQ page on this topic.

Summarising and Closing the Review

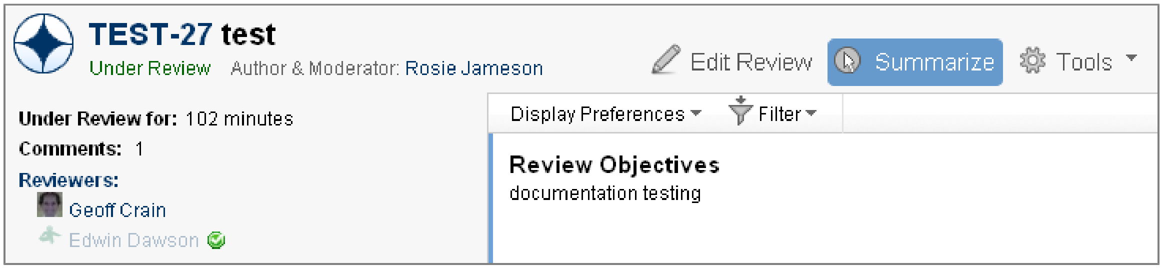
As the moderator, you can choose to **summarize a review** at any time.

ⓘ Normally, we recommend that you wait for all reviewers to complete their reviews.
The reviews that the reviewers have completed will be in your '**To Summarize**' menu on the Dashboard.

To summarize a review,

- Click the '**Summarize**' button at the right of the screen.
- Optionally enter a summary of the review.
- If you have no further comments to add, click the '**Close Review**' button; otherwise, click '**Continue Without Closing**'.

Screenshot: 'Summarize' button



TEST-27 test

Under Review Author & Moderator: Rosie Jameson

Under Review for: 102 minutes

Comments: 1

Reviewers:

Geoff Crain

Edwin Dawson ✓

Review Objectives

documentation testing

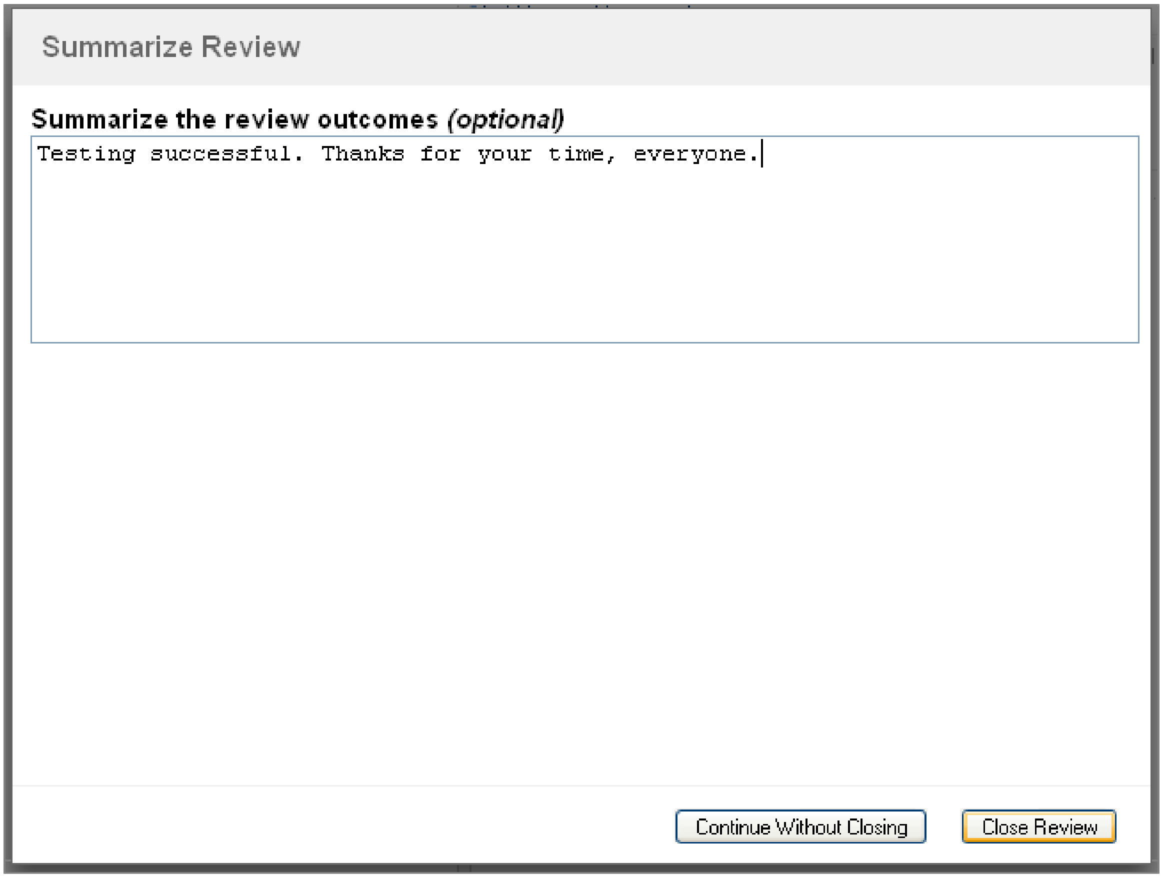
✓ **The above review is not yet complete**
We can see that Geoff Crain has still not finished reviewing, because there is no green tick next to his name.

On clicking 'Summarize', the moderator may be prompted to confirm the action if there are incomplete reviews or draft comments in the review.

⚠ **The requests for confirmation are warnings only**
The review can still be summarized and closed.

Once the review is in the 'Summarize' state, the moderator can optionally add a review summary, i.e. describe the outcomes/tasks/etc.

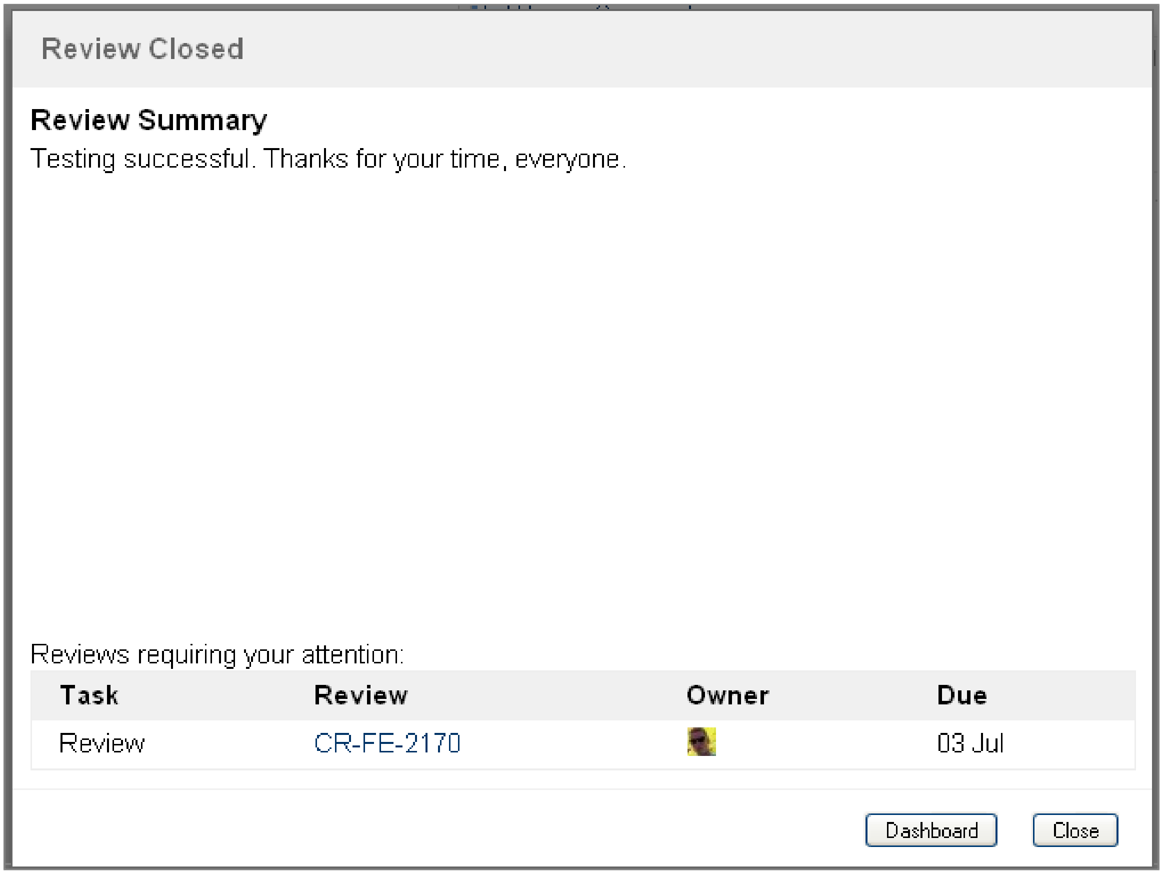
Screenshot: Summarize Review



Summarize the review outcomes *(optional)*

```
Testing successful. Thanks for your time, everyone.
```

Screenshot: Review Closed



Review Summary

Testing successful. Thanks for your time, everyone.

Reviews requiring your attention:

Task	Review	Owner	Due
Review	CR-FE-2170		03 Jul

The summary is sent to all participants and displayed at the top of the closed review.

- The moderator is the only participant who can add comments in 'Summarize' state. This gives the moderator the responsibility of the 'last word'.
- Reviews in the 'Summarize' state can be closed.
- Reviews in the 'Summarize' or 'Closed' state can be re-opened. Re-opening changes the review's state back to 'Under Review', allowing all participants to add comments.

> ⚠ Re-opening a review is not the recommended way to 're-review'. You should create a new review with the reworked changes and link it to its parent review (create a hyperlink back to the original review in the new Review's Objectives field).

Note that you need the '**Summarize**', '**Close**' or '**Re-Open**' permission to summarize, close or re-open a review.

Deleting an Abandoned Review

You can delete reviews that have been abandoned. To do this, follow the instructions below.

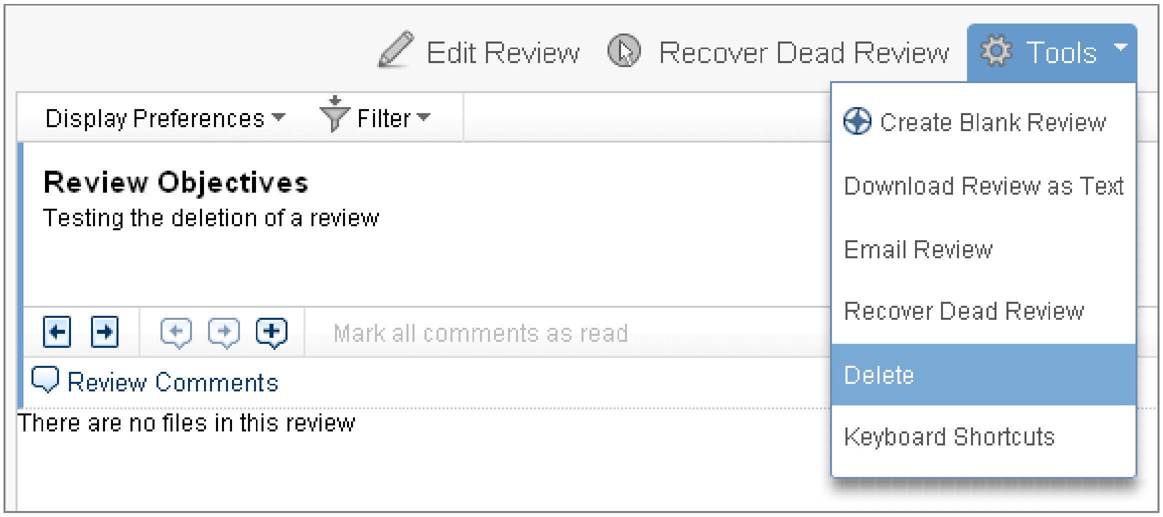
> ⚠ Deleted reviews cannot be retrieved.

Deleting Reviews from the 'Abandoned' list

To delete a review from the 'Abandoned' list;

1. From the '**Crucible Dashboard**', click '**My Abandoned Reviews**' in the left-hand navigation bar.
2. A list of abandoned reviews appears. Click the name of the review you wish to remove.
3. Once the review details are displaying, select '**Delete**' from the '**Tools**' menu. The review will be instantly deleted.

Screenshot: Deleting a Review in Crucible



Review Objectives
Testing the deletion of a review

There are no files in this review

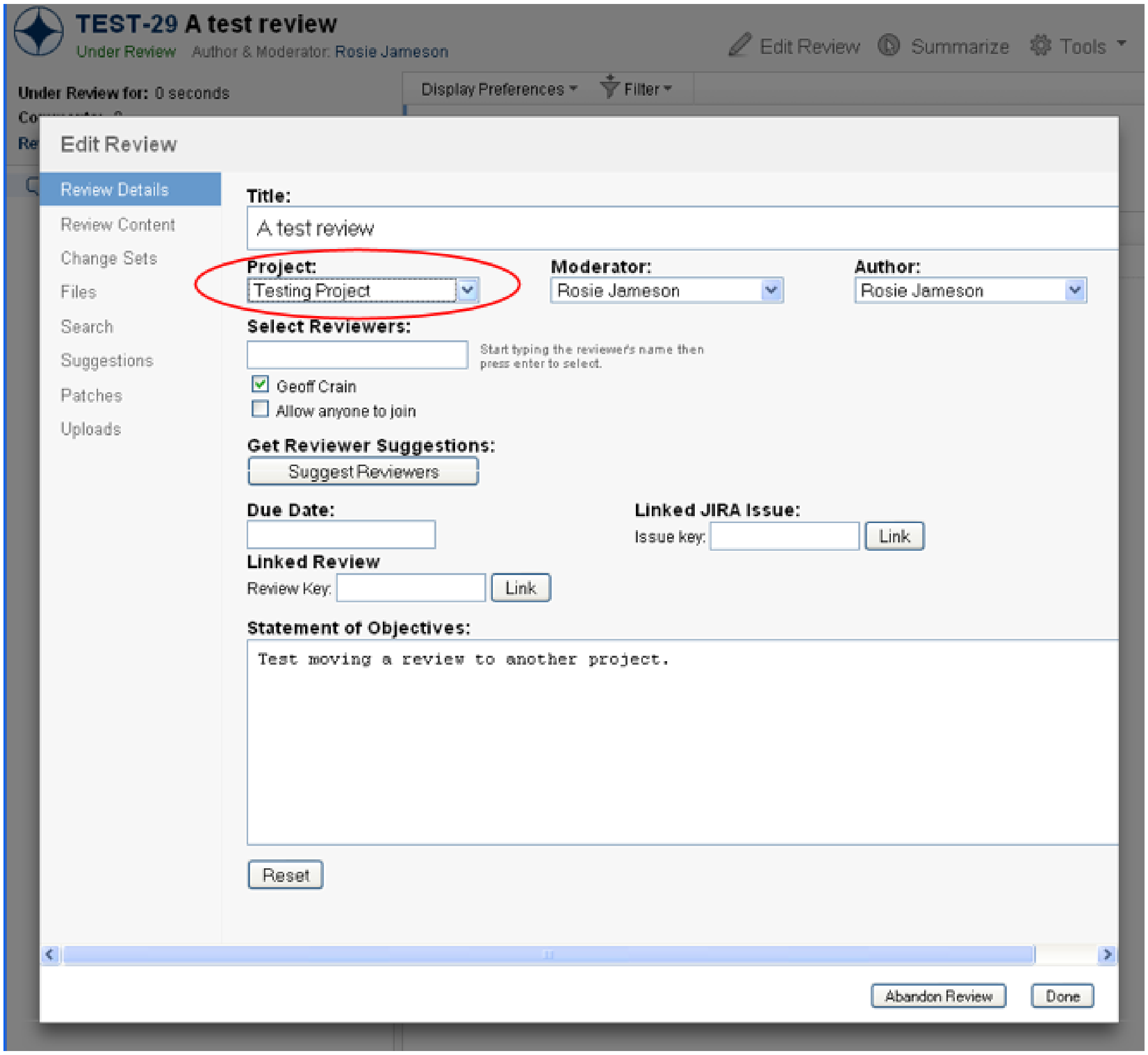
Moving a Review to Another Project

You can move reviews between projects once they have been created.

To move a review between projects,

1. Open the review. Click the '**Edit Review**' button at the top of the screen.
2. The '**Edit Review**' window will open, allowing you to change various aspects of the review.
3. Under '**Project**' click the drop-down menu. This will allow you to select a new parent project for the review.
4. Click the '**Done**' button at the bottom of the screen.

Screenshot: Changing a Review's Parent Project



Title: A test review

Project: Testing Project

Moderator: Rosie Jameson

Author: Rosie Jameson

☑ Geoff Crain

□ Allow anyone to join

Statement of Objectives:

```
Test moving a review to another project.
```

Using the Dashboard

This page contains information on browsing reviews in Crucible.

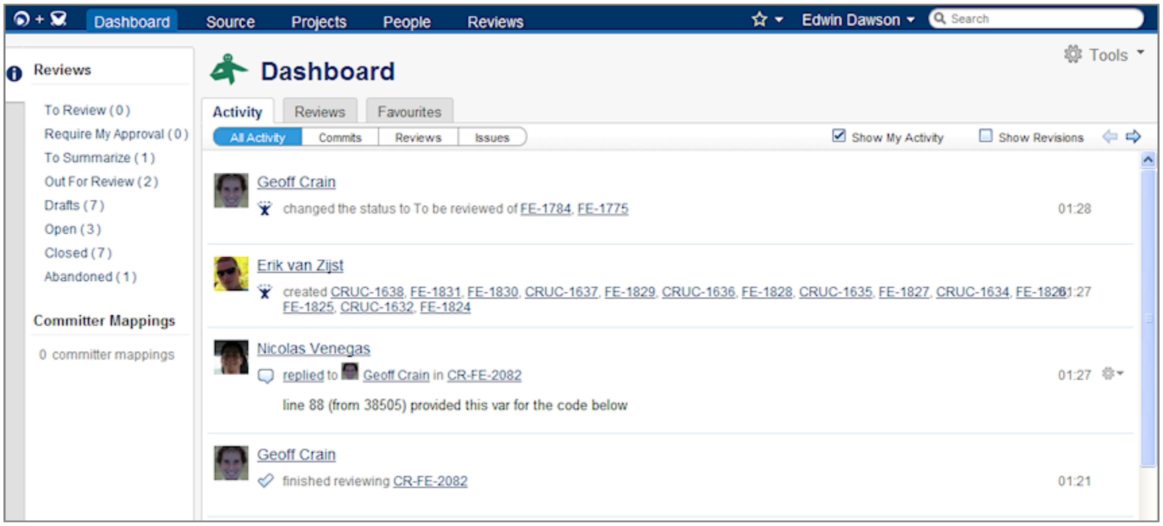
On this page:

- Dashboard Screen Overview
  - Filtering Your View

Dashboard Screen Overview

To browse all your reviews (reviews you are participating in), use the **Dashboard** tab at the top of the page. The Dashboard Screen opens, as shown in the screenshot below.

Screenshot: The Dashboard Screen in Crucible



Geoff Crain
changed the status to To be reviewed of FE-1784, FE-1775 01:28

Erik van Zijst
created CRUC-1638, FE-1831, FE-1830, CRUC-1637, FE-1829, CRUC-1636, FE-1828, CRUC-1635, FE-1827, CRUC-1634, FE-1826 01:27
FE-1825, CRUC-1632, FE-1824

Nicolas Venegas
replied to Geoff Crain in CR-FE-2082 01:27
line 88 (from 38505) provided this var for the code below

Geoff Crain
finished reviewing CR-FE-2082 01:21

By default, the Activity Stream is shown. This is a mix of all activity that is occurring related to Crucible, such as people making review comments, reviews opening and closing, files being committed to a linked repository, or updates to linked JIRA issues.

Filtering Your View

To filter your view, use the constraint options in the sub-nav and the side panel.

Activity Stream filters

You can also filter the items shown in the Activity Stream. To do this, click one of the options in second layer of tabs, as listed in the table below.

Tab Name	Sub-Nav Options
Activity Tab	Sub-Nav Options: • **All Activity** — Shows all activity with no filtering. • **Commits** — Shows commits. • **Reviews** — Shows reviews. • **Issues** — Shows JIRA issues. • **Show/Hide My Activity** — Show/hide your own activity. • **Show/Hide Revisions** — Show/hide activity from other users. • Earlier / Later Activity (arrow buttons) — Pages through activity items, stepping backward or forward through the results.
Reviews Tab (Shows the result of the filter clicked in the left side panel)	Sub-Nav Options: **RSS** — Opens a page with the RSS feed for the current selection.

Favourites Tab (Shows all items you have marked as a 'favourite'.)	Sub-Nav Options: None.

Side Panel

The left navigation panel of the Dashboard shows the number of reviews in different states. Click on any of these states to show the list of reviews in the left-hand panel.

To Review	Reviews where the user still needs to complete their work.
Require My Approval	The user has been assigned the role of moderator for these reviews and needs to approve them.
To Summarize	The user has been assigned the role of moderator for these reviews and needs to summarise and close them.
Out For Review	Reviews created by the user that are currently in progress.
Drafts	These are reviews created by the user that have not yet been moved to the 'Approval' or the 'Require Approval' states.
Open	All open reviews that the user is participating in.
Closed	These are reviews that the user has been involved in and are now closed.
Abandoned	Reviews that are no longer relevant and can be deleted.

Using the Project Dashboard

To browse the content in a project, click the **Projects** tab at the top of the page. The '**Projects**' view opens.

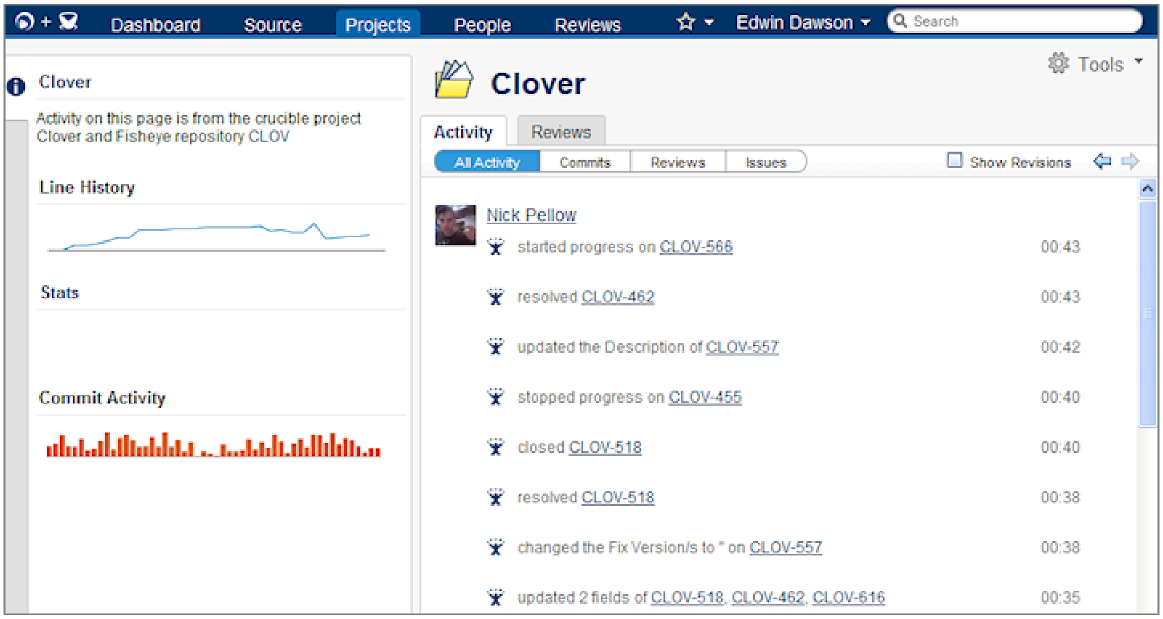
A list of projects will be shown if there is more than one. Click the name of the desired project to open it. The '**Project Activity**' page opens. In the left navigation bar, charts showing overall project statistics are displayed.

There are a number of sub-tabs on this page, listed in the table below.

Sub-Tab Name	Description
Activity	• All Activity — The default view. • Commits — Shows commits in the project (visible when using FishEye). • Reviews — Shows reviews in the project. • Issues — Shows JIRA issues related to this project. ⓘ Only visible if you have set up JIRA Integration in Crucible. • Show Revisions — Shows or hides revisions in the project (visible when using FishEye). • Earlier Activity (Left Arrow icon) — Loads a page of earlier project activity. • Later Activity (Right Arrow icon) — Loads a page of later project activity.
Reviews	Shows recent reviews in the project.

ⓘ The Projects tab is only visible in Crucible. Read more about the definition of a project.

Screenshot: The Crucible Projects View



Clover

Activity on this page is from the crucible project Clover and Fisheye repository CLOV

Nick Pellow

started progress on CLOV-566 00:43

resolved CLOV-462 00:43

updated the Description of CLOV-557 00:42

stopped progress on CLOV-455 00:40

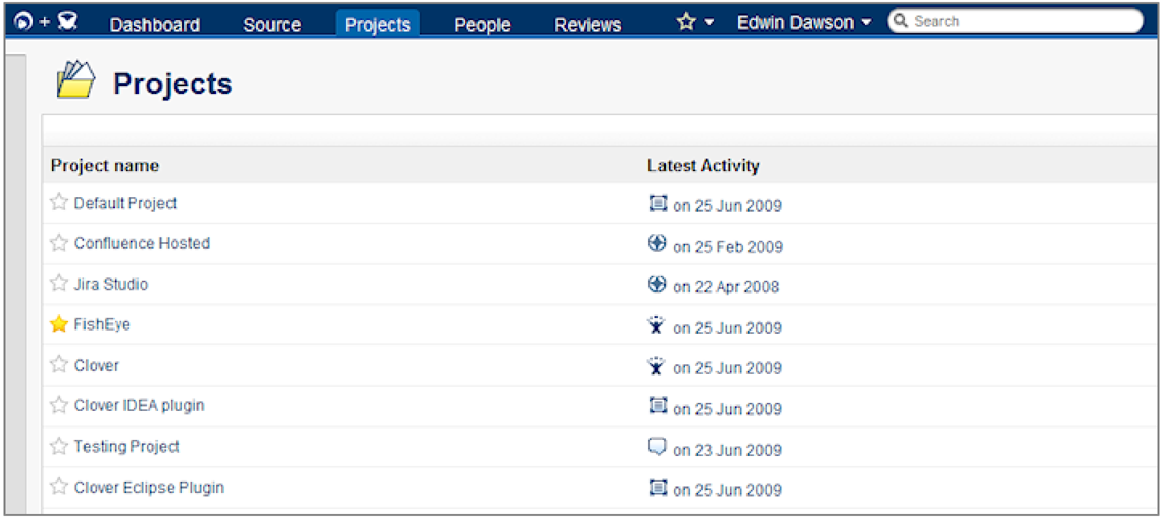
closed CLOV-518 00:40

resolved CLOV-518 00:38

changed the Fix Version/s to '' on CLOV-557 00:38

updated 2 fields of CLOV-518, CLOV-462, CLOV-616 00:35

Screenshot: The Crucible Projects Index



Projects

Project name	Latest Activity
☆ Default Project	on 25 Jun 2009
☆ Confluence Hosted	on 25 Feb 2009
☆ Jira Studio	on 22 Apr 2008
★ FishEye	on 25 Jun 2009
☆ Clover	on 25 Jun 2009
☆ Clover IDEA plugin	on 25 Jun 2009
☆ Testing Project	on 23 Jun 2009
☆ Clover Eclipse Plugin	on 25 Jun 2009

Defining your Workflow

This document describes several forms of Crucible Workflow in detail. Depending on the size of your team, there are four different ways that a development team could use Crucible for code reviews. Choose the workflow which suits your team.

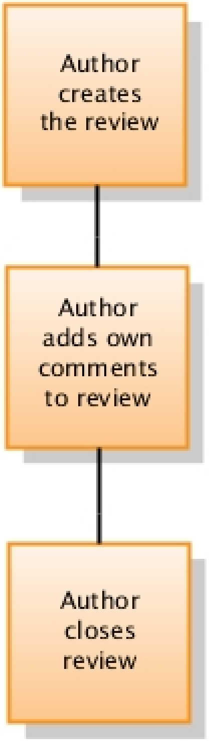
- Lightweight Code Commenting with Crucible (individual)
- One-to-One Reviews (Agile Pair)
- One-to-Many Reviews Without a Moderator (Agile Team)
- Formal Group Reviews (CMM Team)

Lightweight Code Commenting with Crucible (individual)

1. Author commits new work.

2. Author creates the review, and adds comments using the easy web interface.
3. Author summarizes and closes the review, saving the code comments in Crucible's database, which is stored outside the repository.

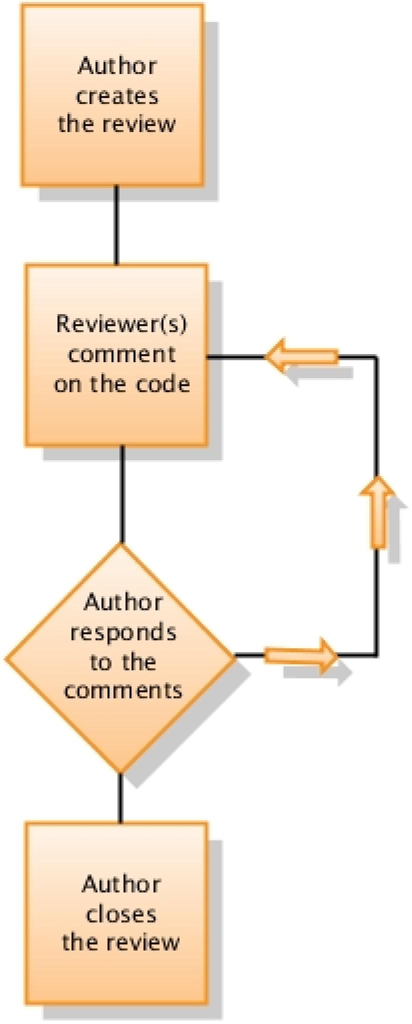
Diagram: Workflow for Lightweight Code Commenting



One-to-One Reviews (Agile Pair)

1. Author creates the review.
2. Author invites reviewer to take part in the review.
3. Reviewer creates comments on the code.
4. Author responds to reviewer comments.
5. Follow-up comments are made if necessary.
6. Reviewer finishes own review process.
7. Author summarizes and closes the review.

Diagram: Workflow for One-to-One Reviews

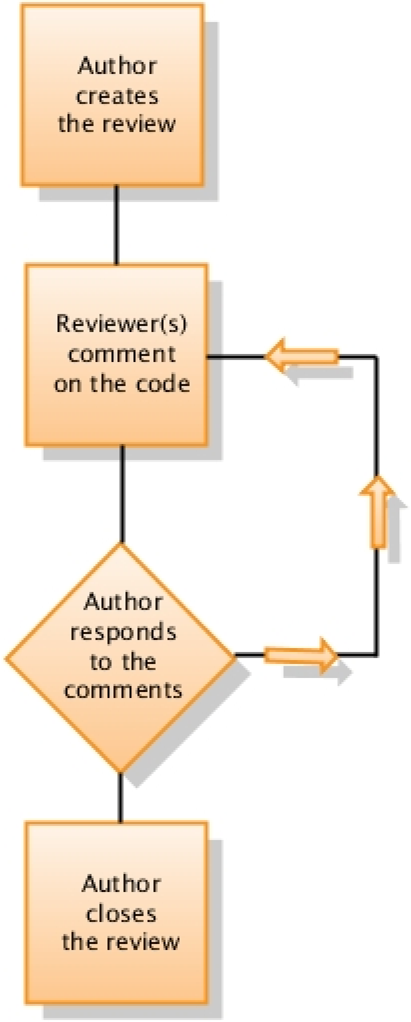

Author creates the review
Reviewer(s) comment on the code
Author responds to the comments
Author closes the review

For more information on one-to-one reviews, see Getting Started with Crucible. The workflow process in Crucible is covered in detail within this document.

One-to-Many Reviews Without a Moderator (Agile Team)

1. Author creates the review.
2. Author invites reviewers to take part in the review.
3. Reviewers make comments on the code.
4. Author responds to reviewer comments, follow-up comments are made if necessary.
5. Reviewers complete their reviews.
6. Author summarizes and closes the review.

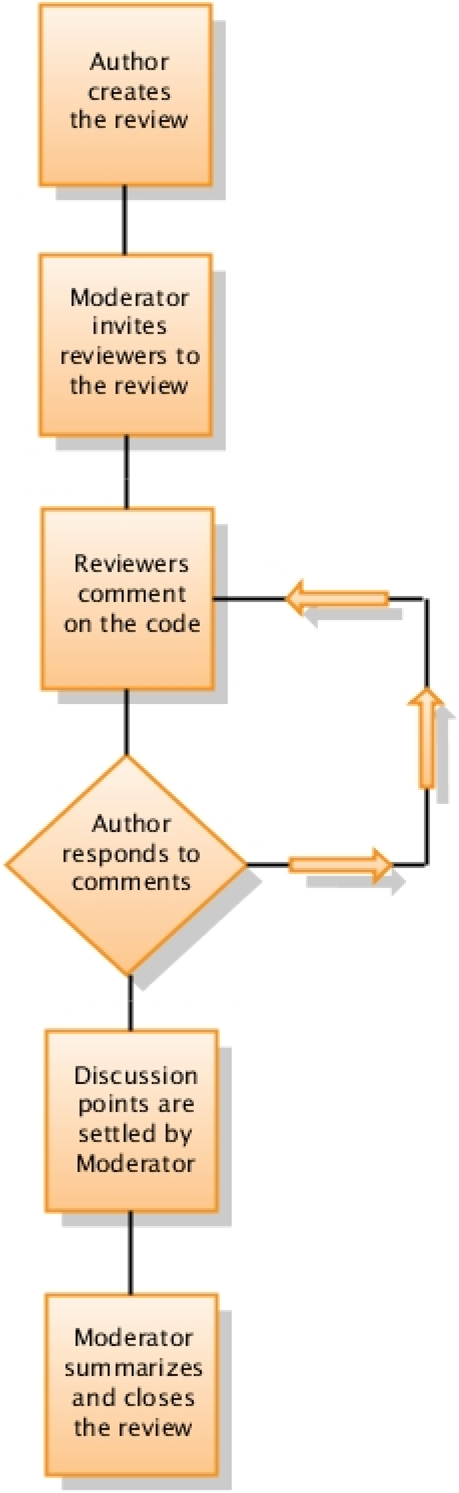
Diagram: Workflow for One-to-Many Reviews


Author creates the review
Reviewer(s) comment on the code
Author responds to the comments
Author closes the review

Formal Group Reviews (CMM Team)

1. Author creates the review.
2. Moderator invites reviewers to take part the review.
3. Reviewers make comments on the code.
4. Author responds to reviewer comments.
5. Follow-up comments are made if necessary.
6. Each discussion point is settled by the Moderator.
7. Moderator summarizes and closes the review.

Diagram: Workflow for Formal Group Reviews


Author creates the review
Moderator invites reviewers to the review
Reviewers comment on the code
Author responds to comments
Discussion points are settled by Moderator
Moderator summarizes and closes the review

To see a simple example of how to use Crucible with two people, see Getting Started with Crucible.

Using Favourites

This page contains instructions on using the '**Favourites**' feature in Crucible to select, view and manage items of interest.

On this page:

Favourites Overview

Crucible allows you to tag certain items as your favourites. You can select code reviews, changesets, files, people and repositories to be added to your favourites. Once your favourites list is created, you can view it or see a stream of all activity relating to your favourite items. We suggest you select items that you are currently working on as favourites, to create a more relevant personalised view.

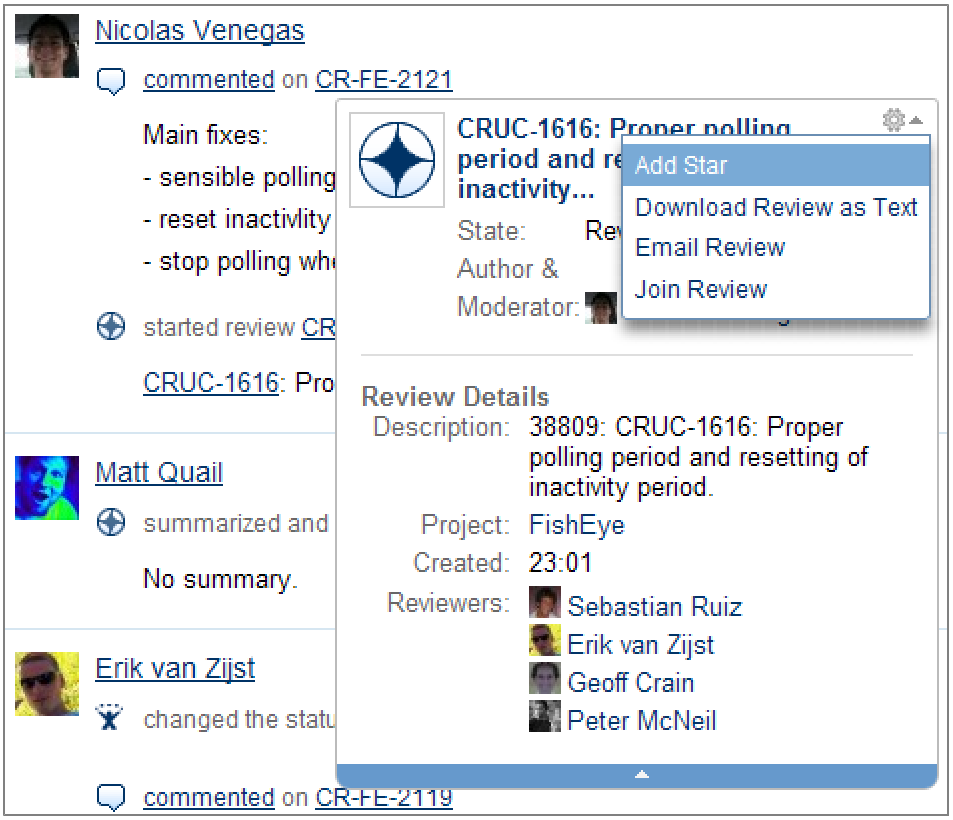
Adding Items to Your Favourites

To add an item to your favourites, follow one of the steps below.

Adding a Review to Your Favourites

To add a review to your favourites, hold the mouse cursor over the review name when it appears in a menu screen. The Review Hover menu appears. At the top right of the Review Hover menu, click the small grey cog icon that indicates the '**Tools**' menu. The Tools menu opens. In the Tools menu, click '**Add Star**'. This will add it to your favourites.

Screenshot: Adding a Review To Your Favourites



Nicolas Venegas

commented on CR-FE-2121

Main fixes:
- sensible polling
- reset inactivlity
- stop polling wh

started review CR

CRUC-1616: Pro

Matt Quail

summarized and

No summary.

Erik van Zijst

changed the statu

commented on CR-FE-2119

CRUC-1616: Proper polling period and re inactivity...

State: Re

Author & Moderator:

Review Details

Description: 38809: CRUC-1616: Proper polling period and resetting of inactivity period.

Project: FishEye

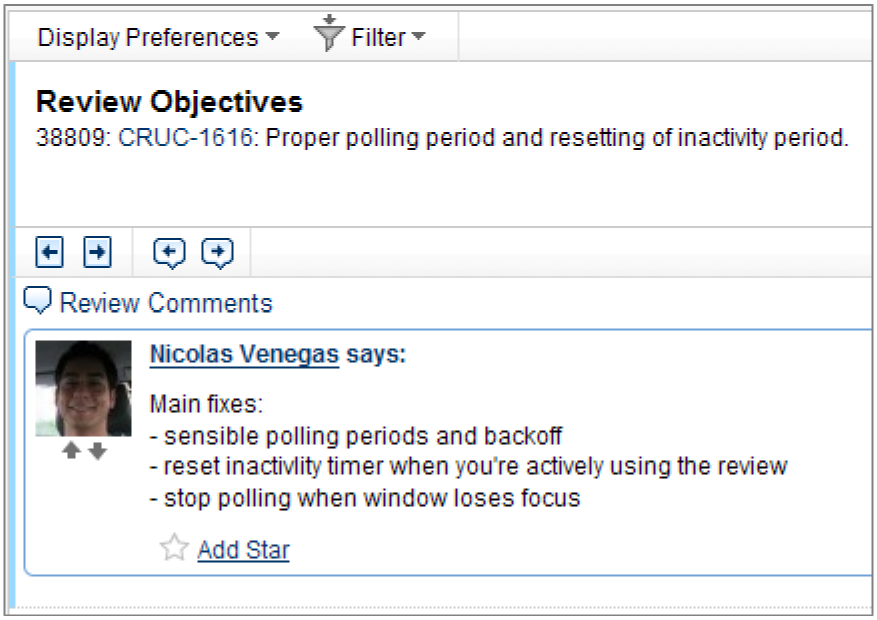
Created: 23:01

Reviewers: Sebastian Ruiz
Erik van Zijst
Geoff Crain
Peter McNeil

Adding a Review Comment Thread to Your Favourites

To add a review comment thread to your favourites, click the link '**Add Star**' next to the grey star icon at the bottom of the first comment of the thread. From then on, new comments will be shown in your favourites activity stream.

Screenshot: Adding a Review Comment Thread to Your Favourites



Review Objectives

38809: CRUC-1616: Proper polling period and resetting of inactivity period.

Review Comments

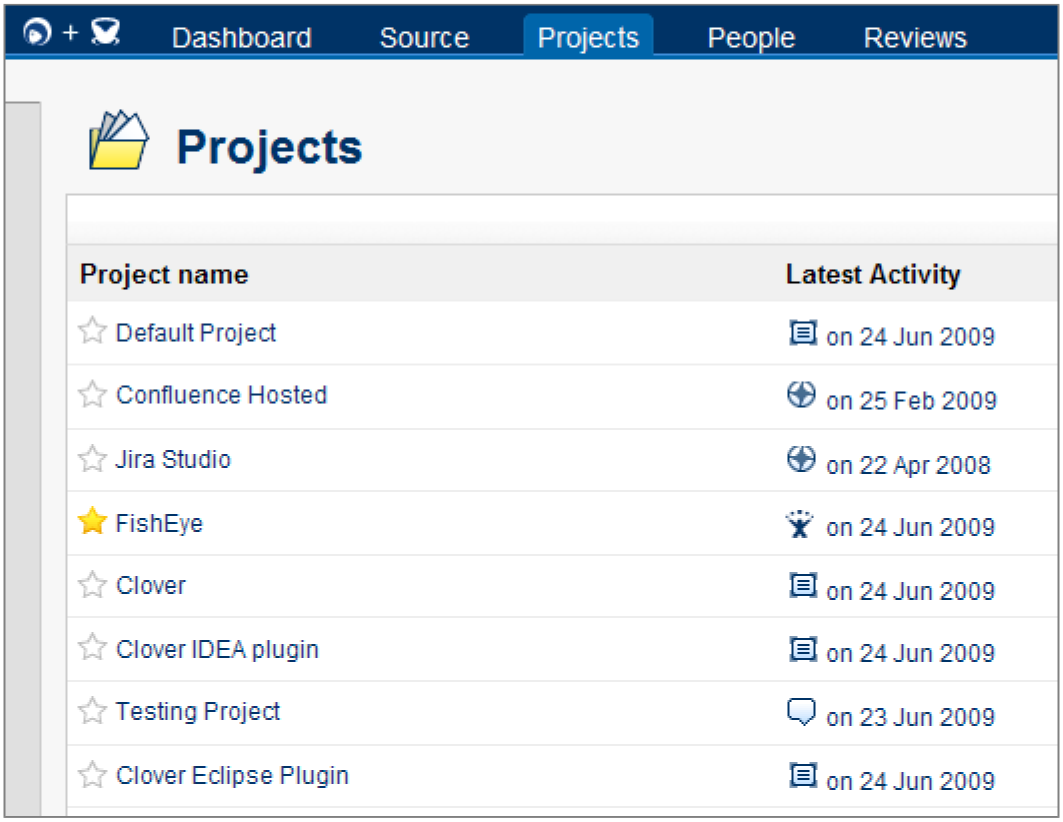
Nicolas Venegas says:

Main fixes:
- sensible polling periods and backoff
- reset inactivlity timer when you're actively using the review
- stop polling when window loses focus

Adding a Project to Your Favourites

To add a project to your favourites, click the '**Projects**' tab. The Projects view appears. Here, simply click the grey star icon that appears next to the desired project name. The star icon will turn yellow, showing that it is selected as a favourite.

Screenshot: Adding a Project to your Favourites



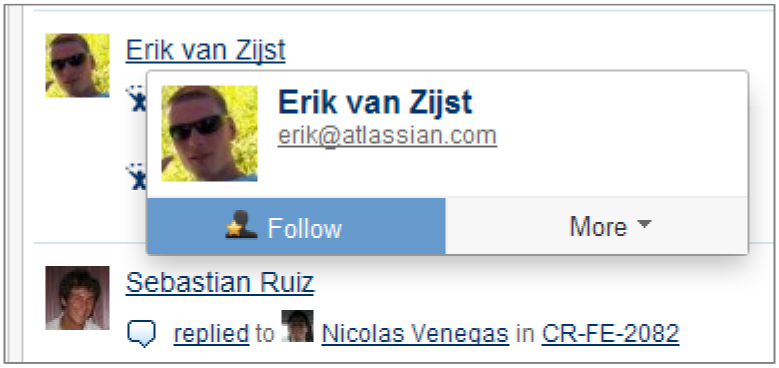
Projects

Project name	Latest Activity
☆ Default Project	on 24 Jun 2009
☆ Confluence Hosted	on 25 Feb 2009
☆ Jira Studio	on 22 Apr 2008
★ FishEye	on 24 Jun 2009
☆ Clover	on 24 Jun 2009
☆ Clover IDEA plugin	on 24 Jun 2009
☆ Testing Project	on 23 Jun 2009
☆ Clover Eclipse Plugin	on 24 Jun 2009

Adding a Person to Your Favourites

To add a person to your favourites, simply hold the mouse cursor over their username wherever it appears. The User Hover menu will appear. In the User Hover menu, click '**Follow**'. This will add the person to your favourites.

Screenshot: Adding a Person to Your Favourites



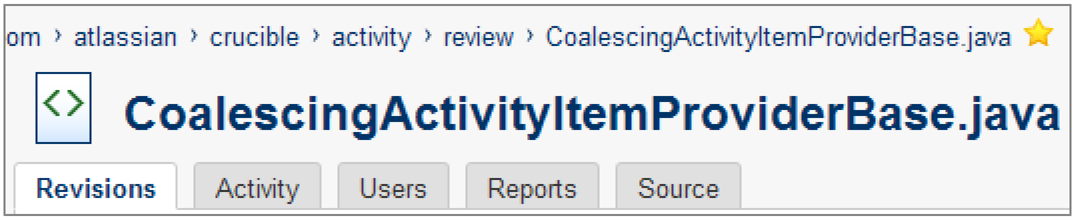
Erik van Zijst
erik@atlassian.com

Sebastian Ruiz
replied to Nicolas Venegas in CR-FE-2082

Adding a Changeset to Your Favourites

To add a changeset to your favourites, firstly open the changeset desired from the '**Source**' tab. Once the changeset is open in Crucible, simply click the grey star icon that appears next to its name. The name appears in the breadcrumb links at the top of the screen.

Screenshot: Adding a Changeset to Your Favourites

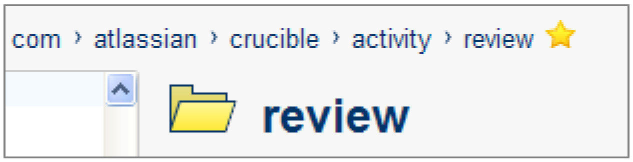

om › atlassian › crucible › activity › review › CoalescingActivityItemProviderBase.java ☆

CoalescingActivityItemProviderBase.java

Adding a File or Folder to Your Favourites

To add a file to your favourites, firstly open the file or folder desired, from the '**Source**' tab. Once the file or folder is open in Crucible, simply click the grey star icon that appears next to its name. The name appears in the breadcrumb links at the top of the screen.

Screenshot: Adding a File or Folder to Your Favourites

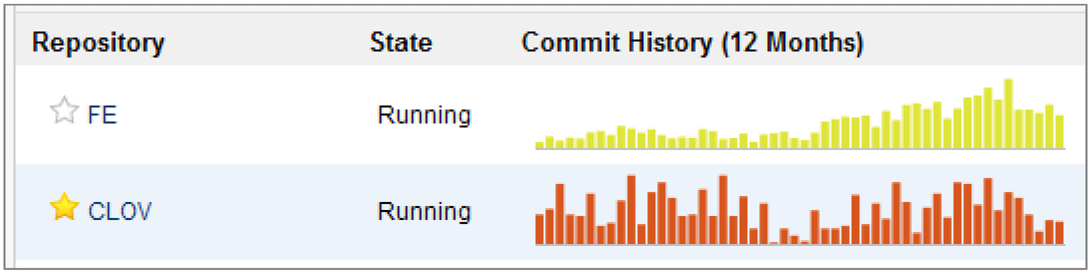

com › atlassian › crucible › activity › review ☆

review

Adding a Repository to Your Favourites

Adding a repository to your favourites (requires FishEye), click the '**Source**' tab. The the '**Source**' view appears. Here, simply click the grey star icon that appears next to the name of the desired repository. The star icon will turn yellow, showing that it is selected.

Screenshot: Adding a Repository to Your Favourites

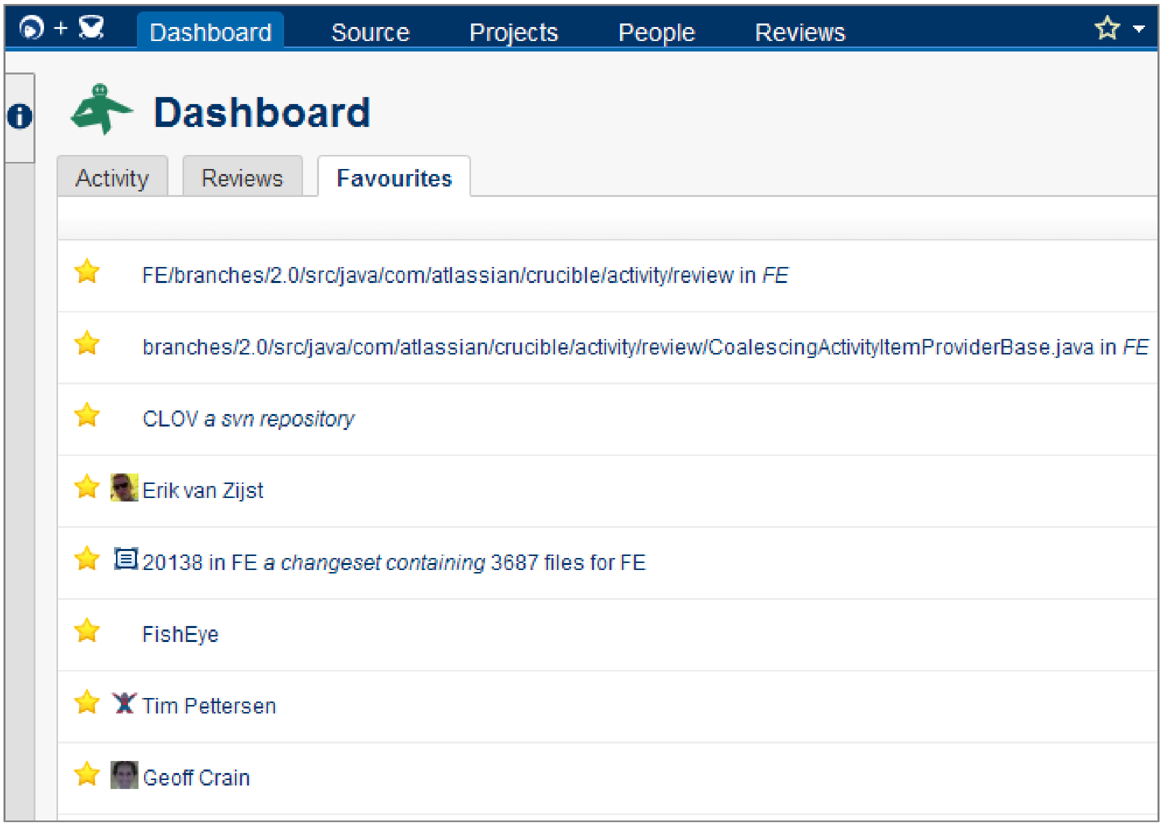


Repository	State	Commit History (12 Months)
☆ FE	Running	
★ CLOV	Running	

Viewing Your Favourite Items

To view your favourite items, click '**Dashboard**' tab at the top left of the page and then the '**Favourites**' second level tab, beneath that.

Screenshot: Viewing Your Favourites



FE/branches/2.0/src/java/com/atlassian/crucible/activity/review in *FE*

branches/2.0/src/java/com/atlassian/crucible/activity/review/CoalescingActivityItemProviderBase.java in *FE*

CLOV *a svn repository*

Erik van Zijst

20138 in FE *a changeset containing* 3687 files for FE

FishEye

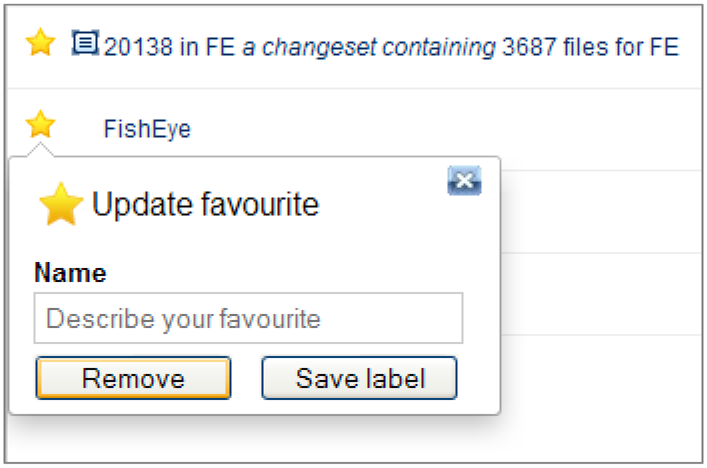
Tim Pettersen

Geoff Crain

Renaming an Item In Your Favourites

To rename an item in your favourites, open the Favourites drop-down menu (the gold star icon located at the top centre of the Crucible screen, next to your user menu). Select the option called '**Manage favourites**'. The Dashboard favourites page opens, showing all of your favourites in the system. To rename any item (changing its favourite display name only — not the name of item itself), simply click the yellow star to the left of its name. A small pop-up menu will appear, the '**Update Favourites**' menu. Type the desired name into the '**Name**' field and click the '**Save label**' button. The label will be updated for the favourites view.

Screenshot: Renaming an Item in Your Favourites



20138 in FE *a changeset containing* 3687 files for FE

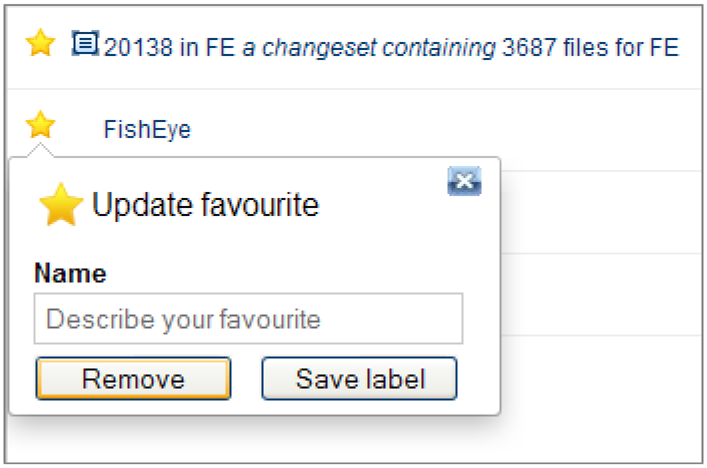
FishEye

Removing an Item From Your Favourites

To remove an item from your favourites, open the Favourites drop-down menu (the gold star icon located at the top centre of the Crucible screen, next to your user menu). Select the option called '**Manage favourites**'. The

Dashboard favourites page opens, showing all of your favourites in the system. To remove any item, simply click the yellow star to the left of its name. A small pop-up menu will appear, the '**Update Favourites**' menu. Click the '**Remove**' button. The star will turn grey, showing that it has been removed from your favourites.

Screenshot: Removing an Item From Your Favourites



20138 in FE *a changeset containing* 3687 files for FE

FishEye

Using Keyboard Shortcuts in Crucible

Crucible provides a number of keyboard shortcuts, allowing you to quickly carry out certain actions without the mouse. Keyboard shortcuts are available for most of the commonly-used functions in Crucible.

To see a list of available shortcuts, firstly navigate to a review in Crucible. Now open the '**Tools**' drop-down menu at the top right corner of the screen, and select the '**Keyboard Shortcuts**' option.

See the tables below for full details:

General Shortcuts

Key	Function
?	Opens reference list of keyboard shortcuts.
escape	Closes reference list of keyboard shortcuts.
alt	Hold down then click and drag to select source line contents.
shift + f	Toggle full screen review mode.

Comment Navigation Shortcuts

Key	Function
n	Go to next review comment.
p	Go to previous review comment.
shift + p	Go to first review comment.
shift + n	Go to last review comment.
l	Go to next thread (skips replies).

h	Go to previous thread (skips replies).
]	Go to next unread comment.
[	Go to previous unread comment.
r	Reply to a comment.
m	Toggle comment read/unread status.

File Navigation Shortcuts

Key	Function
j	Go to next file.
k	Go to previous file.
shift + k	Go to first file.
shift + j	Go to last file.
u	Go to next unreviewed file.
i	Go to previous unreviewed file.
y	Set file reviewed and go to next unreviewed file.
shift + y	Toggle file reviewed/unreviewed status.
e	Expand current file.
c	Collapse current file.
shift + e	Expand all files.
shift + c	Collapse all files.

Using RSS Feeds in Crucible

Subscribing to an RSS Feed

In Crucible, all pages with an activity stream and any page which has a list of reviews will have an RSS option.

To access the RSS feed for a page, open the '**Tools**' drop-down menu at the top right corner of the screen, then click the '**RSS**' option.

This will open a page with the RSS feed displayed; you can also paste the URL from that page into your RSS reader of choice.

Using Wiki Markup in Crucible

Crucible supports Wiki Markup text formatting in comments and review descriptions.

The text markup notation on this page is a reference showing the available formatting commands.

ℹ When using FishEye, you can also render Wiki Markup in commit messages.

Headings

Notation	Description
h1.Biggest heading	Turns text into a heading at size 1. **Biggest Text**
h2.Bigger heading	Turns text into a heading at size 2. **Bigger heading**
h3.Big heading	Turns text into a heading at size 3. **Big heading**
h4.Normal heading	Turns text into a heading at size 4. ***Normal heading***
h5.Small heading	Turns text into a heading at size 5. **Small heading**
h6.Smallest heading	Turns text into a heading at size 6. ***Smallest heading***

Text Effects

Text effects are used to change the formatting of words and sentences.

Notation	Description
bold	Makes text appear **bold**.
italic	Makes text appear in *italics*.
+underline+	Makes text appear underlined.
??citation??	Makes text appear in *citation* form.
-strikethrough-	Makes text appear ~~struck through~~.
^superscript^	Makes text appear in $^{\text{superscript}}$.
~subscript~	Makes text appear in $_{\text{subscript}}$.

`{{monospaced}}`	Placing double curly-brackets around text makes it appear `monospaced`.
bq. Block Quote	To make an entire paragraph into a block quotation, place "bq. " before it. Example: *Some block quoted text*
{quote} here is quoteable content to be quoted {quote}	Quote a block of text that's longer than one paragraph. Example: *here is quoteable* *content to be quoted*

{color:red}
look ma, red text!
{color}
	Changes the color of a block of text.
	Example: look ma, red text!

Text Breaks

Wiki Markup allows you to insert breaks or different kinds of hyphens and dashes.

Notation	Description
(empty line)	Produces a new paragraph
`\\`	Creates a line break. Not often needed, most of the time the wiki renderer will guess new lines for you appropriately.
⎯⎯⎯⎯	Creates a horizontal ruler.
—	Produces em dash — symbol.
–	Produces en dash – symbol.

Links

Creating links is easy with Wiki Markup.

Notation	Description
[Crucible Issue CR-FE-100|CR-FE-100]	Creates a link to a Crucible review or FishEye artifact using the internal key reference for the item.

[#anchor] Link Examples: Link to an anchor on this page: anchor link to top. [Atlassian Crucible|http://atlassian.com]	Creates a link to an external resource, special characters that come after the URL and are not part of it must be separated with a space. External links are denoted with an arrow icon. Examples: • http://www.atlassian.com/crucible • Atlassian Crucible Note: The square brackets [,], around external links are optional in the case you do not want to use any alternate text for the link (i.e. just display the raw URL).
[mailto:mail@example.com]	Creates a link to an email address. Example: mail@example.com
[file:///c:/temp/foo.txt] [file:///z:/file/on/network/share.txt]	Creates a download link to a file on your computer or on a network share that you have mapped to a drive. To access the file, you must right click on the link and choose "Save Target As".
	Creates a bookmark anchor inside the page. You can then create links directly to that anchor. So a link like this: `[My Page#here]` will link to wherever in "My Page" there is an {anchor:here} macro, and the link [#there] will link to wherever in the current page there is an {anchor:there} macro.

Lists

Lists allow you to present information as a series of ordered items. Use asterisks * for bulleted lists and hash symbols # for numbered lists.

Notation	Description
* A bulleted list * Second item ** indented item 1 ** indented item 2 # A numbered list # Second item ## indented item 1 ## indented item 2	Examples: • A bulleted list • Second item • indented item 1 • indented item 2 1. A numbered list 2. Second item a. indented item 1 b. indented item 2

Images

Images can be referenced from remote sources only.

Notation	Description
!http://www.host.com/image.gif!	The image will be displayed from the remote source.

!http://www.host.com/image.gif|align=right, vspace=4!	For any image, you can also specify attributes of the image tag as a comma separated list of name=value pairs as shown in this example.

Tables

Tables allow you to organise content in a rows and columns, with a header row if required.

Notation	Description
||heading 1||heading 2||heading 3|| |col A1|col A2|col A3| |col B1|col B2|col B3|	Makes a table. Use double bars for a table heading row.

The code above produces a table that looks like this:

heading 1	heading 2	heading 3
col A1	col A2	col A3
col B1	col B2	col B3

Advanced Formatting

This section covers panels, code windows and showing plain text with no formatting.

Notation	Description

{noformat}	Makes a preformatted block of text with no syntax highlighting. All the optional parameters of the {noformat} macro are valid for the {panel} macro as well.
	Example:
	`This is a no-formatted piece of text,`
	`so *no* _formatting_ is done here.`

{panel}	Embraces a block of text within a fully customizable panel. The optional parameters you can define are as follows. - title: Title of the panel - borderStyle: The style of the border this panel uses (solid, dashed and other valid CSS border styles) - borderColor: The color of the border this panel uses - borderWidth: The width of the border this panel uses - bgColor: The background color of this panel - titleBGColor: The background color of the title section of this panel Examples: Some text in a basic panel **My Title** Some text with a title **My Title** a block of text surrounded with a **panel** yet *another* line

{code}code goes here{code}
 	
{code:title=Bar.java|borderStyle=solid}
// Some comments here
public String getFoo()
{
return foo;
}
{code}
 	
{code:xml}
<test>
<another tag="attribute"/>
</test>
{code}
	The code macro displays a preformatted block for showing code with syntax highlighting. All the optional parameters of the {panel} macro are valid for {code}. The default language is Java but you can specify JavaScript, ActionScript, XML or SQL.
	
	Examples:
	
	Java with a title bar:
	Bar.java
	`// Some comments here`
	`public String getFoo()`
	`{`
	`    return foo;`
	`}`
	
	A basic display with XML code:
	`<test>`
	`  <another tag="attribute"/>`
	`</test>`

Miscellaneous Markup Features

Emoticons and often-used images can be easily embedded with the following Wiki Markup Syntax:

Notation	Description
\X	Escape special character X (i.e. {)
:), :(	Graphical emoticons (smileys): 🙂, ☹ .

Notation	:)	:(	:P	:D	;)	(y)	(n)	(i)	(/)	(x)	(!)
Image	🙂	☹	😛	😁	😉	👍	👎	ℹ	✅	❎	⚠

Notation	(+)	(-)	(?)	(on)	(off)	(*)	(*r)	(*g)	(*b)	(*y)
Image	➕	⛔	❓	💡	💡	⭐	★	★	★	⭐

Crucible Administrator's Guide

Once you have installed and configured Crucible, you can access the **Administration** pages at the following address (where 'HOSTNAME' is the name of the server where you installed Crucible).

```
http://HOSTNAME:8060/admin/
```

The '**Admin Menu**' allows you to administer your Crucible/FishEye instance and manage your repositories.

For more information on administering FishEye, please refer to the FishEye documentation.

Information in the *Crucible Administrator's Guide*:

- Backing Up and Restoring Crucible Data
- Creating a Permission Scheme
- Creating a Project
- Crucible and FishEye
- Customising Email Notifications
- Customising the Defect Classifications
- Customising the Welcome Message
- Deleting a Project
- Editing a Project
- JIRA Integration in Crucible
- Migrating to an External Database
- Trusted Applications

Backing Up and Restoring Crucible Data

Crucible data can be backed up from the admin interface or command line. This page contains the command syntax, options and the required procedure to backup and restore your Crucible instance.

On this page:

- Backing Up Crucible Data

Backing Up Crucible Data

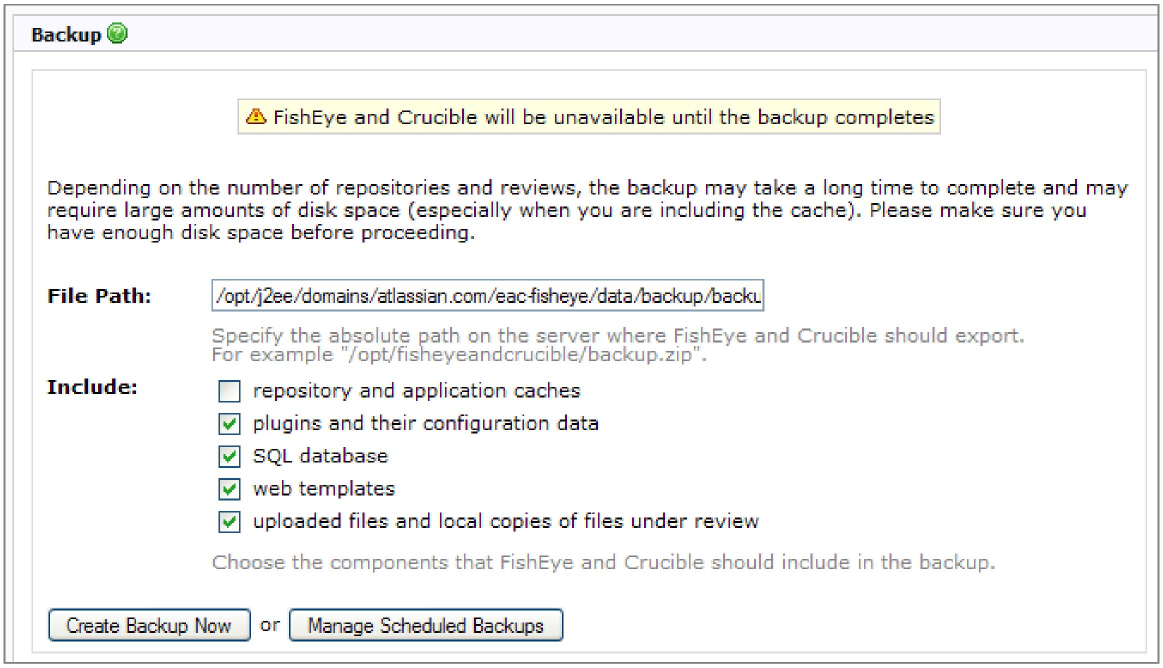
The Crucible Admin Interface Backup Process

1. Navigate to the Crucible '**Admin**' screen (click the '**Administration**' link in the footer of any Crucible page).
2. On the Admin screen, click '**Backup**' under the '**System**' heading in the left navigation bar. The Backup screen opens.
3. On the Backup screen, the '**File Path**' field indicates where the backup file (in .zip format) will be stored. You can manually edit this path to change it. Under the heading '**Include'**, a list of checkboxes is shown, with the following items:

- Plugins and their configuration data
- SQL database
- Web templates
- Uploaded files and local copies of files under review.
- Repository and application caches.
  ℹ Repository and application caches contain temporary data stored from repository scans and library caches that improve startup time. Both will be recreated automatically by re-scanning the source repositories, so the backup files can be reduced by a significant amount by excluding these (if the cost of re-scanning is acceptable).

4. Once you have chosen your options, click '**Create Backup Now**'.

Screenshot: The Crucible Backup Screen



Backup

⚠ FishEye and Crucible will be unavailable until the backup completes

Depending on the number of repositories and reviews, the backup may take a long time to complete and may require large amounts of disk space (especially when you are including the cache). Please make sure you have enough disk space before proceeding.

File Path: /opt/j2ee/domains/atlassian.com/eac-fisheye/data/backup/backu

Specify the absolute path on the server where FishEye and Crucible should export. For example "/opt/fisheyeandcrucible/backup.zip".

Include:

- ☐ repository and application caches
- ☑ plugins and their configuration data
- ☑ SQL database
- ☑ web templates
- ☑ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backup.

The Crucible Command Line Backup Process

Your Crucible instance must be running during the backup.

1. Open a command line interface on the Crucible server computer.
2. Navigate to the `FISHEYE_HOME/bin/` directory.
3. Run the backup command on the command line with the desired options.
4. The backup is created as a new Zip archive file and placed in the `FISHEYE_INST/backup/` directory.
   Note that if your Crucible instance uses a custom `FISHEYE_INST` directory, make sure the environment variable is properly set when running the backup command.

Components of a Crucible Backup

The Crucible backup is highly configurable and allows for many different configurations. This table shows the various components of the backup, what they are for and how they can be used.

Component	Purpose	Defaults
SQL Database	Refers to the SQL content database (used by both FishEye and Crucible and containing all user profile data, reviews and their comments).	Backed up by default.
Cache	The cache contains data that reflects the state of FishEye's repositories. Without it, FishEye must re-scan its repositories after a backup is restored. The cache also contains OSGI library data that increases startup time. These too can be excluded and will be generated automatically when the application is started.	The cache is not backed up by default as it tends to be large (running a risk of pushing the maximum file size for Java backups), whilst also representing replaceable data.
Plugins	Plugins are 3rd-party extensions that you may have installed, and configuration for all plugins (this includes configuration for Crucible's set of standard plugins).	Configuration data for all plugins are backed up by default, as well as all plugins installed in `FISHEYE_INST/var/plugins/user`.

Templates	In this context, these are custom freemarker templates that you or your users have created. They live in `FISHEYE_INST/template`.	Templates are backed up by default. You can choose to exclude them from the backup if your templates directory is covered by some other backup mechanism.
Uploads	In this context, uploads refers to files which are added to Crucible via the web interface (such as patch file reviews). It also includes each repository-backed file that went under review, when Crucible is configured to make a local copy of every reviewed file.	Uploads are backed up by default. You can choose not to back them up for example when the `FISHEYE_INST/var/data/uploads` directory is already covered by some other backup mechanism.

Note that the backup will always include the configuration data (`config.xml`), your license file and the FishEye user data.

Backup Command Line Options

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `FISHEYE_HOME/bin/` directory.

The basic syntax of the backup command is as follows:

```
$ ./fisheyectl.sh backup [OPTIONS]
```

To see inline help for all backup options, run the following command in the `FISHEYE_HOME/bin/` directory:

```
$ ./fisheyectl.sh backup --help
```

|| Option || Switch || Description || Default setting ||

Quiet mode	-q OR --quiet	Suppresses output	No
Output filename	-f OR --file	Specify a different path and filename to the `FISHEYE_INST/backup/backup_YYYY-DD-MM_HHmm.zip` file. When filename is omitted, the backup filename contains the date and time.	`FISHEYE_I` is the defaul
Compression level	--compression OR -c	Sets the Zip compression level, from 1-9. Runs at level 6 if no argument is passed.	Yes (6)

Anonymise	-a OR --anonymise	Anonymises the SQL database by replacing all text with 'x'. This is only useful when sending a backup to Atlassian as part of a support case. **Please do not anonymise data unless the Support Engineer handling your support case has specifically requested the data anonymised (as often anonymised data will not help reproduce the issue).**	No
Cache Backup	--cache	Include the repository caching files in the backup. These hold information gained from scanning the repositories and can be quite large (many gigabytes). However, it can shorten the time needed to re-scan the repositories after data is restored.	No. By defau data is exclu backups.

Command Line Examples

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `FISHEYE_HOME/bin/` directory.

Backing up with compression of 9, quiet mode and setting an output location

```
$ ./fisheyectl.sh backup --compression 9 -q
   -f /application_backups/fisheye/20090215
   .zip
```

Backup including cache data (also includes all default components)

```
$ ./fisheyectl.sh backup --cache
```

Restoring a backup with cache data (also restores all default components)

```
$ ./fisheyectl.sh restore --cache
```

Advanced Backup Command Line Settings

In some cases it might be preferable to only backup a limited set of items. This could be useful when your instance uses an external database such as MySQL or PostgreSQL and your DBA has already configured automatic backups in the database. The commands below allow this.

Option	Switch	Description	Default
Exclude Plugins	--no-plugins	Excludes plugins from the backup.	No. By default, plugins are included in every backup.
Exclude Templates	--no-templates	Excludes templates from the backup.	No. By default, templates are included in every backup.
Exclude Uploads	--no-uploads	Excludes uploaded files (such as patch reviews, stored in Crucible's internal database) from the backup.	No. By default, uploads are included in every backup.
Exclude SQL Database	--no-sql	Excludes the SQL content database used by both FishEye and Crucible.	No. By default, this data is included in every backup.
Show help	--help OR -h	Shows inline help on the command line.	No

Known Limitations

Please note that the below limitations are common for any Java based backup tool.

Archives Containing Over 65535 Files
Versions of Java earlier than v1.6 (b25) are incapable of handling zip files that contain more than 65,535 files. The solution for this problem is to either upgrade to a version of Java later than v1.6 (b25), or ensure that the archive does not exceed the threshold (contains less than 65,535 files). The FishEye cache (not included in backups by default) can be a contributor of many small files. Hence, exclude the cache from backups if this is likely to be a concern.

Archives Larger Than 4GB
Java has trouble reading and writing zip files that are larger than 4GB. As of release 1.5 Java appears capable of reliably creating archives that are over 4GB, but remains unable to extract them. For details see Sun's bug report. Also be aware of the fact that some file systems (including FAT32) have trouble with files larger than 4GB.

As a workaround, make sure you do not create archives that are larger than 4GB. The FishEye cache (not included in backups by default) can be a contributor of a lot of small files (although these tend to compress very well). If you still want to archive everything and end up with an achive that is too large, consider creating separate backups for the FishEye cache and uploaded files respectively.

Scheduling Crucible Backups

To set a schedule for automatic backups, open the administration screen and click '**Backup**' under '**System**' on the left navigation bar. The 'Backup' page opens. Now, click the link '**Manage Scheduled Backups**' at the bottom of the page. The 'Scheduled Backups' page opens.

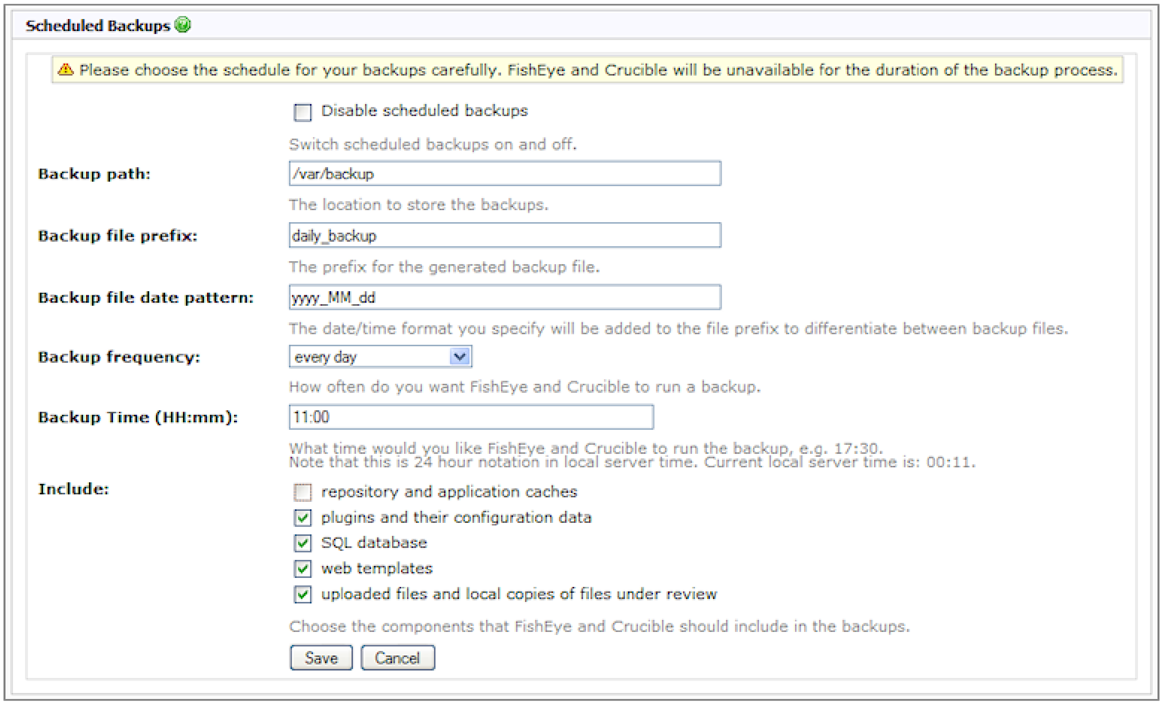
On the 'Scheduled Backups' page, click '**Edit**' to adjust the backup schedule. Set the desired options and click '**Save** '.

The options for scheduled backups are detailed in the table below.

Option name	Description	Allowed Values

Disable Scheduled Backups	Stops regular backups from taking place.	On (disabled) or Off (enabled)
Backup path	The path where the backup .zip file will be stored.	Any system or network path that FishEye or Crucible can access.
Backup file prefix	Characters that will be added to the beginning of the backup file name.	Any string of characters that can be used as part of a filename on the local operating system.
Backup file date pattern	Sets a date for the next (or initial) backup to take place.	Any valid date in the format `yyyy_MM_dd` (year, month, day of the month).
Backup frequency	Sets how often the backup will take place.	Can be set to '**every day**', '**every Sunday**', '**Monday to Friday**' and '**first day of the month**'.
Backup time (HH:mm)	The time when the backup will take place.	Any valid 24-hour time in the format `HH:mm` (hours, minutes).
Include	Specifies which items must be included in the backups (these components are explained at the top of this page).	As per the options for regular on-demand backup (These components are explained at the top of this page).

Screenshot: Scheduling Backups in FishEye and Crucible



Scheduled Backups

⚠ Please choose the schedule for your backups carefully. FishEye and Crucible will be unavailable for the duration of the backup process.

□ Disable scheduled backups

Switch scheduled backups on and off.

Backup path: /var/backup

The location to store the backups.

Backup file prefix: daily_backup

The prefix for the generated backup file.

Backup file date pattern: yyyy_MM_dd

The date/time format you specify will be added to the file prefix to differentiate between backup files.

Backup frequency: every day

How often do you want FishEye and Crucible to run a backup.

Backup Time (HH:mm): 11:00

What time would you like FishEye and Crucible to run the backup, e.g. 17:30.
Note that this is 24 hour notation in local server time. Current local server time is: 00:11.

Include:

- □ repository and application caches
- ☑ plugins and their configuration data
- ☑ SQL database
- ☑ web templates
- ☑ uploaded files and local copies of files under review

Choose the components that FishEye and Crucible should include in the backups.

ℹ Be aware that scheduled backups can fill up disks unless you regularly move or delete old archives.

Restoring Crucible Data

The Crucible Data Command Line Restoration Process

There is currently no way to restore a backup from the web interface because Crucible must be shut down during a data restore.

> Restoring a backup will irreversibly overwrite the data of your installation with the data from the backup archive.
> **If you made a backup from production which connected to an external database, and restore this backup to a test server without specifying another database to restore too, you will drop and restore to your production database**. Thus when restoring to a test server, always ensure you specify the correct database to restore to (or restore to an in-built database).

1. Install Crucible into a new, empty directory (this must be the same version that the backup was created from, or later).
   Note that you cannot restore data into versions of Crucible which are older than the version that created the backup.
2. Make sure the Crucible instance is not running.
3. Open a command line interface on the Crucible server computer.
4. Run the restore command on the command line with the desired options.
5. The specified elements will be restored.
6. Start the Crucible instance.
7. When using FishEye integrated with Crucible, you will need to re-index your repositories after restoring data, unless the backup archive was created with the `--cache` option.

Restore Command Line Options

These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `FISHEYE_HOME/bin/` directory.

The basic syntax of the restore command is as follows:

```
$ ./fisheyectl.sh restore -f
   /path/to/backup_2009-10-02_1138.zip
   [OPTIONS]
```

To see inline help for all backup options, run the following command in the `FISHEYE_HOME/bin/` directory:

```
$ ./fisheyectl.sh restore --help
```

Restores a FishEye/Crucible backup instance.

If you are using an external database (as opposed to the default built-in database), make sure the JDBC driver file is present in the `FISHEYE_INST/lib` directory when running restore.

Option	Switch	Description	Default
Suppress output	--quiet OR -q	Suppress the output messages from the restore program on the command line.	No
Choose file to restore from	--file PATH/FILENAME OR -f PATH/FILENAME	Restore the backup from PATH/FILENAME.	Yes (required)
Show inline help	--help OR -h	Displays help for options on the command line.	No

Advanced Command Line Restore Settings

By default, the restore program will restore all items found in the backup archive (so if you included the caches using the `--cache` option, these will automatically be restored). However, it is possible to only restore a subset of items from the backup, by explicitly specifying the item names on the command line and only those will be restored.

Option	Switch	Description
Restore FishEye cache	--cache	Restore the repository cache backup.
Restore plugins	--plugins	Restore 3rd-party plugins and their configuration data.
Restore templates	--templates	Restore freemarker templates from the backup (the restored instance will use the built-in templates).
Restore uploads	--uploads	Restore uploads (e.g. patch files uploaded into Crucible and contents of files under review).
Restore Crucible reviews	--sql	Restore the SQL database containing user profiles, reviews and review comments.
Set database type	--dbtype OR -t	SQL database type ('mysql', 'postgresql' or 'built-in'). Only required when restoring to a database location different to that used at used at backup time.
Set JDBC URL	--jdbcurl OR -j	JDBC URL of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in').
Set JDBC username	--username OR -u	JDBC username of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in').
JDBC password	--password OR -p	JDBC password of the SQL database. Only required when restoring to a database location different to that used at used at backup time (not applicable for 'built-in').

JDBC class	--driver OR -d	Specifies the JDBC driver class name needed to access the SQL database. Only required when restoring to a database location different to that used at used at backup time and when using a different JDBC driver than the standard driver associated with the database specified through `--dbtype`. (Not applicable for 'built-in'.)

Notes on Migrating Backup Data

When the process restores a SQL database, it looks at the configuration data (`config.xml`) included in the backup archive to learn which database product was used and how to connect to it. When Crucible uses the built-in HSQLDB database (which is the default), the restored instance will also use that.
However, when the restored instance will use a different database than the backed up instance (for instance, HSQLDB was used at the time the backup was created, but it needs to be restored on MySQL), use the command line options to point the process to the new database.

Command Line Example: Migrating Backup Data to MySQL

ℹ These examples are for use in a Linux-like operating system. When using these commands on Windows, use the filename `fisheyectl.bat` and use the correct slashes. Run the command from the `FISHEYE_HOME/bin/` directory.

Restoring to a Crucible instance that uses a different database (ensure the mysql driver jar file is present in the `FISHEYE_INST/lib` directory)

```
$ ./fisheyectl.sh restore \
--username john \
--password smith \
--jdbcurl jdbc:mysql:
  //localhost:3306/crucible \
--dbtype mysql \
--file /path/to/backup_2009-10-02_1138.zip
```

Creating a Permission Scheme

This page contains information on how to create a permission scheme in Crucible.

On this page:

Introduction to Crucible Permissions

A *permission* is the ability to perform a particular action in Crucible, e.g. 'Create Review'.

A *permission scheme* assigns particular permissions to any or all of the following:

- Particular Users.
- Particular Groups.
- All logged-in users.
- Anonymous Users
- People in particular Review Roles, such as:
  - 'Author';
  - 'Reviewer';
  - 'Creator';
  - 'Moderator'.

The scheme's permissions will apply to all reviews belonging to the project(s) with which the scheme is associated.

You can create as many permission schemes as you wish. Each permission scheme can be associated with many projects or just one project, allowing you to tailor appropriate permissions for individual projects as required.

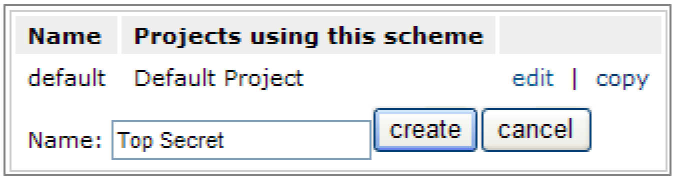
Creating a Permission Scheme

To create a permission scheme,

1. From the '**Admin Menu**', click '**Permission Schemes**'.
2. The '**Permission Schemes**' page will be displayed, showing a list of existing permission schemes. Click the '**Create a New Permission Scheme**' link, which appears below the list.
3. In the '**Name**' field, type a short phrase to uniquely identify your project (see screenshot 1 below).
4. Click the '**Create**' button to create your new permission scheme. The '**Edit Permission Scheme**' page will be displayed for your new permission scheme (see screenshot two, below).
   Your new permission scheme will have the default assignees shown in the table above.
5. To edit the assignees for a permission, click the '**Edit**' link corresponding to the permission. The '**Edit Permission Scheme**' page will be displayed.
6. Choose the appropriate assignee(s) for this permission:
   ⚠ Note: for ongoing ease of management, it is recommended that you grant permissions to groups or participants rather than to individual users.
   - To assign this permission to anonymous users, select the '**Allow Anonymous users**' check-box.
   - To assign this permission to all logged-in users, select the '**Allow All logged in users**' check-box.
   - To assign this permission to a particular user, type their username into the '**Individual users**' field (hint: you can type just part of the name, then press <Enter> to select from a list of matching usernames).
   - To assign this permission to a particular group of users, type the group name into the '**Groups**' field (hint: you can type just part of the group name, then press <Enter> to select from a list of matching groups).
   - To assign this permission to users who belong to a particular participant ('**Reviewer**' / '**Moderator**' / '**Author**' / '**Creator'**), select the corresponding check-box.
7. Click the '**Save**' button.

Next step: see Associating a Permission Scheme with a Project.

Screenshot 1: Adding a Permission Scheme



Name	Projects using this scheme	
default	Default Project	edit | copy

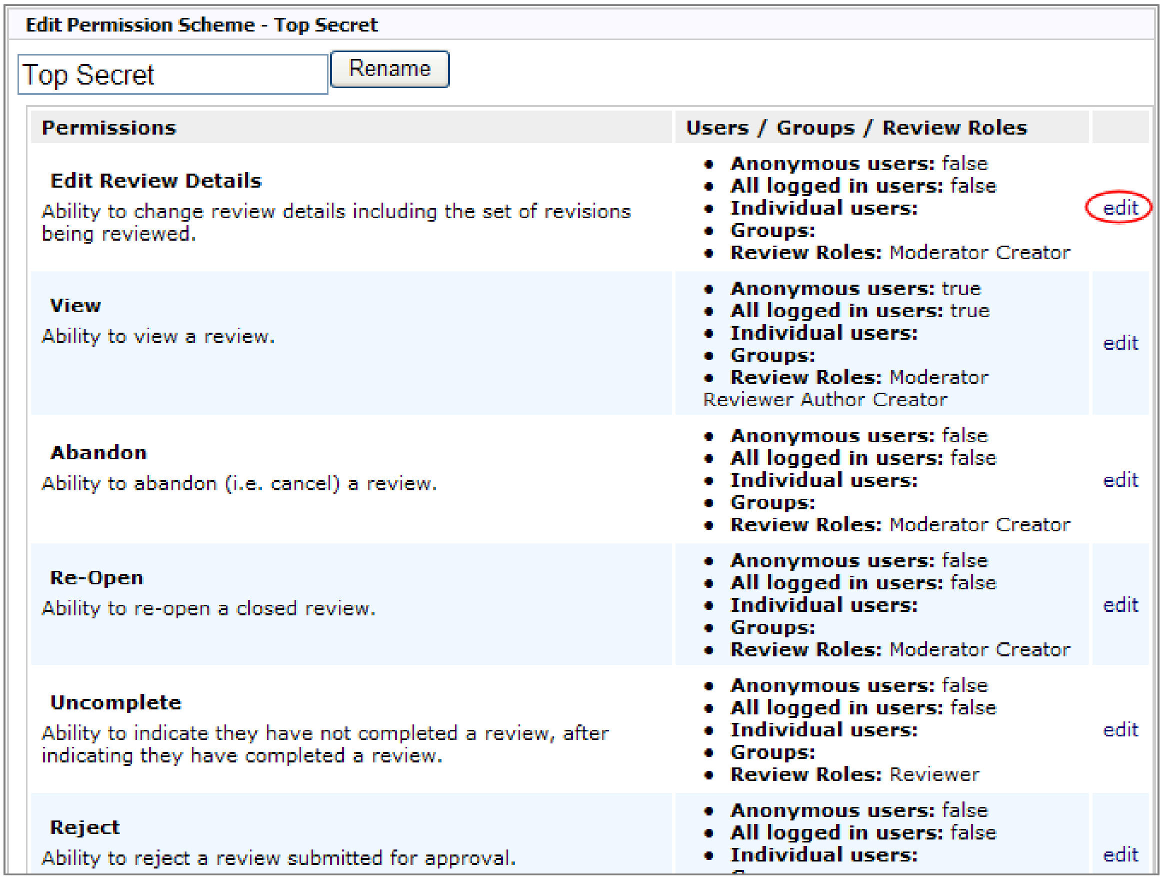
Name: Top Secret

Editing a Permission Scheme

To edit a permission scheme,

1. From the '**Admin Menu**', click '**Permission Schemes**'.
2. Click '**edit**' next to the scheme you wish to change. The '**Edit Permission Scheme**' page will be displayed.
3. On the '**Edit Permission Scheme**' page, you can change the groups or users that are allowed individual permissions by clicking '**edit**' next to the permission in question.
4. When you have finished editing, click the '**Save**' button.

Screenshot: Edit a Permission Scheme

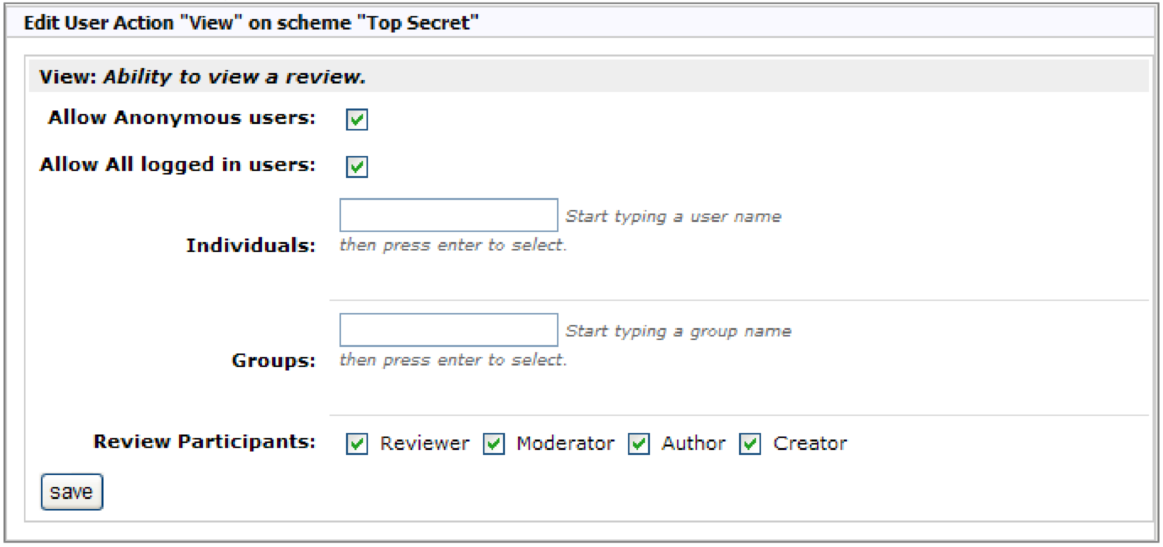


Edit Permission Scheme - Top Secret

Top Secret

Permissions	Users / Groups / Review Roles	
Edit Review Details Ability to change review details including the set of revisions being reviewed.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Review Roles:** Moderator Creator	edit
View Ability to view a review.	**Anonymous users:** true **All logged in users:** true **Individual users:** **Groups:** **Review Roles:** Moderator Reviewer Author Creator	edit
Abandon Ability to abandon (i.e. cancel) a review.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Review Roles:** Moderator Creator	edit
Re-Open Ability to re-open a closed review.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Review Roles:** Moderator Creator	edit
Uncomplete Ability to indicate they have not completed a review, after indicating they have completed a review.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Review Roles:** Reviewer	edit
Reject Ability to reject a review submitted for approval.	**Anonymous users:** false **All logged in users:** false **Individual users:**	edit

Screenshot: Editing the 'View' Permission



Edit User Action "View" on scheme "Top Secret"

View: *Ability to view a review.*

Allow Anonymous users: ☑

Allow All logged in users: ☑

Individuals: *Start typing a user name then press enter to select.*

Groups: *Start typing a group name then press enter to select.*

Review Participants: ☑ Reviewer ☑ Moderator ☑ Author ☑ Creator

List of Crucible Permissions

The following permissions are available:

Permission	Description	Default Assignees
'**Edit**'	Ability to edit a review's details and change the set of revisions being reviewed.	'Creator' 'Moderator'
'**View**'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users 'Creator' 'Author' 'Reviewer' 'Moderator'
'**Abandon**'	Ability to abandon (i.e. cancel) a review.	'Moderator' 'Creator'
'**Re-Open**'	Ability to re-open a closed or abandoned review.	'Creator' 'Moderator'
' **Uncomplete** '	Ability of a reviewer to change their individual review status from 'Complete' to 'Uncomplete'.	'Reviewer'
'**Reject**'	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	'Moderator'
'**Complete**'	Ability of a reviewer to change their individual review status to 'Complete'.	'Reviewer'

'**Comment**'	Ability to add or remove a comment to or from a review.	'Creator' 'Author' 'Reviewer' 'Moderator'
'**Approve**'	Ability to approve a review (i.e. issue it to the reviewers).	'Moderator'
'**Submit**'	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	'Creator' 'Author'
'**Close**'	Ability to close a review once it has been summarised.	'Moderator'
'**Delete**'	Ability to delete a review.	'Creator' 'Moderator'
'**Summarise**'	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	'Moderator'
'**Create**'	Ability to create a review.	All logged-in users
'**Recover**'	Ability to resurrect an abandoned (i.e. cancelled) review.	'Creator' 'Moderator'

Associating a Permission Scheme with a Project

This page explains how to associate a permission scheme with a Crucible project and show details of the default permission schemes included with Crucible.

On this page:

Associating a Permission Scheme with a Crucible Project

To associate a permission scheme with a project,

1. From the '**Admin Menu**', click '**Project List**'.
2. The '**Projects List**' page will be displayed. Find the project you wish to associate with your permission scheme, and click its '**Edit**' link.
3. The '**Edit Project'** page will be displayed.
4. Under the heading '**Project Permissions Scheme**', click the '**Permission Scheme**' drop-down list to select your permission scheme.
   ℹ You will be shown a list of the schemes that have been created in Crucible. You can create a new permission scheme if necessary.
5. Click the '**Save**' button.

Overview of the Permission Schemes Bundled with Crucible

Crucible comes with two permission schemes. '**Default**' and '**Agile**'. The following tables show the default settings in detail; note that these can be easily edited by admin users to suit your needs.

Default Permission Scheme Settings

This table shows the various permissions and which user groups have them by default.

Permission	Anonymous	All Logged In	Individuals	Groups	Review Roles
Abandon	false	false	None	None	Creator, Moderator
Approve	false	false	None	None	Moderator
Close	false	false	None	None	Moderator
Comment	false	false	None	None	Reviewer, Creator, Author, Moderator
Complete	false	false	None	None	Reviewer
Create	false	true	None	None	None
Delete	false	false	None	None	Creator, Moderator
Edit Review Details	false	false	None	None	Creator, Moderator
Recover	false	false	None	None	Creator, Moderator
Reject	false	false	None	None	Moderator
Re-Open	false	false	None	None	Moderator
Submit	false	false	None	None	Creator
Summarize	false	false	None	None	Moderator
Uncomplete	false	false	None	None	Reviewer
View	false	true	None	None	Reviewer, Creator, Author, Moderator

ℹ The default permission scheme has changed since Crucible 1.6.

Agile Permission Scheme Settings

This table shows the various permissions and which user groups have them by default.

Permission	Anonymous	All Logged In	Individuals	Groups	Review Roles
Abandon	false	false	None	None	Creator, Author, Moderator
Approve	false	false	None	None	Creator, Author, Moderator
Close	false	false	None	None	Reviewer, Creator, Author, Moderator
Comment	false	false	None	None	Reviewer, Creator, Author, Moderator
Complete	false	false	None	None	Reviewer
Create	false	true	None	None	None
Delete	false	false	None	None	Creator, Author, Moderator
Edit Review Details	false	false	None	None	Reviewer, Creator, Author, Moderator
Recover	false	false	None	None	Reviewer, Creator, Author, Moderator
Reject	false	false	None	None	Creator, Author, Moderator

Re-Open	false	false	None	None	Reviewer, Creator, Author, Moderator
Submit	false	false	None	None	Creator, Author, Moderator
Summarize	false	false	None	None	Reviewer, Creator, Author, Moderator
Uncomplete	false	false	None	None	Reviewer
View	true	true	None	None	Reviewer, Creator, Author, Moderator

Related Links

- Creating a Permission Scheme

Creating a Project

A Crucible *project* is a collection of reviews, typically reviews that all relate to the same application. In addition to providing a logical way of grouping reviews together, a project allows you to

- define default moderators, authors and reviewers for the reviews in that project.
- define which people are eligible to be reviewers for the reviews in that project.
- use permission schemes to restrict who can perform particular actions (e.g. 'Create Review') in that project.

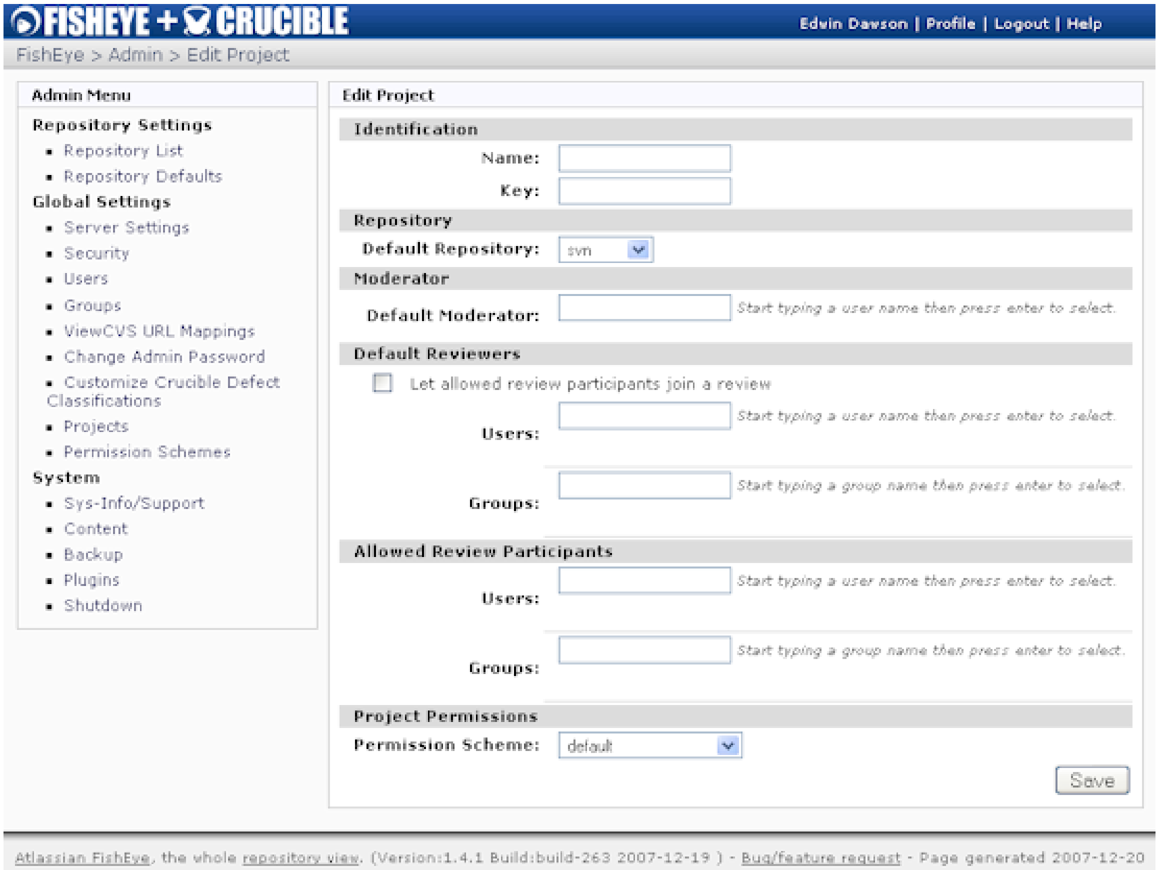
Every Crucible review belongs to a project. Each project has a *name* (e.g. **ACME Development**) and a *key* (e.g. **ACME**). The project key becomes the first part of that project's *review keys*, e.g. **ACME-101**, **ACME-102**, etc:

By default, Crucible contains one project. This default project has the key '**CR**' and the name '**Default Project**'.

To create a new project,

1. From the '**Admin Menu**', click '**Project List**'.
2. The '**Projects List**' page will be displayed. Click the '**Create a New Project**' link, which appears at the bottom of the list of existing projects.
3. The '**Create Project'** page will be displayed.
4. In the '**Name**' field, type a short phrase that describes your project.
5. In the '**Key**' field, type a few characters to uniquely identify your project. This key must consist of alphabetic and/or numeric characters and hyphens only.
6. In the '**Default Repository**' field, select the repository which contains source code relating to this project.
   ℹ This repository is the one that will be searched by default when you add files to a review.
7. In the '**Default Moderator**' field, type the name of the person who will appear by default in the 'Moderator' field when you create a new review; or leave this field blank to force the review's creator to choose a moderator.
8. *(Optional)* Under '**Default Reviewers**', select the people to whom new reviews in this project will be assigned by default:
   - Select the '**Let allowed review participants join a review**' check-box if you wish to determine the default for the '**Allow anyone to join**' option on the 'Adding Reviewers' screen.
   - In the '**Users**' field, type the name(s) of individual users to whom new reviews will be assigned by default.
   - In the '**Groups**' field, type the name(s) of groups to whose members new reviews will be assigned by default.
9. *(Optional)* Under **Allowed Review Participants**', select who will be allowed to have a role (i.e. be an author/creator/moderator/reviewer) in this project's reviews:
   - In the '**Users**' field, type the name(s) of individual users who will be eligible to be authors/creators/moderators/reviewers for reviews in this project*.
   - In the '**Groups**' field, type the name(s) of groups whose members will be eligible to be authors/creators/moderators/reviewers for reviews in this project*.
     * These users will be the only ones whose names appear when a review is assigned.
10. In the '**Permission Scheme**' drop-down list, select the relevant permission scheme for this project. (A permission scheme controls who can perform particular actions, e.g. 'Create Review'.)
11. In the **Review Duration** section, you can enter a value for the number of working days that you want the review to run for. Simply type the number of days into the text entry field marked '**Default duration in week days**' and click '**Save**'.

Screenshot: The Create/Edit Project Screen



Edit Project

Identification

Name:

Key:

Repository

Default Repository: svn

Moderator

Default Moderator: Start typing a user name then press enter to select.

Default Reviewers

□ Let allowed review participants join a review

Users: Start typing a user name then press enter to select.

Groups: Start typing a group name then press enter to select.

Allowed Review Participants

Users: Start typing a user name then press enter to select.

Groups: Start typing a group name then press enter to select.

Project Permissions

Permission Scheme: default

Atlassian FishEye, the whole repository view. (Version:1.4.1 Build:build-263 2007-12-19) - Bug/feature request - Page generated 2007-12-20

Setting Crucible to Store all Revisions

When creating a project or editing a project's properties, you can set Crucible to save all revisions that are associated with a review to Crucible's database. This allows you to be able to view that file content whether or not the repository is online or accessible to Crucible. It also creates an enhanced audit trail should you require it, saving the review content regardless of whether or not it is deleted or lost from the repository.

ℹ Note that the storage of revisions must be set per-project. Also, the storage only applies to reviews created after Revision Storage is enabled. This means that for existing projects, pre-existing reviews will not be stored unless you look at them again after Revision Storage is enabled.

Enabling Revision Storage on a new project

To enable Revision Storing on a new project,

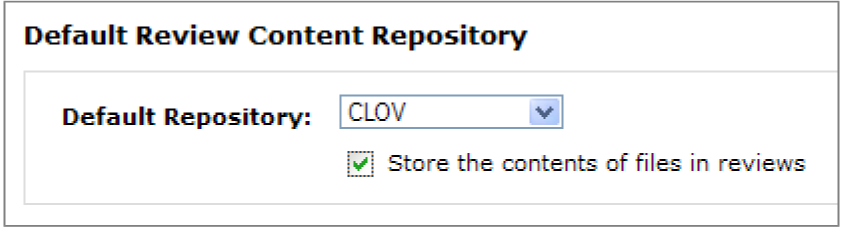
1. When creating a new project, you have the option to turn on revision storing on the '**Create New Project**' page.
2. Under '**Default Content Review Repository**', Click the checkbox labelled '*Store the contents of files in reviews*'.
3. Click '**Save**' to finish.

Enabling Revision Storage on an existing project

To enable Revision Storing on an existing project,

1. From the '**Admin**' screen, click '**Projects**' from the left navigation bar.
2. Click '**Edit**' next to the desired project.
3. Under '**Default Content Review Repository**', Click the checkbox labelled '*Store the contents of files in reviews*'.
4. Click '**Save**' to finish.

Screenshot: Enabling Revision Storage



Default Review Content Repository

Default Repository: CLOV

☑ Store the contents of files in reviews

Crucible and FishEye

This page gives an overview of the joint installation of Crucible and FishEye. Both Crucible and FishEye are Atlassian products.

- **FishEye** allows you to extract information from your source code repository and display it in sophisticated reports.
- **Crucible** allows you to request, perform and manage code reviews.
- Both of these products can run in isolation. However if you are using CVS, Subversion or Perforce you can significantly enhance your Crucible experience by also using FishEye.

> ⚠ **Your Crucible installation package includes the files required for FishEye**
> If you use FishEye and Crucible together, they run as one instance.

Purchasing and Installing Crucible/FishEye

- If you install Crucible, there is no need to do a separate installation of FishEye.
- Upgrading Crucible to also use FishEye requires only a simple licence change in the admin screens.
- When upgrading to Crucible when you have an existing FishEye installation, you can either keep the original FishEye installation or install Crucible and FishEye as a fresh install. Refer to the guide on upgrading from FishEye to Crucible.

FISHEYE_HOME and FISHEYE_INST

Throughout the Crucible documentation, references are made to `FISHEYE_HOME`, which refers to the location of the FishEye application. Because most Crucible users also run FishEye, we use a single value for the sake of simplicity.

Crucible also makes use of this FishEye environment variable:

- `FISHEYE_INST` – the location of the FishEye data.

Refer to the FishEye documentation for more about the environment variables and how they are used in the FishEye installation.

Detailed Documentation

You can find more information in:

- Crucible Installation Guide
- FishEye Installation Guide

Customising Email Notifications

Email notifications in Crucible can be customised to change their formatting, by editing template files. This page contains instructions for this process.

Editing Crucible Email Templates

Template files for Crucible are stored in the `FISHEYE_HOME/templates/` folder.

For Crucible, the set of templates is for plain-text email only. Note that these templates do not support embedding full diffs into notifications. They are only for changing the appearance and order of certain content inside the messages.

> ⚠ If you edit the templates of an operational Crucible instance, you may disrupt notifications that are being sent at that time. To avoid this, shut Crucible down during template editing.

Editing the Subject Line

1. Open the '**crucible-notification-subject.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Type in your new text for the email subject, ensuring that all of the content is contained within line 1 of the template. '**crucible-notification-subject.ftl**' is used as the subject template for all Crucible email notifications.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

Editing the Header

Header information will be included at the beginning of the email body text.

1. Open the '**crucible-notification-header.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Add your new header content. '**crucible-notification-header.ftl**' is used as the header template for all Crucible email notifications.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

Editing the Footer

Footer information will be included at the end of the email body text.

1. Open the '**crucible-notification-footer.ftl**' template file from `FISHEYE_HOME/templates/` in a text editor. '**crucible-notification-footer.ftl**' is used as the footer template for all Crucible email notifications.
2. Add your new footer content.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

After an edit, the change to the email template will take place immediately. No restart is required.

> Try and avoid editing the live template file, as Crucible may try to use it while you are editing. This could have unpredictable results. Instead, back up the template file (it's wise to keep original versions of all these files), edit a copy you have made, then overwrite the 'live' template once you have finished.

Advanced Editing of Crucible Email Templates

The email notification templates use the Freemarker format. Freemarker is a general templating engine enabling automated content.

If you are familiar with Freemarker, more advanced customisations can be made to the email notification templates. However, you make such adjustments at your own risk.

Note: In Crucible, email notifications are limited to plain-text format only.

Crucible Email Template File List

The following template files for Crucible notification are stored in the `FISHEYE_HOME/templates/` folder.

Template filename	Purpose
crucible-notification-subject.ftl	Subject template
crucible-notification-header.ftl	Header template
crucible-notification-footer.ftl	Footer template
state-closed-notification.ftl	State Closed template
all-completed-notification.ftl	All Completed template
state-changed-notification.ftl	State Changed template
completed-notification.ftl	Completed template
general-notification.ftl	General notification template
uncompleted-notification.ftl	'Uncompleted' template
all-no-longer-completed-notification.ftl	All-No-Longer-Completed template
comment-notification.ftl	Comment template
reply-notification.ftl	Reply template
review-precis-plain.ftl	Precis template

See also Customising FishEye Email Notifications.

Freemarker Data Model for Email Templates

Customising Crucible email templates with Freemarker

See the Freemarker documentation for instructions on Freemarker syntax. Use the templates that ship with Crucible as a guide to the properties available on each object.

Specific email types will have extra data associated with them, and this data will be available in that particular template (but not in others).

Example

The syntax to access the data-model, using the data model object '**link**' as an example, place this code into the email at the desired position.

```
${notification.link}
```

Customising the Defect Classifications

This page explains how to customise defects and their classifications in Crucible.

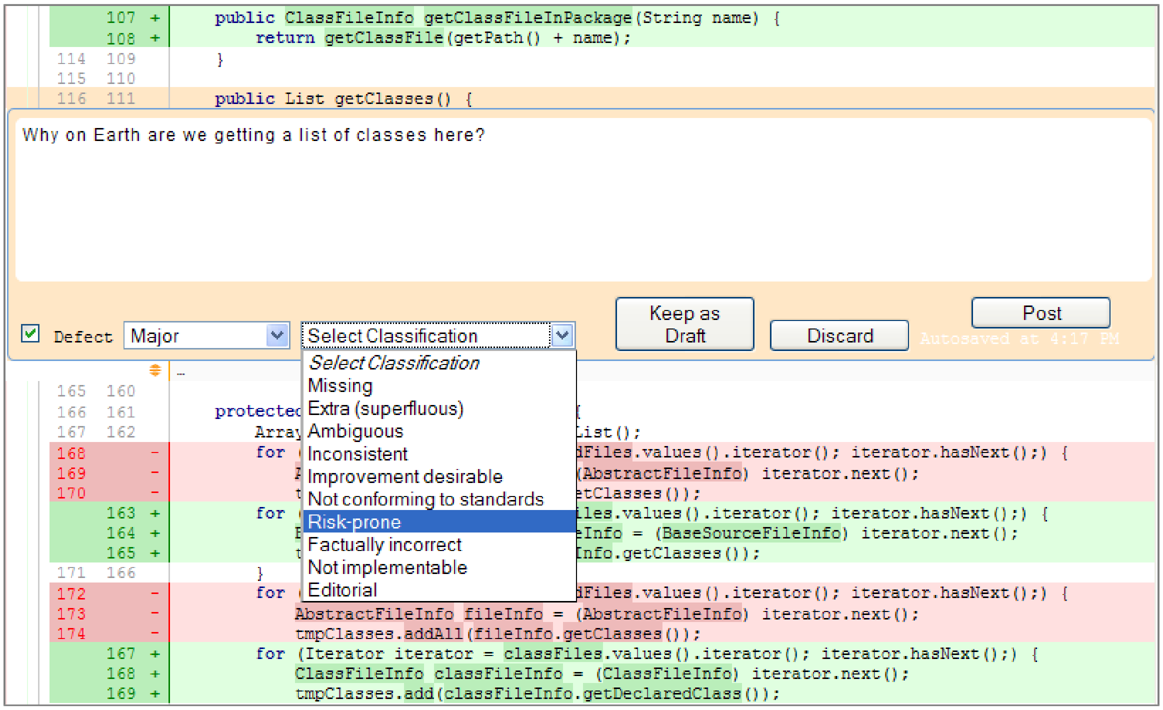
On this page:

- Defects in Crucible Comments
- Changing Classification Settings
- Default Crucible Classifications
    - Ranking
    - Classification

Defects in Crucible Comments

Defects are comments made by reviewers that indicate a problem in a review. Defects can be classified by rank and type, custom classifications can also be defined. The default classifications are shown in the screenshot below.

Screenshot: The List of Defect Classifications



```
+   public ClassFileInfo getClassFileInPackage(String name) {
+       return getClassFile(getPath() + name);
    }

    public List getClasses() {
```

Why on Earth are we getting a list of classes here?

☑ Defect Major

```
    protecte                                  {
        Array                                 List();
-       for (                                 dFiles.values().iterator(); iterator.hasNext();) {
-           A                                 (AbstractFileInfo) iterator.next();
-           t                                 etClasses());
+       for (                                 les.values().iterator(); iterator.hasNext();) {
+           B                                 eInfo = (BaseSourceFileInfo) iterator.next();
+           t                                 Info.getClasses());
        }
-       for (                                 dFiles.values().iterator(); iterator.hasNext();) {
-           AbstractFileInfo fileInfo = (AbstractFileInfo) iterator.next();
-           tmpClasses.addAll(fileInfo.getClasses());
+       for (Iterator iterator = classFiles.values().iterator(); iterator.hasNext();) {
+           ClassFileInfo classFileInfo = (ClassFileInfo) iterator.next();
+           tmpClasses.add(classFileInfo.getDeclaredClass());
```

Changing Classification Settings

To change the default classifications:

1. Open the Crucible Admin screen. The '**Admin Menu**' opens.
2. Click '**Customize Crucible Defect Classifications**' under '**Global Settings**' in the Admin Menu.

ℹ Only Crucible Admin users have access to this screen.
Any changes made within 'Customize crucible defect classifications' will only affect reviews created after the setting is changed.

Default Crucible Classifications

There are two default defect classifications that are preset in Crucible; ranking and classification. These settings (and their sub-categories) can be edited or removed; other custom classifications can be added.

Ranking

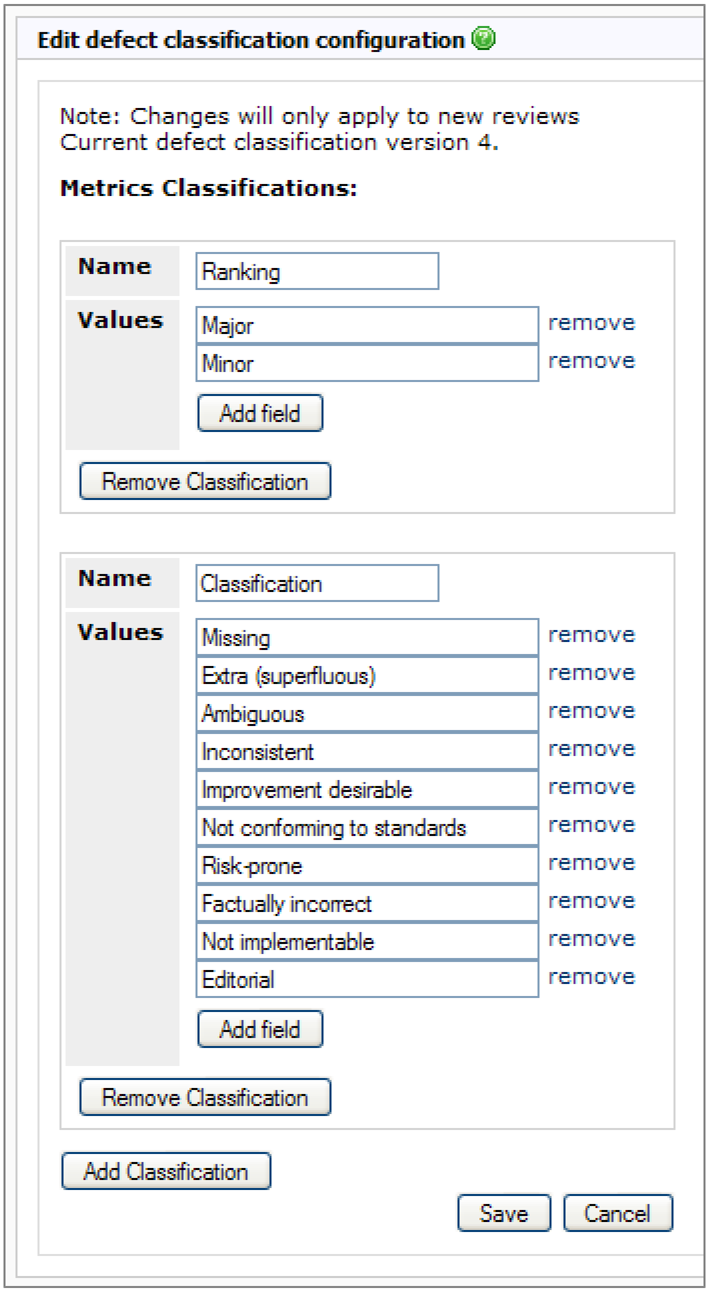
This classification can be set to **'Major'** or **'Minor'**, indicating the importance of the defect.

Classification

This setting helps to define the nature of the defect in particular detail. This classification can be set to one of the options in the following table; the meaning of these is detailed in the table below.

Value	Description
Missing	The defect applies to code or information that is missing (absent).
Extra (superfluous)	The defect applies to code or information that should be removed.
Ambiguous	The defect applies to code or information that is not clear or easy to understand.
Inconsistent	The defect applies to code or information that is applied in several different ways.
Improvement desirable	The defect applies to code or information that needs to be revised.
Not conforming to standards	The defect applies to code or information that breaks established conventions.
Risk-prone	The defect applies to code or information that takes unacceptable risks.
Factually incorrect	The defect applies to code or information that is wrong.
Not implementable	The defect applies to code or information that may be impossible to create.
Editorial	The defect applies to code or information where the classification as a defect may be subject to personal opinion.

Screenshot: Editing Defect Classifications in Crucible



Edit defect classification configuration

Note: Changes will only apply to new reviews
Current defect classification version 4.

Metrics Classifications:

Name	Ranking
Values	Major Minor

Name	Classification
Values	Missing Extra (superfluous) Ambiguous Inconsistent Improvement desirable Not conforming to standards Risk-prone Factually incorrect Not implementable Editorial

Customising the Welcome Message

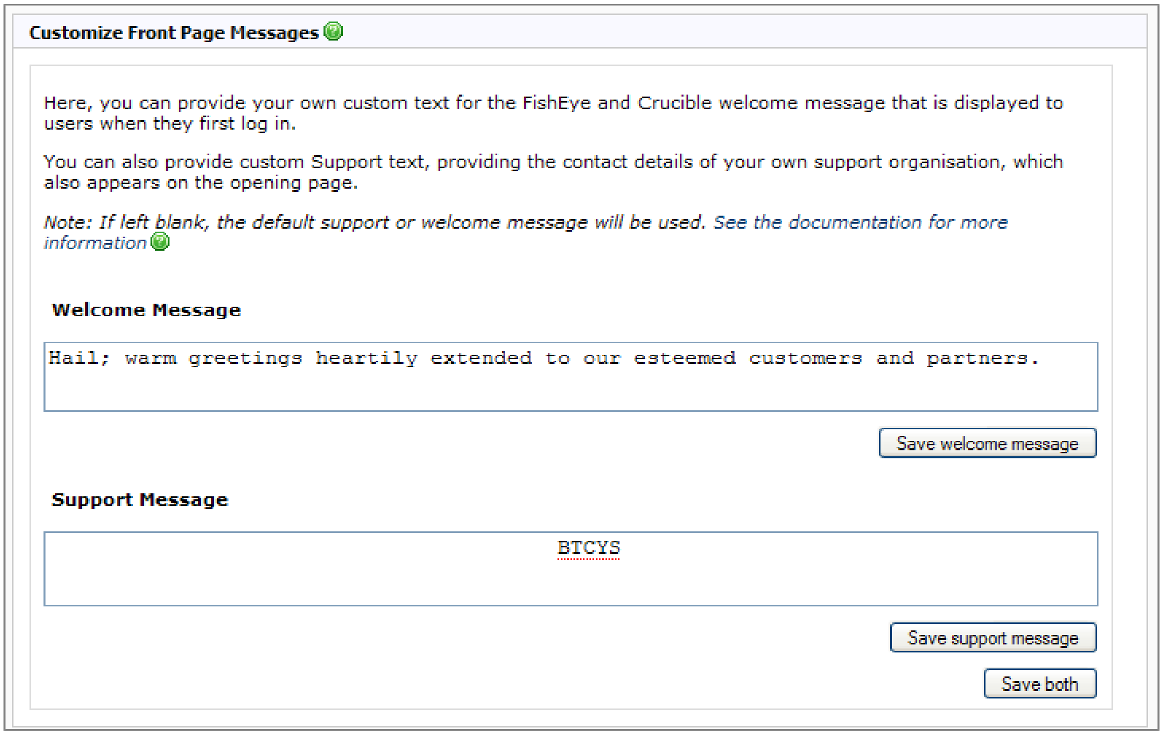
To customise the welcome message which is shown when Crucible opens, access the administration page and click '**Customize Front Page**' under '**Global Settings**' on the left navigation bar.

The '**Customize Front Page Messages**' page opens.

On this page, you can provide your own custom text for the Cruciblee welcome message that is displayed to users when they first log in. You can also provide custom Support text, providing the contact details of your own support organisation, which also appears on the opening page.

You can enter text into the boxes provided for either message and click the small '**Save Welcome Message**' or '**Save Support Message**' button to save it, or enter text for both messages and click '**Save All**'. The changes are made immediately.

Screenshot: Crucible Customize Welcome and Support Messages



Customize Front Page Messages

Here, you can provide your own custom text for the FishEye and Crucible welcome message that is displayed to users when they first log in.

You can also provide custom Support text, providing the contact details of your own support organisation, which also appears on the opening page.

Note: If left blank, the default support or welcome message will be used. See the documentation for more information

Welcome Message

Hail; warm greetings heartily extended to our esteemed customers and partners.

Support Message

BTCYS

Using HTML

The content in the welcome screen can be arranged using basic HTML tables, image references or anchor tags such as the following:

```
<a href="http://www.atlassian.com">Link to
   Atlassian Home Page</a>
```

Restoring the default messages

To revert to the default Welcome or Support messages, simply delete all text shown in the text box and click the corresponding '**Save**' button.

Manually editing the opening screen

You can also directly edit the XML file that contains the welcome and support messages. This file is called `config.xml`, located in your installation folder.

To do this, simply add the following XML tags to `config.xml`:

```
<content>
  <front-page-message>Example welcome message
    here</front-page-message>
  <support-message>Example support message
    here</support-message>
</content>
```

Deleting a Project

Admin users can delete projects under Crucible. To do this, follow the instructions below.

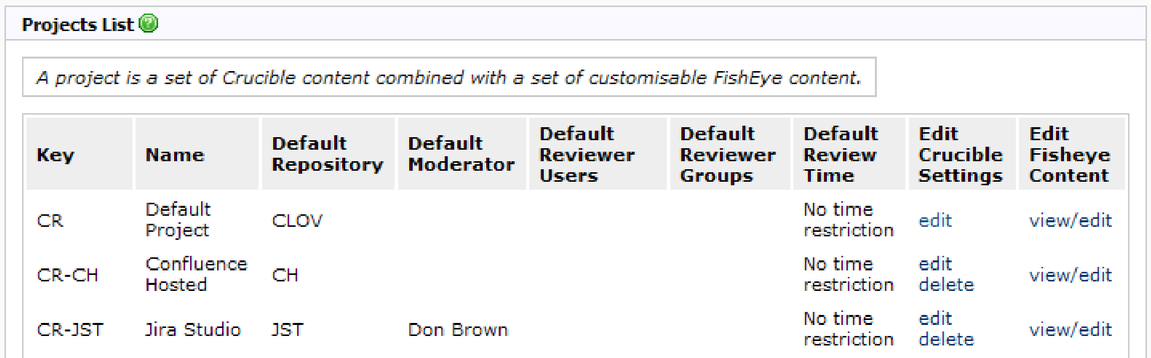
⚠ Deleted projects cannot be recovered.

Deleting Projects from the Project List

To delete a project from the Project List;

1. From the '**Admin Menu**', click '**Project List**'.
2. A list of projects appears. With care, click the '**Delete**' link situated to the right of the project you wish to remove. If empty, the project instantly disappears.
3. If the project contains reviews, you will be prompted to either delete all reviews in the project, or move them into the default project.

Screenshot: The Crucible Project Listing



A project is a set of Crucible content combined with a set of customisable FishEye content.

Key	Name	Default Repository	Default Moderator	Default Reviewer Users	Default Reviewer Groups	Default Review Time	Edit Crucible Settings	Edit Fisheye Content
CR	Default Project	CLOV				No time restriction	edit	view/edit
CR-CH	Confluence Hosted	CH				No time restriction	edit delete	view/edit
CR-JST	Jira Studio	JST	Don Brown			No time restriction	edit delete	view/edit

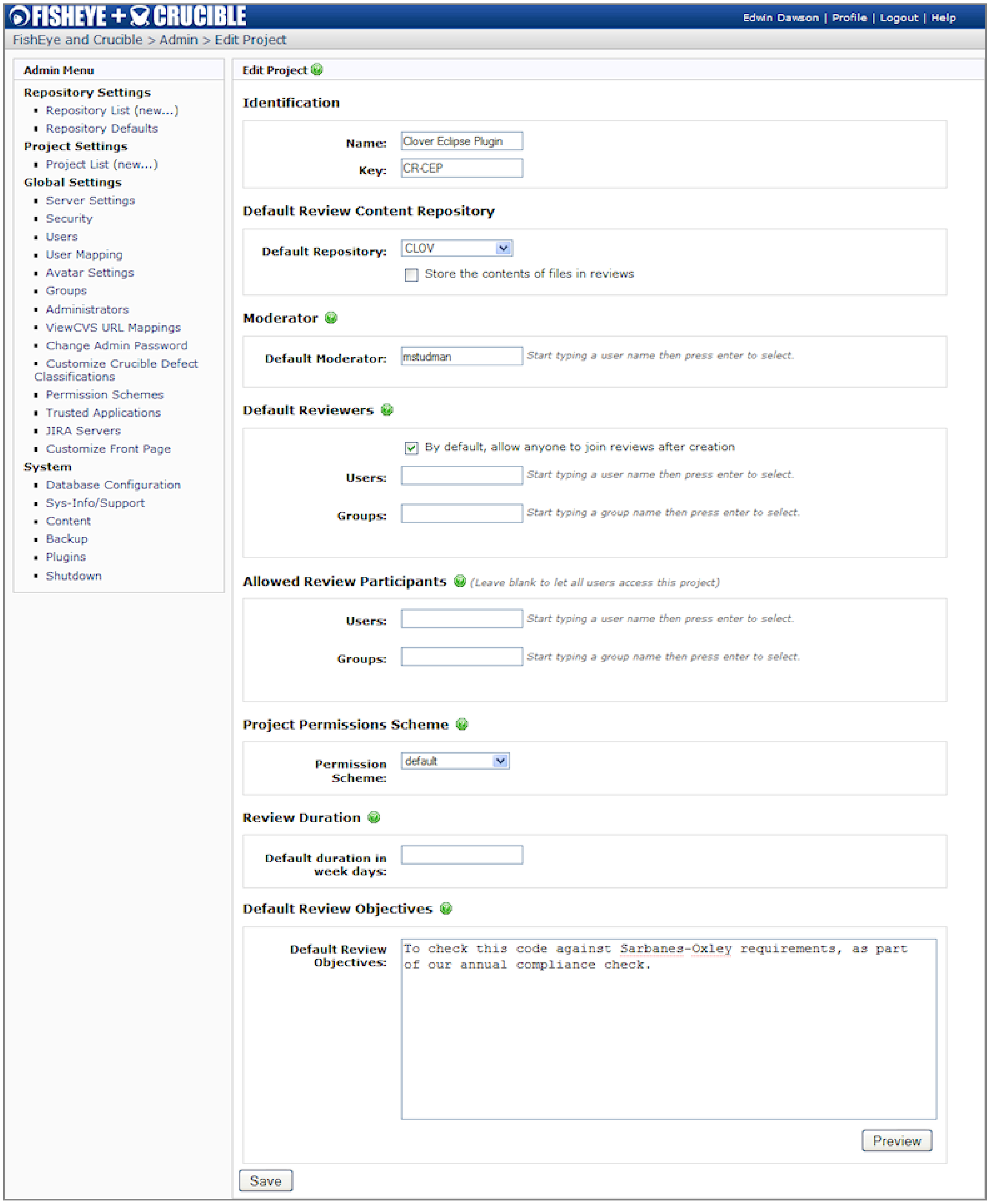
Editing a Project

Once projects are created, you can return to the project settings page to change the defaults such as repository, moderator, allowed reviewers, allowed groups and permissions.

To edit project settings,

1. From the '**Admin Menu**', click '**Project List**'.
2. The list of projects will be displayed. Click the '**Edit**' link for the desired project, which appears to the right of the existing project name.
3. The '**Edit Project'** page will be displayed. You can now adjust any of the given settings as desired.
4. In the '**Identification**' section, you can change the the plain language name (as displayed in the Crucible interface) and the project key (used when giving reviews their unique code names).
5. In the '**Default Review Content Repository**' field, you can adjust the repository which contains source code relating to this project.
   ⓘ This repository is the one that will be searched by default when you add files to a review.
   The check box here labelled '**Store the contents of files in reviews**' will cause the source files under review to be stored in the Crucible database along with the comments and review data. This will retain a copy of all the source files that go under review even in the event of disconnecting the repository from Crucible.
6. In the '**Default Moderator**' field, you can adjust the name of the person who will appear by default in the 'Moderator' field when you create a new review; or leave this field blank to force the review's creator to choose a moderator.
7. *(Optional)* Under '**Default Reviewers**', you can adjust the people to whom new reviews in this project will be assigned by default:
   - Select the '**Let allowed review participants join a review**' check-box if you wish to determine the default for the '**Allow anyone to join**' option on the 'Adding Reviewers' screen.
   - In the '**Users**' field, you can adjust the name(s) of individual users to whom new reviews will be assigned by default.
   - In the '**Groups**' field, you can adjust the name(s) of groups to whose members new reviews will be assigned by default.
8. *(Optional)* Under **Allowed Review Participants**', you can adjust who will be allowed to have a role (i.e. be an author/creator/moderator/reviewer) in this project's reviews:
   - In the '**Users**' field, you can adjust the list of individual users who will be eligible to be authors/creators/moderators/reviewers for reviews in this project*.
   - In the '**Groups**' field, you can adjust the list of groups whose members will be eligible to be authors/creators/moderators/reviewers for reviews in this project*.
     * These users will be the only ones whose names appear when a review is assigned.
9. In the '**Permission Scheme**' drop-down list, you can adjust the relevant permission scheme for this project. (A permission scheme controls who can perform particular actions, e.g. 'Create Review'.)
10. In the '**Review Duration**' section, you can define the default length of time (in week days) for reviews in this project.
11. In the '**Default Review Objectives Section**', you can define some text that will appear by default in the Review Objectives field of each new review. This text can be edited, as any text contained in the Review Objectives text box can.

Screenshot: The Edit Project screen in Crucible



Edit Project

Identification

Name: Clover Eclipse Plugin

Key: CR-CEP

Default Review Content Repository

Default Repository: CLOV

□ Store the contents of files in reviews

Moderator

Default Moderator: mstudman *Start typing a user name then press enter to select.*

Default Reviewers

☑ By default, allow anyone to join reviews after creation

Users: *Start typing a user name then press enter to select.*

Groups: *Start typing a group name then press enter to select.*

Allowed Review Participants *(Leave blank to let all users access this project)*

Users: *Start typing a user name then press enter to select.*

Groups: *Start typing a group name then press enter to select.*

Project Permissions Scheme

Permission Scheme: default

Review Duration

Default duration in week days:

Default Review Objectives

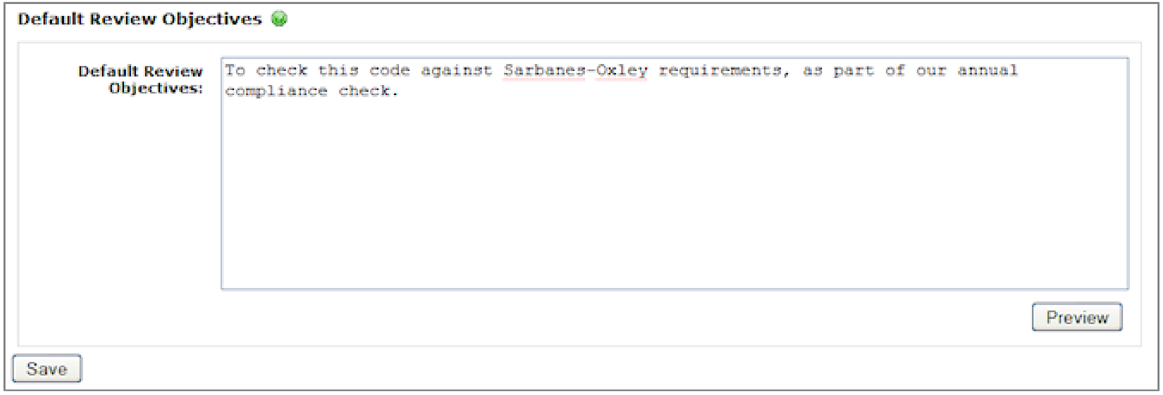
Default Review Objectives: To check this code against Sarbanes-Oxley requirements, as part of our annual compliance check.

Setting Default Review Objectives

To set default review objectives for all the reviews in a given project, carry out the following steps:

1. Open the '**Administration**' page, '**Project List**', then click '**Edit**' to open the Edit Project screen.
2. In the '**Default Review Objectives Section**', you can define some text that will appear by default in the '**Review Objectives**' field of each new review. Click '**Done**' to save your changes. This text can be edited, as any text contained in the Review Objectives text box can.

Screenshot: Default Review Objectives in Crucible



Default Review Objectives

Default Review Objectives: To check this code against Sarbanes-Oxley requirements, as part of our annual compliance check.

Setting the Default Review Duration for a Project

You can set a default time period (duration) that all reviews under a given project will run for. Reviews that are overdue will show up in red on the reviewer's dashboards.

To set a default duration for all reviews in a project,

1. From the '**Admin Menu**', click '**Project List**'.
2. The list of projects will be displayed. Click the '**Edit**' link for the desired project, which appears to the right of the existing project name.
3. The '**Edit Project'** page will be displayed. You can now adjust any of the given settings as desired.
4. In the '**Review Duration**' section, you can define the default length of time (in week days) for reviews in this project.

ⓘ Note that the 'Review Duration' only affects the default due date that appears when creating a review . The review's creator or moderator can specify a different date if they wish.

To see instructions for the other items on this page, see the documentation for Editing a Project.

JIRA Integration in Crucible

This page contains instructions for setting up JIRA integration.

ⓘ JIRA is Atlassian's issue tracking product, which can be used to manage projects and associated work.

⚠ Before you begin: Your JIRA instance will require configuration to support the following respective features:

- All JIRA integration features require Remote API access to be allowed on your JIRA instance.
- JIRA Hover and JIRA issues being automatically turned into hyperlinks requires Remote API access to be allowed on your JIRA instance.
- Creating JIRA subtasks from Crucible comments requires enabling sub-tasks in JIRA. It is also recommended that you enable unassigned issues in JIRA.
- Enabling Trusted Application support (where each user can use their own JIRA account) requires enabling Trusted Applications on your JIRA server and entering the details of your FishEye server there.

The instructions on this page have been tested with JIRA 4.0.0.

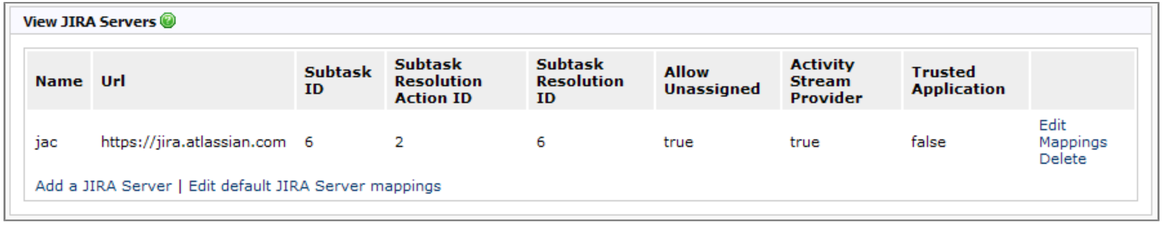
JIRA issues can be viewed in the main Dashboard view. This requires you to enter details on the required JIRA server(s) via the administration screens.

On this page:

Opening the Administration Screen for JIRA Integration

To set up JIRA integration, open the Administration screen and then click '**JIRA Servers**' under the '**Global Settings** ' sub-menu on the left navigation bar. The '**View JIRA Servers**' administration page opens.

Screenshot: The View JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

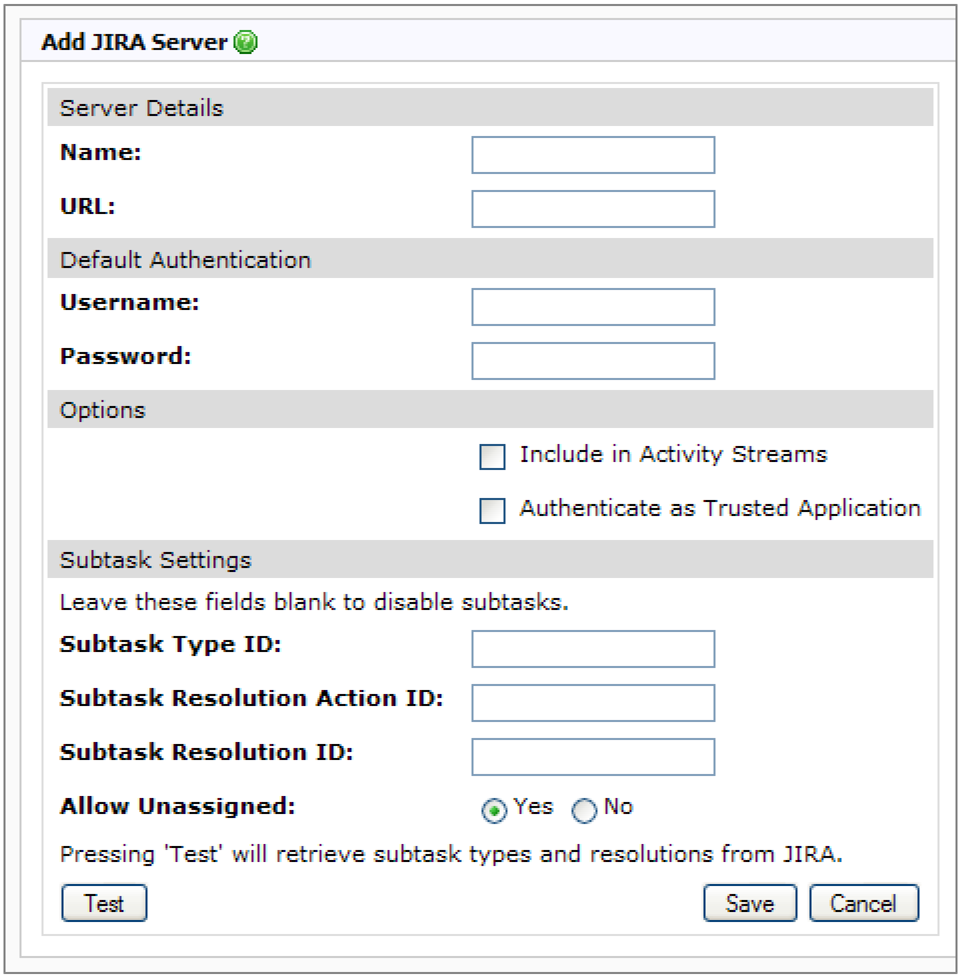
On the View JIRA Servers page, you can carry out a number of operations as described below.

Adding a New JIRA Server

To add a new JIRA server from the View JIRA Servers page, click '**Add JIRA Server**'.

The '**Add JIRA Server**' page opens.

Screenshot: The Add JIRA Server Page



Add JIRA Server

Server Details

Name:

URL:

Default Authentication

Username:

Password:

Options

□ Include in Activity Streams

□ Authenticate as Trusted Application

Subtask Settings

Leave these fields blank to disable subtasks.

Subtask Type ID:

Subtask Resolution Action ID:

Subtask Resolution ID:

Allow Unassigned: ● Yes ○ No

Pressing 'Test' will retrieve subtask types and resolutions from JIRA.

A number of fields and options must be filled out or selected on this page. See the table below for information on each field.

Option	Type	Description	Required
Name	Text Field	A descriptive name for the JIRA server.	Yes
URL	Text Field	The base URL of the JIRA server, e.g. {{ http://jira.atlassian.com }}	Yes
Username	Text Field	The username of an account on the JIRA instance. (All activity that takes place will be attributed to this user, unless using the Trusted Application setting).	Yes
Password	Text Field	The password for the account on the JIRA instance.	Yes
Include in Activity Streams	Check Box	Allows JIRA information to appear in the Activity Streams.	No
Authenticate as Trusted Application	Check Box	Allows the system to interface with JIRA and let users log on with their own accounts (and use their own accounts on the JIRA server). See the FishEye documentation and JIRA documentation on trusted applications. Known Issues.	No

Subtask Type ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number.	No
Subtask Resolution Action ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number (not required for standalone FishEye).	No
Subtask Resolution ID	Number	This is required to enable creating issues from a Crucible comment (not required for standalone FishEye). See the instructions on obtaining this number (not required for standalone FishEye).	No
Allow Unassigned	True/False Button	Allow unassigned sub-tasks. This setting must match your 'Allow Unassigned Issues ' setting in JIRA. .	No

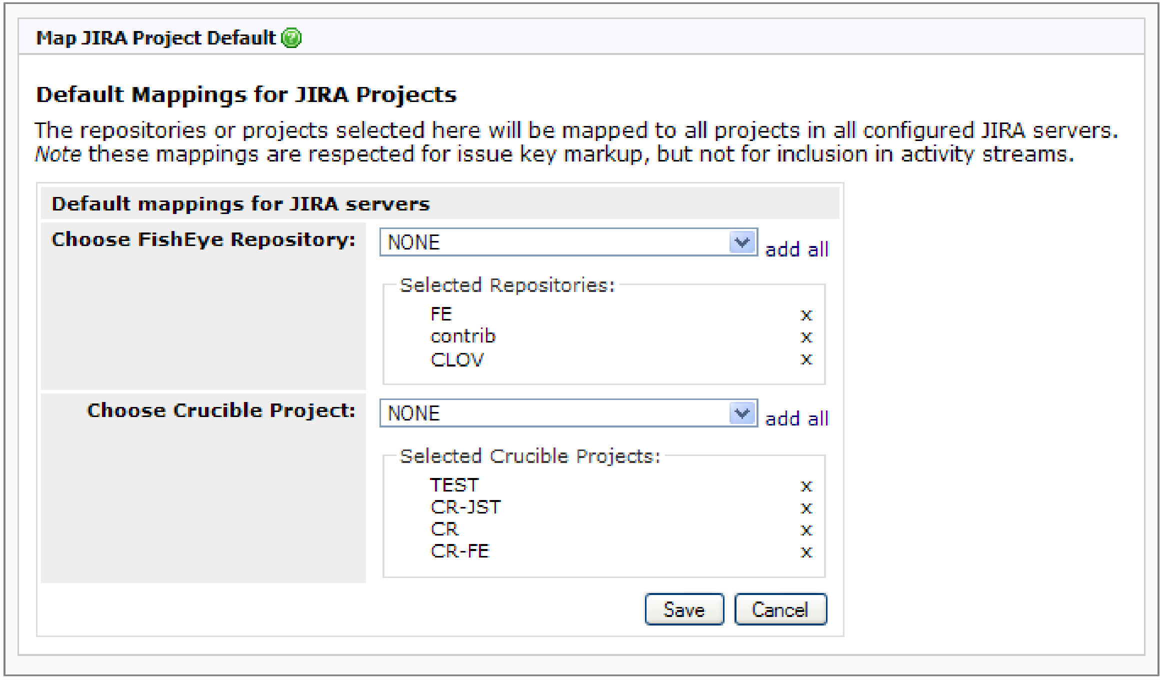
Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

Editing Default JIRA Server Mappings

This setting has two effects; it enables the feature that shows JIRA information in a dynamic window when you hover the mouse over a JIRA issue key; it will also turn every issue key into a hyperlink to that issue.

To enable this feature, click '**Edit Default JIRA Server Mappings**' from the View JIRA Servers page. The '**Map JIRA Project Default**' page opens.

Screenshot: The Default JIRA Server Mappings Page



Map JIRA Project Default

Default Mappings for JIRA Projects

The repositories or projects selected here will be mapped to all projects in all configured JIRA servers. *Note* these mappings are respected for issue key markup, but not for inclusion in activity streams.

Default mappings for JIRA servers

Choose FishEye Repository: NONE

Selected Repositories:
FE
contrib
CLOV

Choose Crucible Project: NONE

Selected Crucible Projects:
TEST
CR-JST
CR
CR-FE

On this page, select the FishEye Repositories or Crucible Projects that you wish to associate with all the JIRA servers that you have configured for use in FishEye/Crucible. You can click '**add all**' to quickly include them all in this category. You can remove individual items by clicking the small 'X' marks.

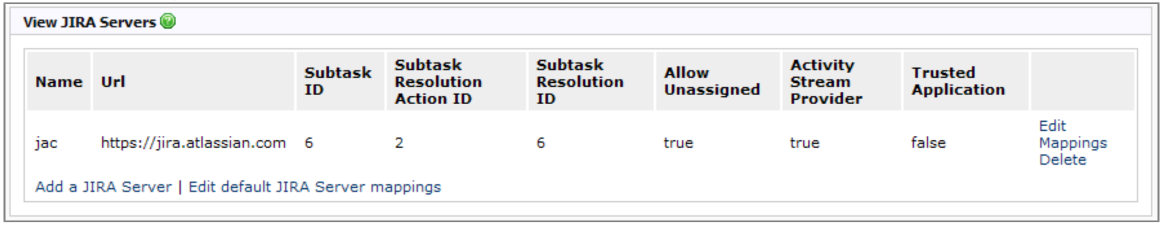
Once you've finished, click '**Save**'.

You should disable any existing Crucible linkers you have set up for JIRA, as they will override this feature and prevent the dynamic dialog box from appearing when you mouse over an issue.

Operations on Existing Servers

Once you have configured an existing JIRA server, there are three main operations you can carry out on it: '**Edit**', '**Mappings**' and '**Delete**'. These options appear on the far right of the screen.

Screenshot: Operations in the JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

Edit settings for an existing JIRA server

When you click '**Edit**', you can adjust any of the general settings you configured when you first added the server.

Edit mappings for an existing JIRA server

When you click '**Mappings**', a page is loaded that is almost identical to the '**Default Mapping**' screen, but allows you to choose mappings only for that specific JIRA server.

Delete an existing JIRA server

Clicking '**Delete**' will remove the server from the list.

Obtaining Subtask Values for Crucible Configuration

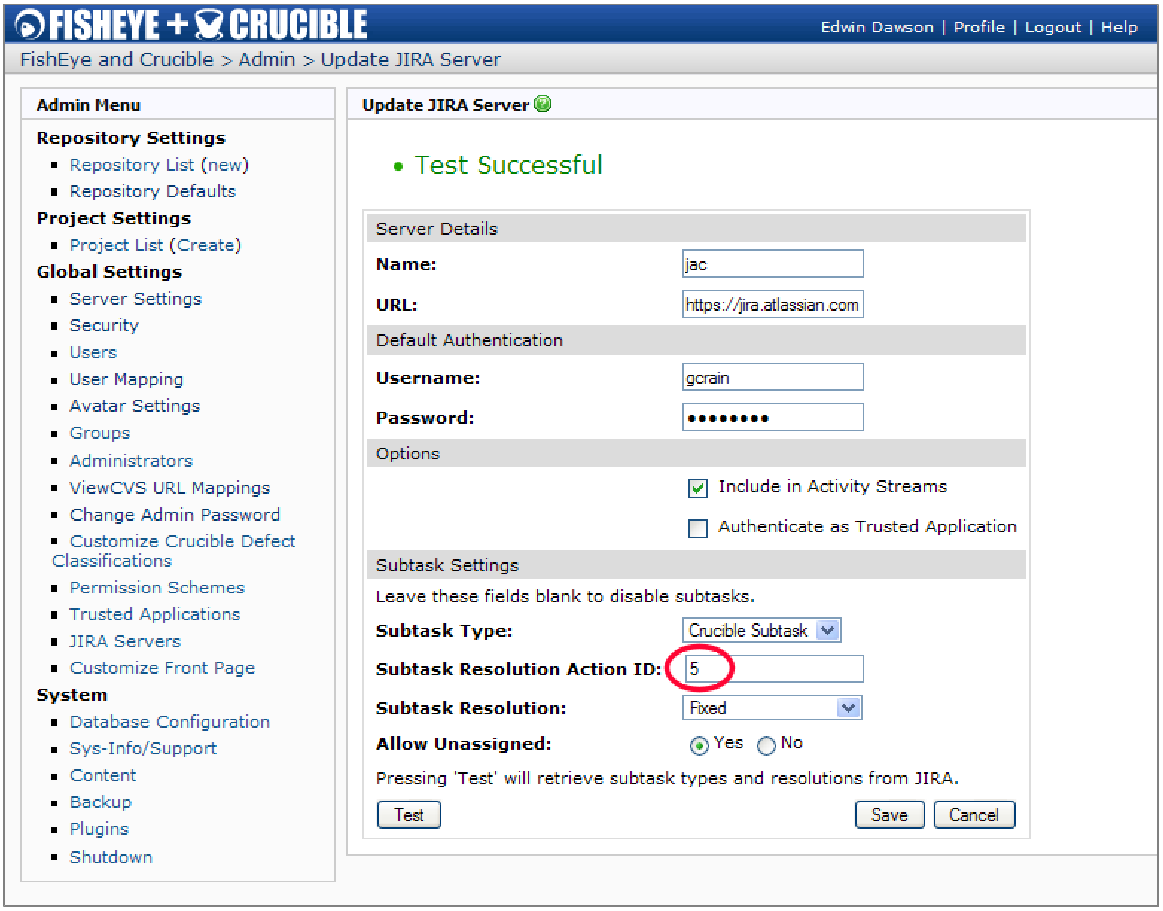
Obtaining the Subtask ID Values

This value is required (along with the Subtask Resolution ID and Subtask Resolution Action ID) to enable creating issues from a Crucible comment. This is the subtask type that will be created when you create a JIRA subtask in Crucible.

To set this up in Crucible, carry out the following steps.

1. Enable sub-tasks on your JIRA instance from the '**JIRA Administration**' > '**Sub-Tasks**' page. See the JIRA documentation for details on this step.
2. Return to the Crucible Administration screen and then click '**JIRA Servers**' under the '**Global Settings**' sub-menu on the left navigation bar. Click '**Edit**' next to the JIRA server you have configured.
3. Your JIRA server's basic details should appear. Click '**Test**' once again. The field for '**Subtask Type ID**' will change to a drop-down menu, showing the available subtask types. Choose the correct one. The field for '**Subtask Resolution**' will also turn into a drop-down menu. Select the desired item from this menu as well.
4. Save your Crucible configuration settings.

Screenshot: Filling in Subtask Values



Update JIRA Server

- Test Successful

Server Details

Name:	jac
URL:	https://jira.atlassian.com

Default Authentication

Username:	gcrain
Password:	••••••••

Options

☑ Include in Activity Streams

□ Authenticate as Trusted Application

Subtask Settings

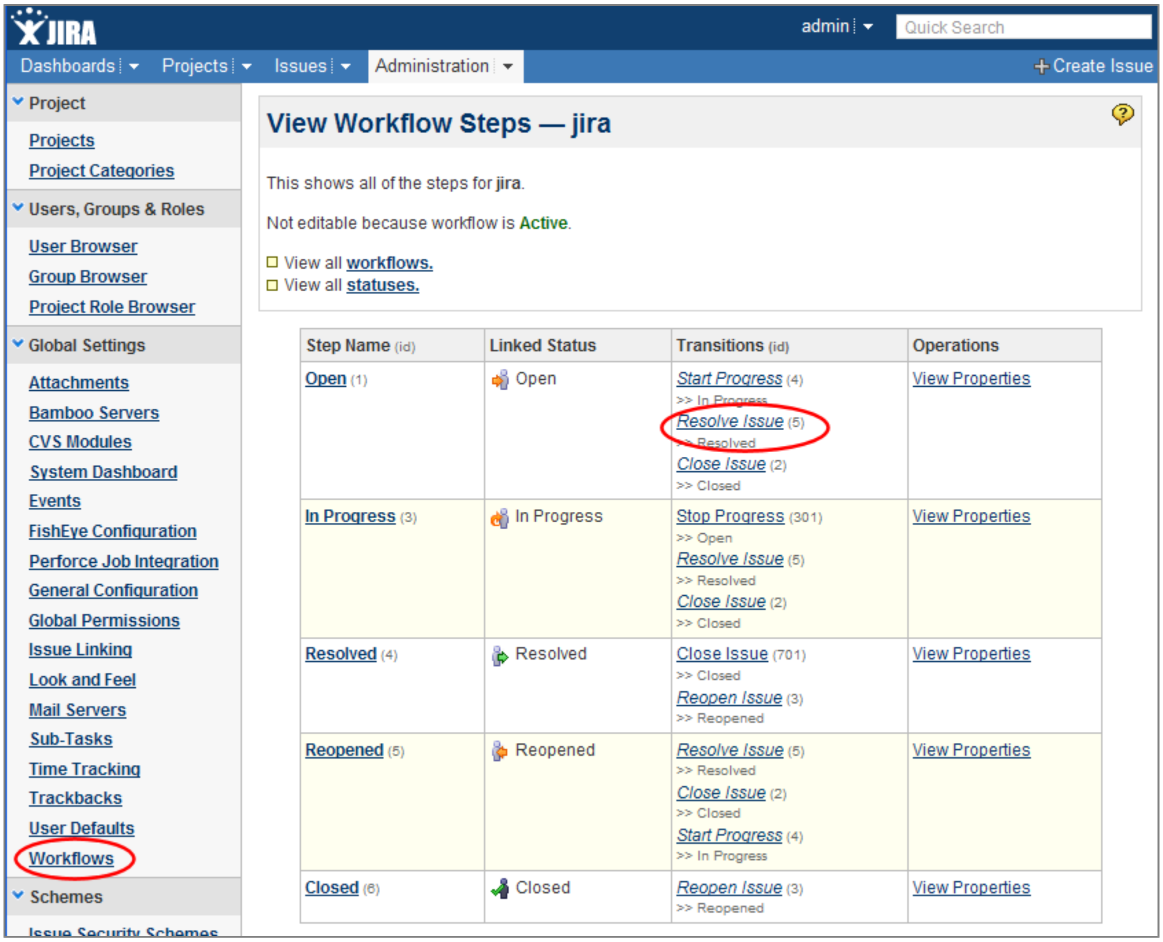
Leave these fields blank to disable subtasks.

Subtask Type:	Crucible Subtask
Subtask Resolution Action ID:	5
Subtask Resolution:	Fixed
Allow Unassigned:	● Yes ○ No

Pressing 'Test' will retrieve subtask types and resolutions from JIRA.

5. Open your JIRA instance and go to '**Administration**' > '**Workflows**'. The '**Workflows**' screen opens. By default, the '**JIRA**' workflow is shown on screen in a table.
6. Click the '**Steps**' link in the far right table cell. The '**View Workflow Steps — JIRA**' page opens.
7. The '**Subtask Resolution Action ID**' is in the '**Open**' row, under the '**Transitions**' column. Look at the link in that cell named '**Resolve Issue**'. The ID number is shown in brackets next to that heading '**Resolve Issue**' (shown in the screenshot below as 5).
8. Enter the number into the field in Crucible.
9. Save your Crucible configuration settings.
10. Your Crucible JIRA integration should now be complete.

Screenshot: Obtaining the Subtask Resolution Action ID



View Workflow Steps — jira

This shows all of the steps for **jira**.

Not editable because workflow is **Active**.

- View all workflows.
- View all statuses.

Step Name (id)	Linked Status	Transitions (id)	Operations
Open (1)	Open	Start Progress (4) >> In Progress Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
In Progress (3)	In Progress	Stop Progress (301) >> Open Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
Resolved (4)	Resolved	Close Issue (701) >> Closed Reopen Issue (3) >> Reopened	View Properties
Reopened (5)	Reopened	Resolve Issue (5) >> Resolved Close Issue (2) >> Closed Start Progress (4) >> In Progress	View Properties
Closed (6)	Closed	Reopen Issue (3) >> Reopened	View Properties

Known Issues

If you elect to use Trusted Applications for authentication with your JIRA server, activity streams and subtasks created from review defects will be generated using the currently logged in user. However JIRA project mapping and issue key linking (including the associated 'hovering' content) will be retrieved using the user specified on the JIRA Server configuration page in the FishEye administration section.

We are working towards supporting Trusted Applications for issue key linking and project mapping. If this issue is important to you, please vote for CRUC-1910.

See Also

- The JIRA documentation on Integrating JIRA with FishEye, which enables you to view FishEye data from within JIRA.
- The FishEye documentation on JIRA Integration in FishEye, which enables you to view JIRA data from within FishEye.

Migrating to an External Database

This page contains instructions on migrating your Crucible database from its default embedded form to an external database. This may be useful for the following reasons:

- **Improved Protection Against Data Loss**: The Crucible built-in database, running HSQLDB is somewhat susceptible to data loss during system crashes. External databases are generally more resistant to data loss during a system crash.

- **Performance & Scalability**: if you have many users on your Crucible instance, running the database on the same server as FishEye may slow it down. When using the embedded database, the database will always be hosted and run on the same server as Crucible.
- **Data Stored in the Crucible Database**: The Crucible database stores all information besides the cache for repository scans. This means all reviews, comments, review states, user data and user preferences information.

On this page:

- Overview
- Migrating to MySQL
- Migrating to PostgreSQL
- Support for other Databases

Overview

Crucible and FishEye offer alternatives to the built-in HSQLDB database for storing its relational data. At the time of writing, MySQL and PostgreSQL are supported (see System Requirements for version numbers). This page outlines the steps required for switching to an external database.

Migrating to MySQL

To switch from the built-in HSQLDB database to MySQL, install MySQL and follow the steps below.

1. Download the MySQL JDBC driver .JAR file from the MySQL website and copy the .JAR to your `FISHEYE_INST/lib` directory (create the `lib/` directory if it doesn't already exist). Restart FishEye or Crucible to have it pick up the driver.

2. Create a UTF-8 Database:

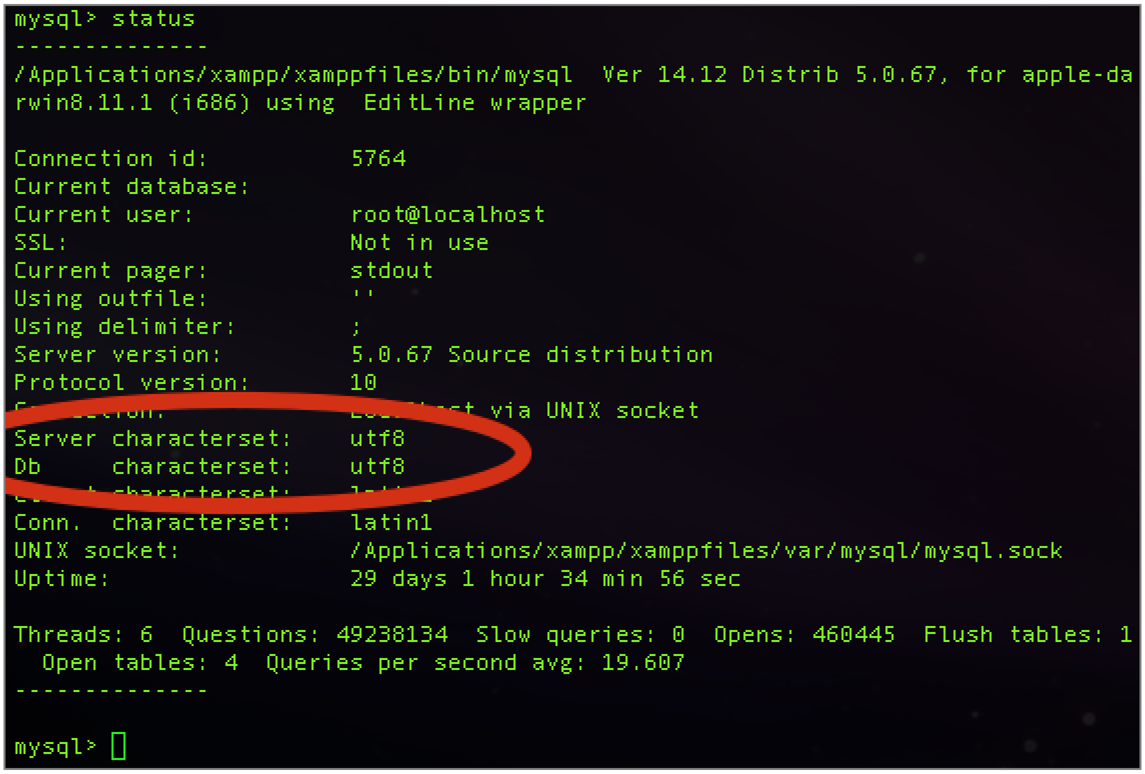
```
CREATE DATABASE crucible CHARACTER SET utf8
COLLATE utf8_bin;
```

3. You will also need to set the `Server Characterset` to `utf8`. This can be done by adding the following in `my.ini` for Windows or `my.cnf` for other OS. It has to be declared in the Server section, which is the section after `[mysqld]`:

```
[mysqld]
default-character-set=utf8
```

4. Use the **`status`** command to verify database character encoding information:

Screenshot: Using the MySQL Status Command



```
mysql> status
--------------
/Applications/xampp/xamppfiles/bin/mysql  Ver 14.12 Distrib 5.0.67, for apple-da
rwin8.11.1 (i686) using  EditLine wrapper

Connection id:          5764
Current database:
Current user:           root@localhost
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.0.67 Source distribution
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8
Db     characterset:    utf8
Client characterset:    latin1
Conn.  characterset:    latin1
UNIX socket:            /Applications/xampp/xamppfiles/var/mysql/mysql.sock
Uptime:                 29 days 1 hour 34 min 56 sec

Threads: 6  Questions: 49238134  Slow queries: 0  Opens: 460445  Flush tables: 1
  Open tables: 4  Queries per second avg: 19.607
--------------

mysql> 
```

5. Create a user that can log in from the host that Crucible or FishEye is running on and make sure that the user has full access to the newly created database. In particular, the user should be allowed to create and drop tables, indexes and other constraints.

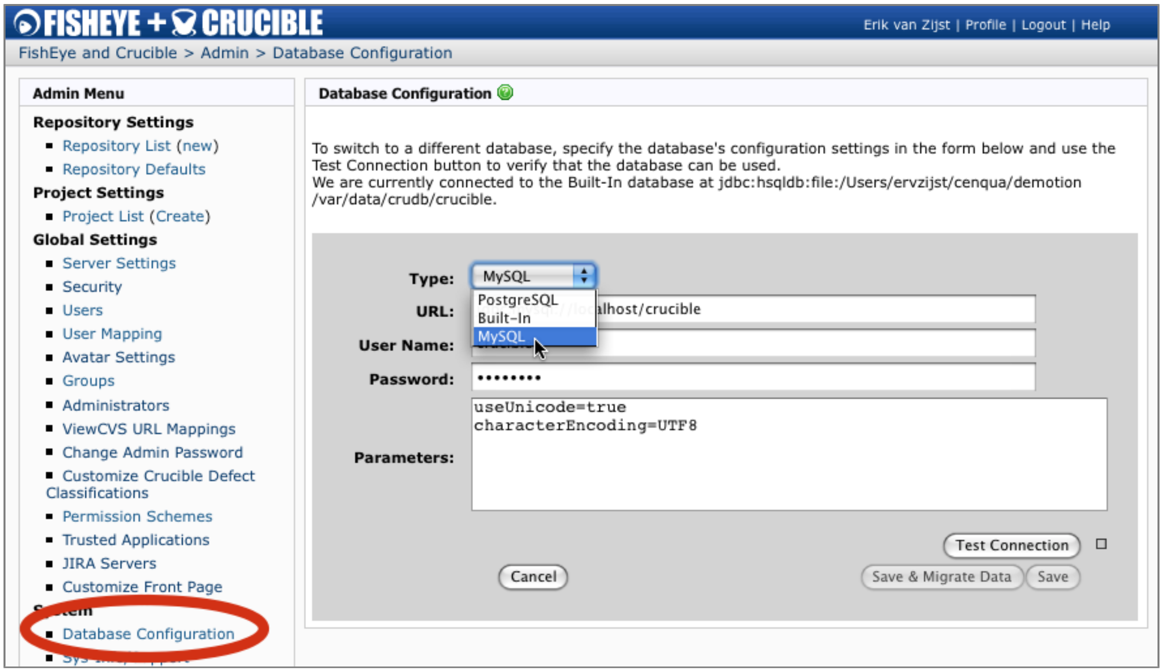
For instance, when Crucible and MySQL run on the same machine (accessible through `localhost`), issue the following commands (replacing username and password with the appropriate values):

```
mysql> grant all on crucible.* to
'username'@'localhost' identified by
'password';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

6. With the database prepared, navigate to the '**Database Configuration**' section in the admin interface, select MySQL from the drop down and fill out the database URL, username and password.

Then click '**Test Connection**' to verify that Crucible or FishEye can log in to the database:

Screenshot: Testing the Connection



Database Configuration

To switch to a different database, specify the database's configuration settings in the form below and use the Test Connection button to verify that the database can be used.
We are currently connected to the Built-In database at jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion/var/data/crudb/crucible.

Type:	MySQL
URL:	lhost/crucible
User Name:	
Password:	••••••••

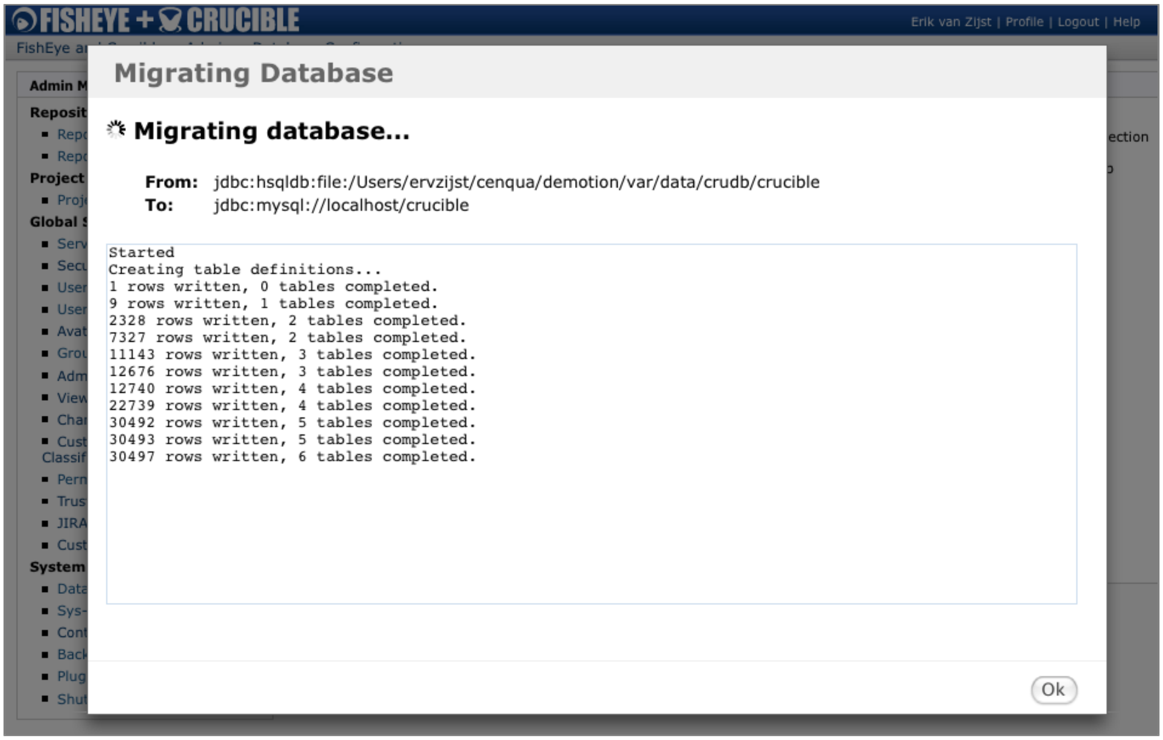
Parameters:	useUnicode=true
	characterEncoding=UTF8

If this fails, verify that you have the MySQL JDBC driver .JAR file in the classpath (by placing the .JAR file in `FISHEYE_INST/lib`). Also, ensure that the database user can log in to the database from the machine that Crucible or FishEye is running on and that all the required privileges are present.

7. Click '**Save & Migrate Data**' to start the migration process.

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Screenshot: Migrating the Database



Migrating Database

Migrating database...

From: jdbc:hsqldb:file:/Users/ervzijst/cenqua/demotion/var/data/crudb/crucible
To: jdbc:mysql://localhost/crucible

```
Started
Creating table definitions...
1 rows written, 0 tables completed.
9 rows written, 1 tables completed.
2328 rows written, 2 tables completed.
7327 rows written, 2 tables completed.
11143 rows written, 3 tables completed.
12676 rows written, 3 tables completed.
12740 rows written, 4 tables completed.
22739 rows written, 4 tables completed.
30492 rows written, 5 tables completed.
30493 rows written, 5 tables completed.
30497 rows written, 6 tables completed.
```

Migrating to PostgreSQL

To switch from the built-in HSQLDB database to PostgreSQL, install PostgreSQL and follow the steps below.

1. Download the PostgreSQL JDBC driver .JAR file from the PostgreSQL website and copy the .JAR to your `FISHEYE_INST/lib` directory (create the `lib/` directory if it doesn't already exist). Restart FishEye or Crucible to have it pick up the driver.

2. Create a new database user (replacing 'username' and 'password' with the appropriate values):

```
$ psql
> create user username password 'password';
```

3. Create a UTF-8 database and make the newly created user the owner:

```
> create database crucible ENCODING 'UTF-8'
OWNER username;
```

4. Make sure the user has full access to the database:

```
> grant all on database crucible to username;
```

During the migration process (which will take several minutes, depending on the size of your database and network throughput), the product will be inaccessible to users and external API clients. Users will see a maintenance screen that informs them of the process. Should the migration fail for any reason, Crucible will not switch to the new database and report on the encountered problems. Because the destination database may now contain some, but not yet all data, drop all tables, indexes and constraints before attempting a new migration.

Support for other Databases

Crucible and FishEye currently ship with support for MySQL and PostgreSQL as external databases (see System Requirements for version numbers).

If you are looking for support for Oracle or Microsoft SQL Server, please vote for the issues below. Your vote will help us prioritise them.

- **Request Oracle Support**: CRUC-1489
- **Request MS-SQL Support**: CRUC-1407

If you are using another database product that you would like to see supported, please create a JIRA issue for it under the Crucible project.

Trusted Applications

This page contains information about trusted application support in Crucible and how you can configure a trusted application relationship between Crucible and JIRA or Confluence.

On this page:

A 'trusted application' is an application that can access specific functions in Crucible, on behalf of any user — without the user logging in to Crucible.

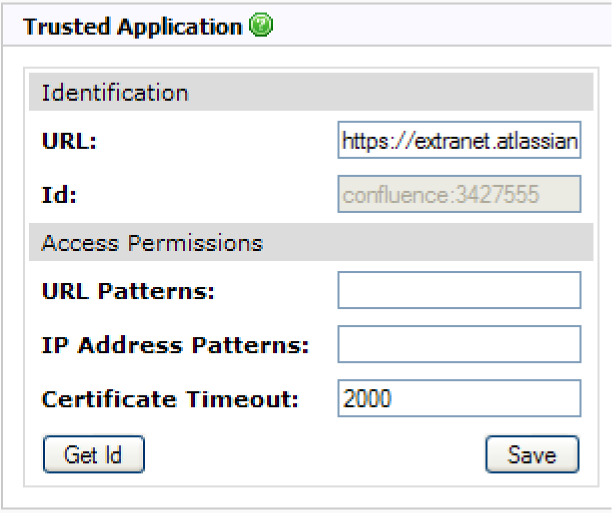
ℹ Crucible and FishEye share the same trusted applications; an application trusted by FishEye is also trusted by Crucible. At this time, JIRA 3.12 and Confluence 2.7 onwards can be configured as trusted applications.

> ℹ Before you begin, note that configuring a trusted application requires the transmission of sensitive data. To prevent 'man-in-the-middle attacks', it is recommended that you use an encrypted SSL connection while configuring a trusted application.

Adding a Trusted Application

To add a trusted application to Crucible:

1. Access the '**Administration Screen**'.
2. Click '**Trusted Applications**' under '**Global Settings**' on the left navigation bar.
3. Click '**Add a Trusted Application**'. The '**Trusted Application**' screen opens.



Trusted Application

Identification

URL: https://extranet.atlassian

Id: confluence:3427555

Access Permissions

URL Patterns:

IP Address Patterns:

Certificate Timeout: 2000

Screenshot: Configuring Trusted Applications

On this page, there are two areas, the '**Identification**' area and the '**Access Permissions**' area.

Configuring Identification Settings

Under the '**Identification**' heading, there are two fields, '**URL**' and '**Id**'.

URL field

In this field is where you will enter the Trusted Application Public Key URL of the application you wish to trust. For example, if your application's base URL is;

```
'http://www.mycompany/jira/'
```

you would enter that into the URL field. Once you've done this, click the 'Get ID' button. Crucible will then retrieve the Trust Certificate Id from the other application and display it in the Id field. If this step fails, you may not have not entered the correct URL for the other application.

Id field

This field contains the **Trust Certificate ID**, once you have filled out the URL field correctly (see above) and clicked the '**Get ID**' button. The contents of this field are not editable.

Configuring Access Permissions

Under the **Access Permissions** heading, there are three fields, **URL Patterns**, **IP Address Patterns** and **Certificate Timeout**. These allow you to further restrict requests from a trusted application.

URL Patterns field

With this field, you can limit the access a trusted application has to Crucible. It it not necessary to specify anything for this field; in fact a blank value is a sensible default. The default behaviour is no restriction.

The text that you specify should not include your hostname, IP address or port number, rather it relates to folders on the server, that start with the text you provide.

For example, if you use this setting:

```
/foo
```

then Crucible will trust only the requests to Crucible URLs starting with `/foo`, e.g. `/foo/bar`, `/foobar` and `/foo/bar/baz/x`. You can specify multiple URLs by separating them with a comma.

> URL Patterns do not support wildcard characters or regular expressions in Crucible.

IP Address Patterns field

With this field, you can limit the trusted network addresses for other applications. You can use wildcards to specify a number range, and multiple addresses can be separated with commas. For example, if you use this setting:

```
192.168.*.*,127.0.0.0
```

then Crucible will only trust requests from machines with the IP addresses 192.168.*anything*.*anything*(a group of network addresses) and 127.0.0.0 (a single host). The default is no restriction.

Certificate Timeout field

With this field, you can set the number of milliseconds before the certificate times out. This feature's purpose is to prevent 'replay attacks'. For example, if an attacker intercepts a request, they may attempt to extract the certificate and send it again independently. With the certificate timeout, the application will be able to tell that this is no longer a valid request. The default value is 1000 (one second).

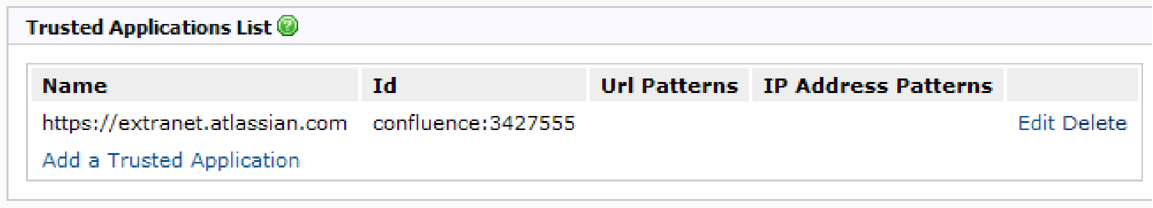
> ⚠ A shorter time out is more secure, but if set too short, it may cause valid requests to be rejected on slower networks.

Once you've finished entering the settings for the Trusted Application, click the '**Save**' button to confirm and activate the trust relationship.

Editing Trusted Application Settings

Once you have configured your trusted application(s), you can view the settings on the main **'Trusted Applications'** page.

Screenshot: Trusted Applications list



Trusted Applications List

Name	Id	Url Patterns	IP Address Patterns	
https://extranet.atlassian.com	confluence:3427555			Edit Delete

Add a Trusted Application

From this screen, you can click **'Edit'** to make changes to the trusted application settings, or click **'Delete'** to remove the trust relationship for that application.

Crucible Release Notes

> ⚠ Crucible 2.1 has now been released. Read the Release Notes.

Crucible Release Notes and Changelogs

- Crucible Release Summary
- Crucible 2.1 Release Notes
  - Crucible 2.1 Changelog
- Crucible 2.0 Release Notes
  - Crucible 2.0 Upgrade Notes
  - Crucible 2.0 Changelog
- Crucible 2.0 Beta Release Notes
  - JIRA Integration in Crucible 2.0 Beta
  - Crucible 2.0 Beta Upgrade Notes
  - Crucible 2.0 Beta Reviewer's Guide
- Crucible 1.6 Release Notes
  - Crucible 1.6 Changelog
    - Crucible 1.6.3 Upgrade Guide
- Crucible 1.5 Release Notes
  - Crucible 1.5 Changelog
- Crucible 1.2 Release Notes
  - Crucible 1.2 Upgrade Guide
  - Crucible 1.2 Changelog
- Crucible 1.1 Release Notes
  - Crucible 1.1 Upgrade Guide
  - Crucible 1.1 Changelog

- For changes prior to 1.1, see the 1.0.x Changelog

Installation

You can download Crucible from here. Information on installing Crucible can be found here.

If upgrading from a previous version, please follow the Upgrade Guide.

- As of version 1.0, Crucible now requires a JVM version 1.5 or later. Previously, 1.4+ was required.
- Crucible 1.1.4 includes FishEye 1.3.8.
- Upgrading from 1.0.4 (or earlier) will force a complete re-index of P4 repositories.

Crucible 2.1 Release Notes

12 November 2009

> ℹ For details on minor releases since Crucible 2.1, see the Crucible Changelog.

Atlassian presents Crucible 2.1

Crucible 2.1 adds Wiki Markup rendering, a new review history dialog, new review blockers report, and runs significantly faster.

Highlights of this release:

- Wiki Markup Rendering
- Progress Tracking
- Usability and Productivity Updates
- Streamlined JIRA Integration
- Review Time Tracking
- Review History Dialog
- "Blockers" Reports
- Threaded Comments
- Plugin Developer Tools
- Plus numerous improvements and bug fixes

Thank you for your interest in Crucible 2.1.

See the documentation on Upgrading to this version.

Installing Crucible 2.1

You can now download Crucible 2.1 from here. See the documentation on Upgrading to this version.

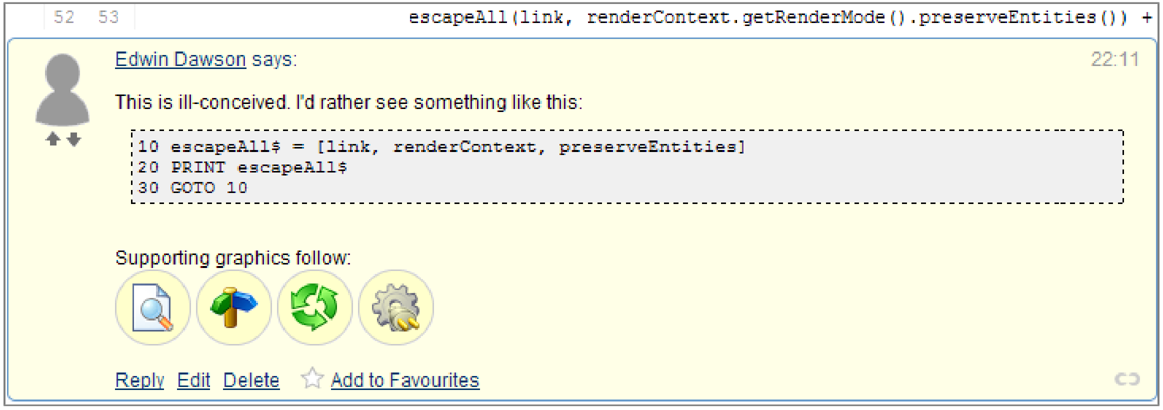
Highlights of Crucible 2.1

1

Wiki Markup Rendering

The Wiki Markup language that's used in Confluence and JIRA can now be rendered by Crucible. Review comments, review descriptions and commit messages will now be shown rendering Wiki Markup code, allowing insertion of images, diagrams and text formatting. See the documentation for more.

Screenshot: Wiki Markup Rendering in Crucible



```
escapeAll(link, renderContext.getRenderMode().preserveEntities()) +
```

Edwin Dawson says: 22:11

This is ill-conceived. I'd rather see something like this:

```
10 escapeAll$ = [link, renderContext, preserveEntities]
20 PRINT escapeAll$
30 GOTO 10
```

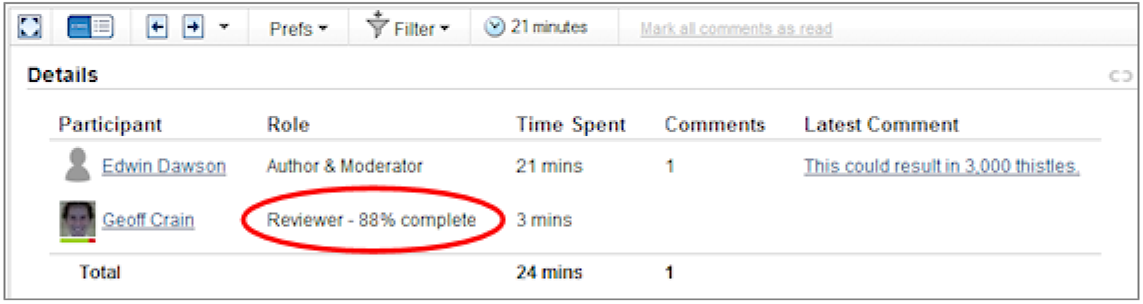
Supporting graphics follow:

2

Progress Tracking

While reviewing, Crucible will now automatically remember which files you've read and show this as a percentage in the Review Details panel. you no longer have to explicitly click a checkbox to mark a file as reviewed. If you've only skimmed the page and plan to revisit it, you can select that a file's status be left unread.

Screenshot: Crucible Progress Tracking



Details

Participant	Role	Time Spent	Comments	Latest Comment
Edwin Dawson	Author & Moderator	21 mins	1	This could result in 3,000 thistles.
Geoff Crain	Reviewer - 88% complete	3 mins		
Total		24 mins	1	

3

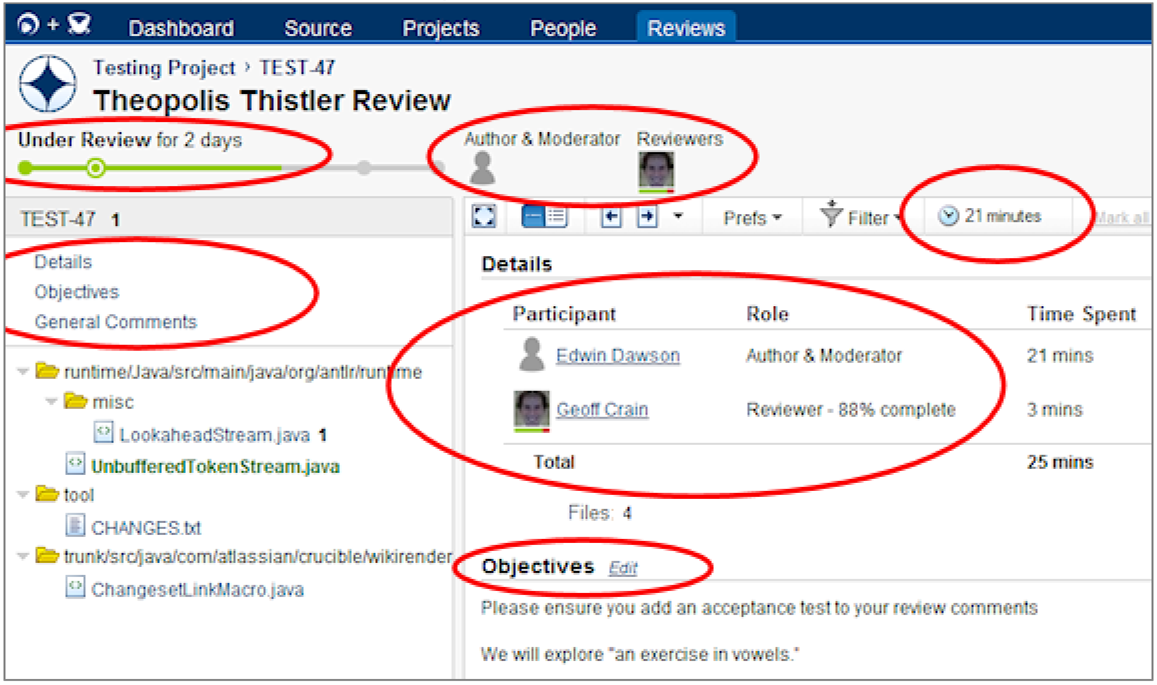
Usability and Productivity Updates

- **User Interface Update**
  The user interface has also been improved, consolidating items from the left navigation bar into the centre, freeing up space for the directory tree and other menus.
- **Floating File Mastheads**
  When viewing a review, you now have a floating masthead that contains information about the file in context. This gives you access to more meta-data about the file, you'll also have more control of the file and how it interacts with its peers. Additionally, more relevant data can be kept in focus.
- **Inline Editing for Review Details**

Sometimes you just want to tweak a review's title, objective or summary – you can now edit these inline. The title will give you a cue that its editable by turning yellow when you hover over it. Click it to start editing and save your changes. The other areas have an edit link for you to use, come with a preview and are Wiki Markup ready as you'd expect.

- **Default Review Objectives**
  You can pre-populate reviews with default text so that you can avoid manually entering the same objectives for many reviews, where the goals for each one are similar.
- **JIRA auto-linker**
  When you create a review now, the title and objectives are scanned, looking for a JIRA reference. And when we find one, that JIRA is automatically associated with the Review.
- **Performance**
  Performance was also enhanced. The team focused on the main review page, the users page and FishEye pages that display large changesets.
- **Simplified navigation**
  In reviews, you can now choose to navigate at comment, defect or file level. You can easily jump between these items using a new, universal navigation control.

Screenshot: Usability and Interface Updates



Testing Project › TEST-47

Theopolis Thistler Review

Under Review for 2 days

Author & Moderator Reviewers

Details

Participant	Role	Time Spent
Edwin Dawson	Author & Moderator	21 mins
Geoff Crain	Reviewer - 88% complete	3 mins
Total		25 mins

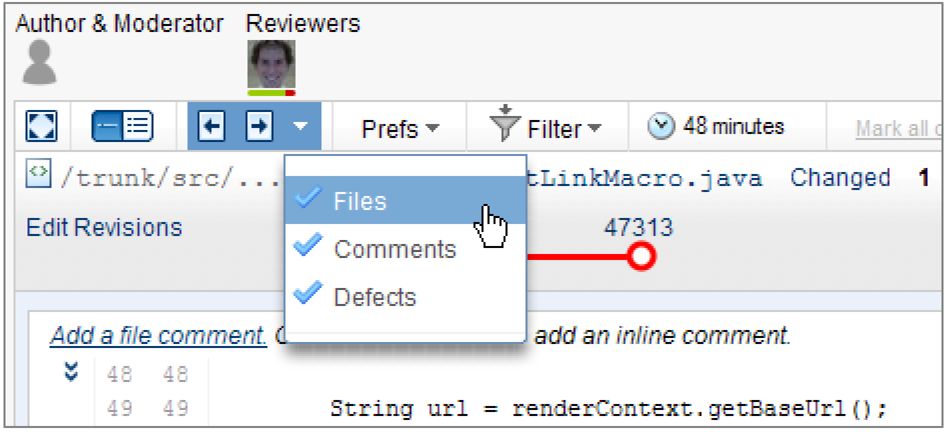
Files: 4

Objectives *Edit*

Please ensure you add an acceptance test to your review comments

We will explore "an exercise in vowels."

Screenshot: Simplified Review Navigation



/trunk/src/...tLinkMacro.java Changed 1

Add a file comment. C... add an inline comment.

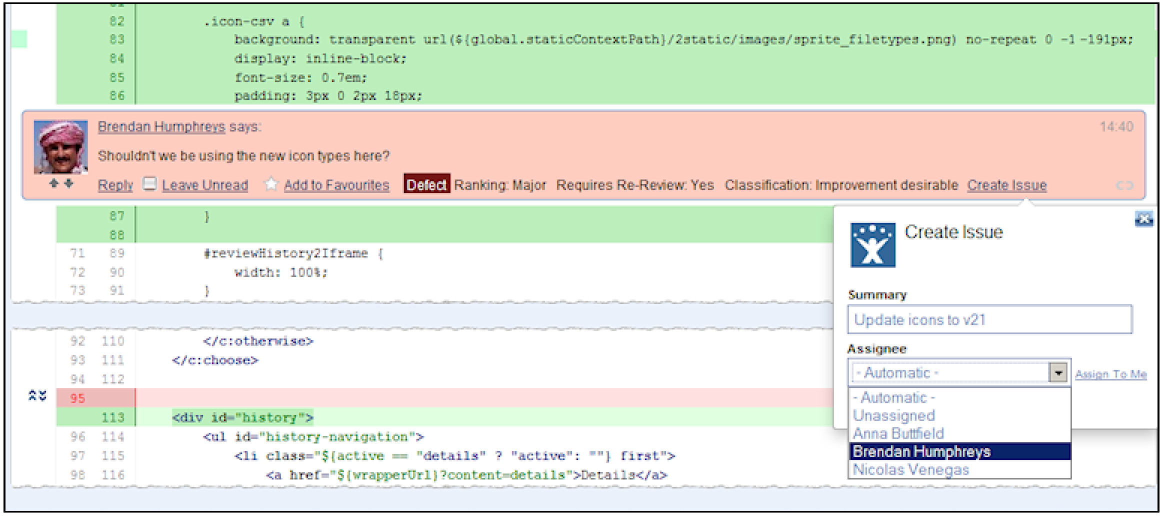
```
                String url = renderContext.getBaseUrl();
```

4

Streamlined JIRA Integration

When you create a review, the title and objectives are now parsed in the hunt for a JIRA reference. When we find one, that JIRA issue is automatically associated with the review. And if we've been a little over eager in sourcing a reference, simply click to remove it. Creating JIRA issues from within Crucible has also been enhanced, with an 'Assignee' drop-down being added:

Screenshot: Streamlined JIRA Integration in Crucible



```
    .icon-csv a {
        background: transparent url(${global.staticContextPath}/2static/images/sprite_filetypes.png) no-repeat 0 -1 -191px;
        display: inline-block;
        font-size: 0.7em;
        padding: 3px 0 2px 18px;
```

Brendan Humphreys says: 14:40

Shouldn't we be using the new icon types here?

Defect Ranking: Major Requires Re-Review: Yes Classification: Improvement desirable

```
    }

    #reviewHistory2Iframe {
        width: 100%;
    }

        </c:otherwise>
    </c:choose>

<div id="history">
    <ul id="history-navigation">
        <li class="${active == "details" ? "active": ""} first">
            <a href="${wrapperUrl}?content=details">Details</a>
```

Create Issue

Summary: Update icons to v21

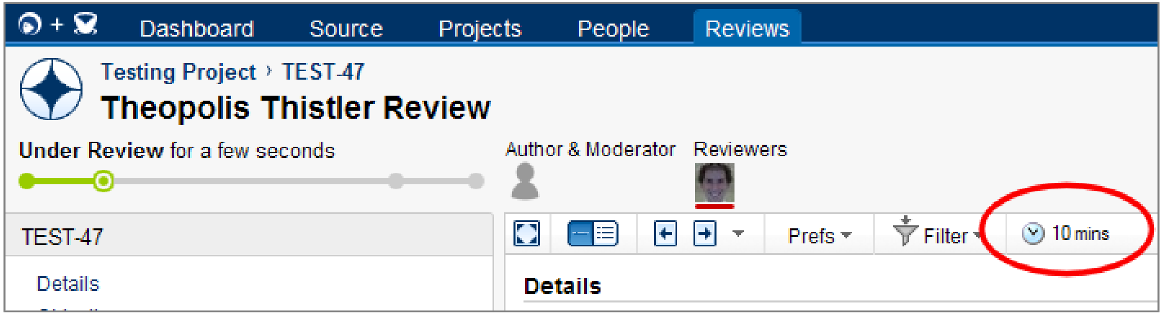
Assignee: - Automatic -

5

Review Time Tracking

Crucible now has time tracking for review participants. When you've got a review open in your browser, Crucible will track the time you have spent on that particular review. You can also click to change the amount of time recorded. Totals are displayed in the Review Details panel.

Screenshot: Crucible Time Tracking



Testing Project › TEST-47

Theopolis Thistler Review

Under Review for a few seconds

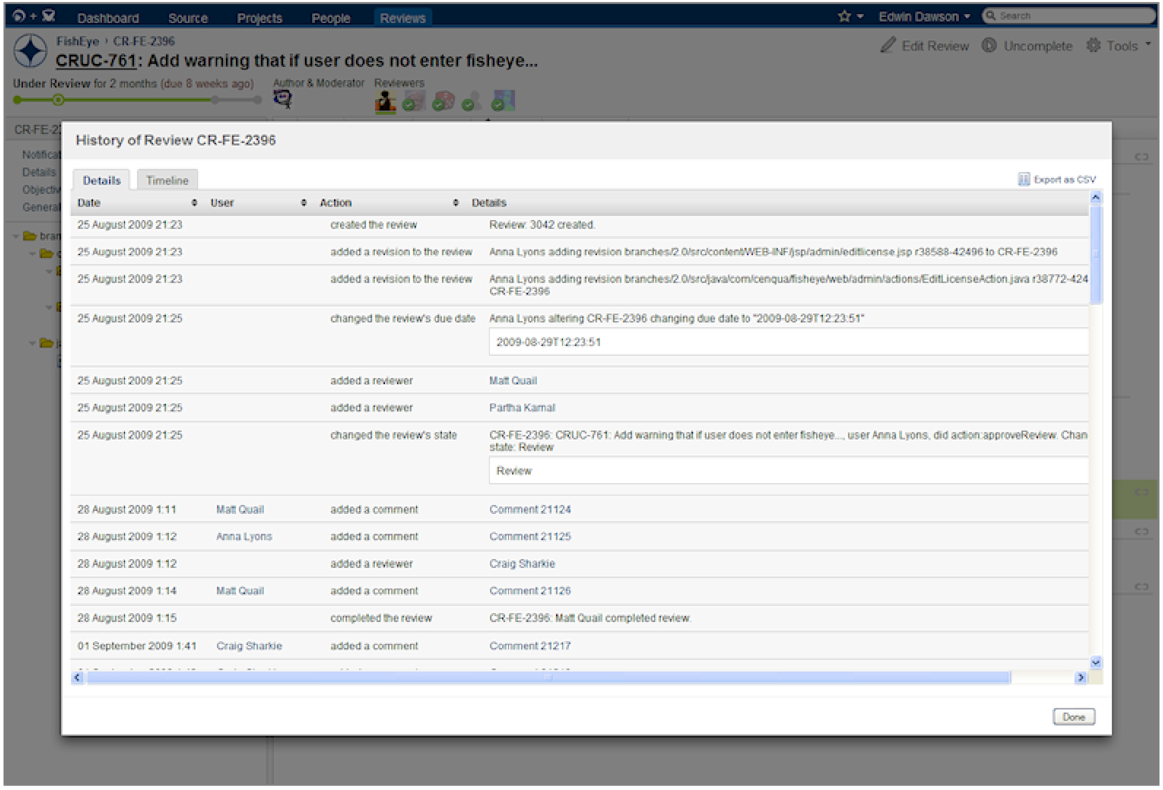
6

Review History Dialog

The new History Dialog differentiates old and new states of your interactions with a review. The result is that now you get richer information about those interactions and more control. You can sort the information by date, actor, or action. This information can also be displayed in the new timeline mode.

Additionally, you can get access to the entire Review history through the CSV download link in the upper right corner, allowing for easy data import into a spreadsheet or other application.

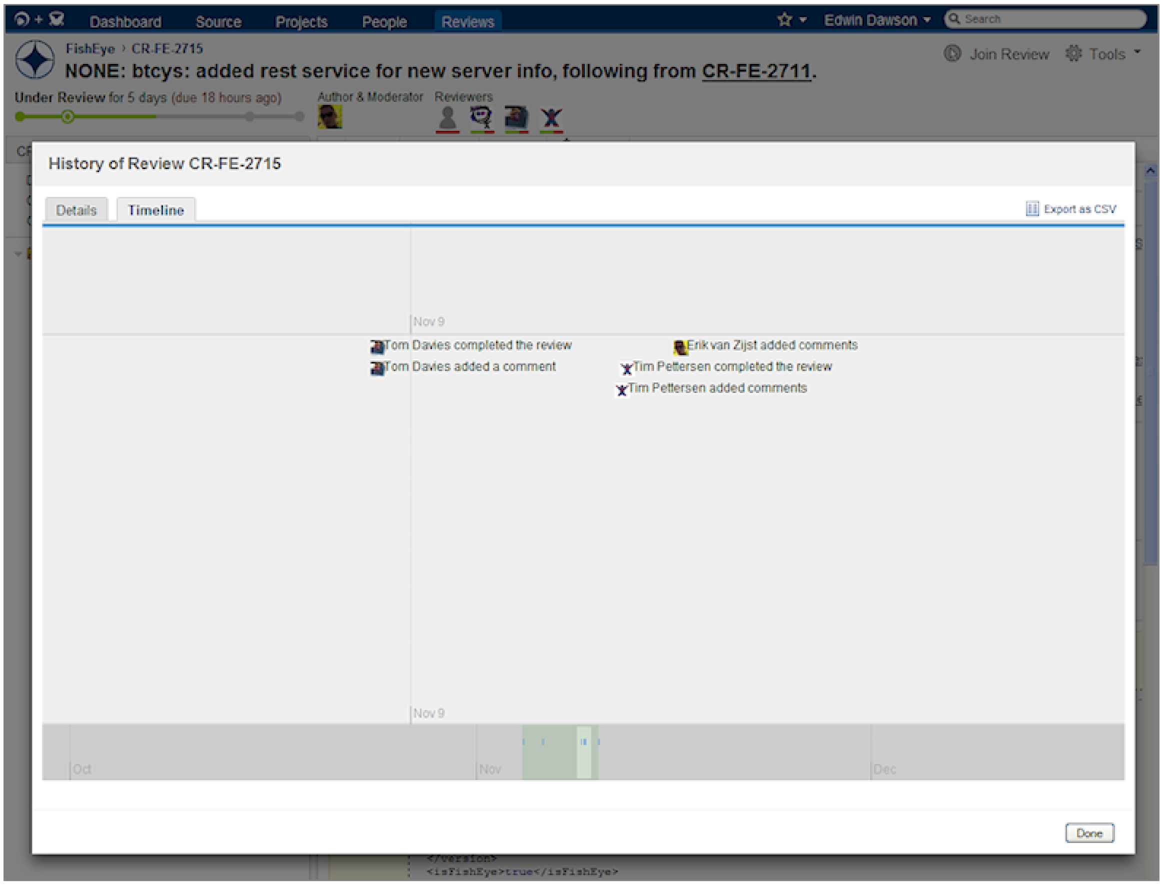
Screenshot: Crucible Review History Dialog



History of Review CR-FE-2396

Date	User	Action	Details
25 August 2009 21:23		created the review	Review: 3042 created.
25 August 2009 21:23		added a revision to the review	Anna Lyons adding revision branches/2.0/src/content/WEB-INF/jsp/admin/editlicense.jsp r38588-42496 to CR-FE-2396
25 August 2009 21:23		added a revision to the review	Anna Lyons adding revision branches/2.0/src/java/com/cenqua/fisheye/web/admin/actions/EditLicenseAction.java r38772-424 CR-FE-2396
25 August 2009 21:25		changed the review's due date	Anna Lyons altering CR-FE-2396 changing due date to "2009-08-29T12:23:51" 2009-08-29T12:23:51
25 August 2009 21:25		added a reviewer	Matt Quail
25 August 2009 21:25		added a reviewer	Partha Kamal
25 August 2009 21:25		changed the review's state	CR-FE-2396: CRUC-761: Add warning that if user does not enter fisheye..., user Anna Lyons, did action:approveReview. Chan state: Review Review
28 August 2009 1:11	Matt Quail	added a comment	Comment 21124
28 August 2009 1:12	Anna Lyons	added a comment	Comment 21125
28 August 2009 1:12		added a reviewer	Craig Sharkie
28 August 2009 1:14	Matt Quail	added a comment	Comment 21126
28 August 2009 1:15		completed the review	CR-FE-2396: Matt Quail completed review.
01 September 2009 1:41	Craig Sharkie	added a comment	Comment 21217

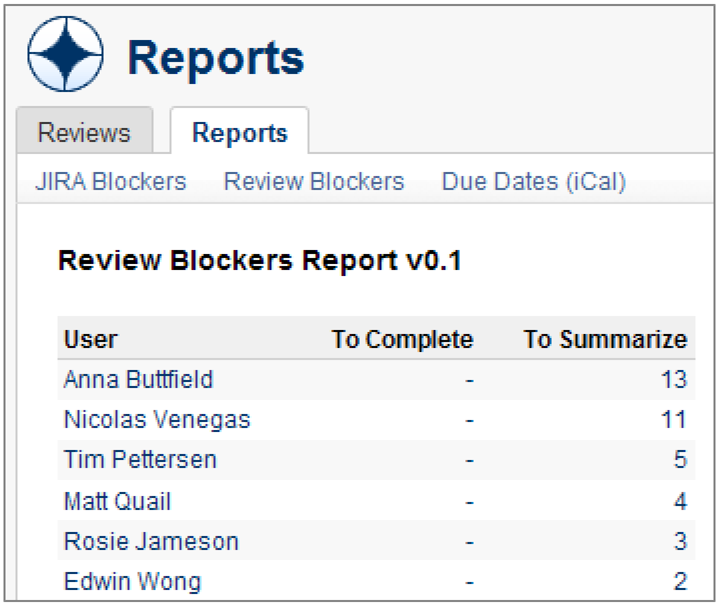
Screenshot: Timeline Mode in the Crucible Review History Dialog


Dashboard
Source
Projects
People
Reviews
Edwin Dawson
Search
FishEye › CR-FE-2715
NONE: btcys: added rest service for new server info, following from CR-FE-2711.
Join Review
Tools
Under Review for 5 days (due 18 hours ago)
Author & Moderator
Reviewers
History of Review CR-FE-2715
Details
Timeline
Export as CSV
Nov 9
Tom Davies completed the review
Tom Davies added a comment
Erik van Zijst added comments
Tim Pettersen completed the review
Tim Pettersen added comments
Nov 9
Oct
Nov
Dec
Done

"Blockers" Reports

Every now and then, someone in the team can become a bottleneck in the review process. The new review blockers report helps by identifying team members that have a lot of reviews waiting in their inbox. Additionally, the JIRA blockers report accepts a JIRA feed, which you can find under the 'View' drop-down on your JIRA 4 Issue Navigator page. Add the URL to the report and you get an insight to which issues and participants you need to follow up. Both these reports are plugins that come bundled with Crucible.

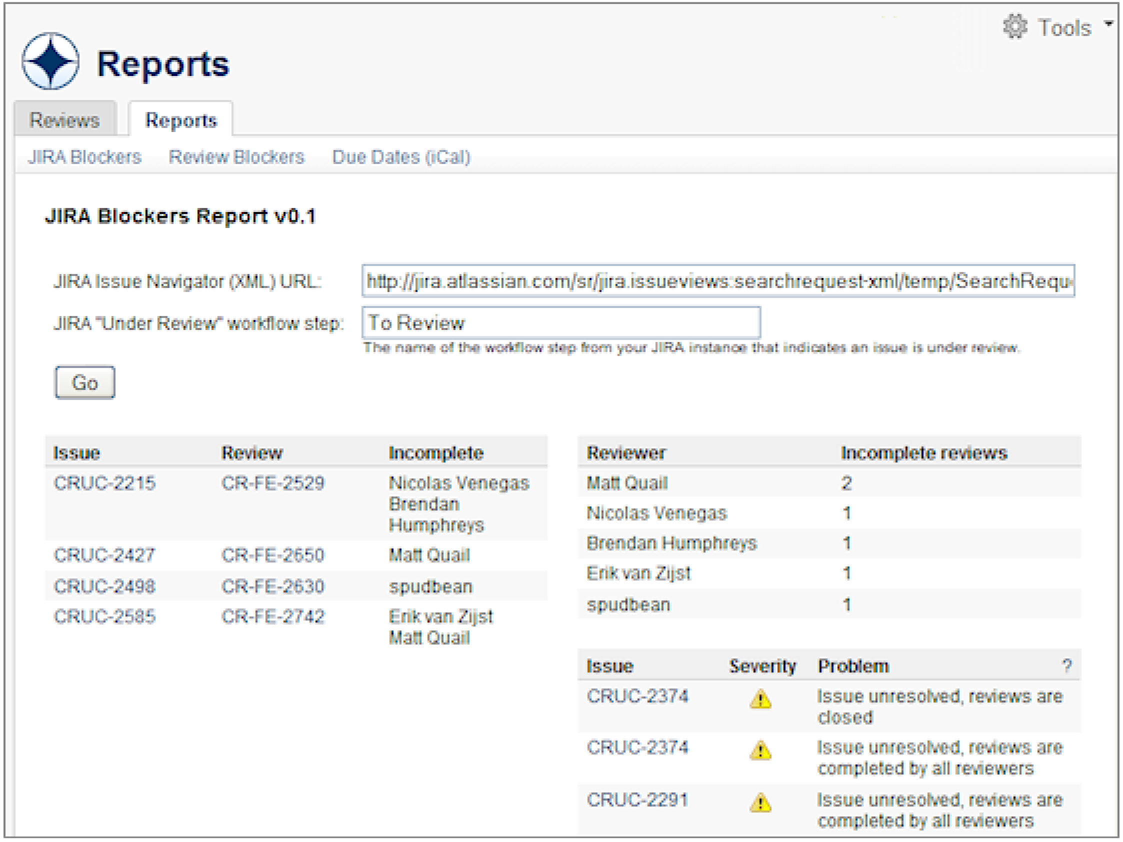
Screenshot: Crucible Review Blockers Report



Review Blockers Report v0.1

User	To Complete	To Summarize
Anna Buttfield	-	13
Nicolas Venegas	-	11
Tim Pettersen	-	5
Matt Quail	-	4
Rosie Jameson	-	3
Edwin Wong	-	2

Screenshot: Crucible JIRA Blockers Report



JIRA Blockers Report v0.1

JIRA Issue Navigator (XML) URL: http://jira.atlassian.com/sr/jira.issueviews:searchrequest-xml/temp/SearchRequ

JIRA "Under Review" workflow step: To Review

The name of the workflow step from your JIRA instance that indicates an issue is under review.

Issue	Review	Incomplete
CRUC-2215	CR-FE-2529	Nicolas Venegas Brendan Humphreys
CRUC-2427	CR-FE-2650	Matt Quail
CRUC-2498	CR-FE-2630	spudbean
CRUC-2585	CR-FE-2742	Erik van Zijst Matt Quail

Reviewer	Incomplete reviews
Matt Quail	2
Nicolas Venegas	1
Brendan Humphreys	1
Erik van Zijst	1
spudbean	1

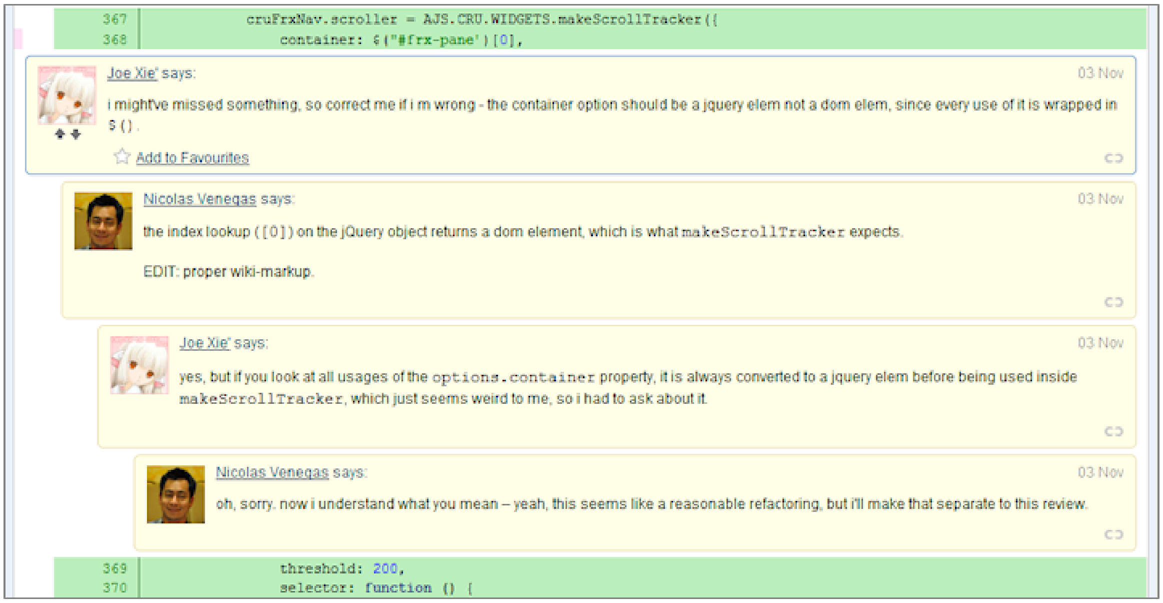
Issue	Severity	Problem ?
CRUC-2374	⚠	Issue unresolved, reviews are closed
CRUC-2374	⚠	Issue unresolved, reviews are completed by all reviewers
CRUC-2291	⚠	Issue unresolved, reviews are completed by all reviewers

8

Threaded Comments

Crucible comments now support fully threaded discussions. Threads can also be collapsed.

Screenshot: Crucible Threaded Comments



```
            cruFrxNav.scroller = AJS.CRU.WIDGETS.makeScrollTracker({
                container: $("#frx-pane")[0],
```

Joe Xie' says: 03 Nov

i might've missed something, so correct me if i m wrong - the container option should be a jquery elem not a dom elem, since every use of it is wrapped in `$()`.

Add to Favourites

Nicolas Venegas says: 03 Nov

the index lookup (`[0]`) on the jQuery object returns a dom element, which is what `makeScrollTracker` expects.

EDIT: proper wiki-markup.

Joe Xie' says: 03 Nov

yes, but if you look at all usages of the `options.container` property, it is always converted to a jquery elem before being used inside `makeScrollTracker`, which just seems weird to me, so i had to ask about it.

Nicolas Venegas says: 03 Nov

oh, sorry. now i understand what you mean – yeah, this seems like a reasonable refactoring, but i'll make that separate to this review.

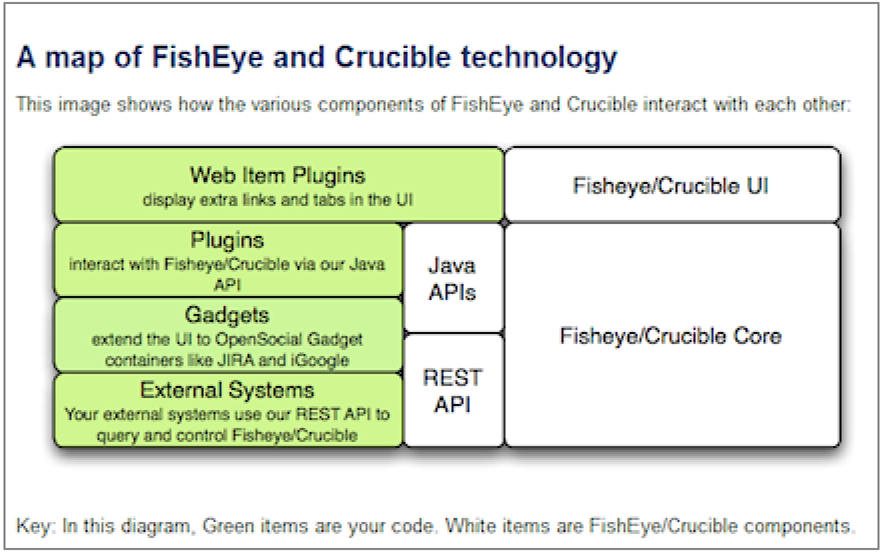
```
                threshold: 200,
                selector: function () {
```

9

Plugin Developer Tools

This release adds Atlassian Plugin SDK support to Crucible and FishEye, simplifying build management for plugin developers. The developer documentation for FishEye and Crucible has been co-located into a new documentation space as well.

Screenshot: Plugin Developer Tools for Crucible



A map of FishEye and Crucible technology

This image shows how the various components of FishEye and Crucible interact with each other:

Web Item Plugins
display extra links and tabs in the UI
Fisheye/Crucible UI
Plugins
interact with Fisheye/Crucible via our Java API
Java APIs
Gadgets
extend the UI to OpenSocial Gadget containers like JIRA and iGoogle
Fisheye/Crucible Core
External Systems
Your external systems use our REST API to query and control Fisheye/Crucible
REST API

Key: In this diagram, Green items are your code. White items are FishEye/Crucible components.

10

Plus numerous improvements and bug fixes

Visit our issue tracker to see the full list of improvements and bug fixes.

Crucible 2.1 Changelog

On this page:

- From 2.1.1 to 2.1.2
- From 2.1.0 to 2.1.1

From 2.1.1 to 2.1.2

19th November 2009

This is a bugfix release.

The following issues are addressed by this release:

JIRA Issues (3 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-2738	Getting java.lang.NoClassDefFoundError: Could not initialize class com.cenqua.fisheye.web.themer.BriefCheckinCommentTag	⬇	Closed
CRUC-2713	Caused by: org.osgi.framework.BundleException: Unresolved constraint in bundle 32: package; (&(package=org.apache.commons.collections)(version>=3.2.0)(!(version>=4.0.0)))	⬆	Closed
CRUC-2620	HSQLDB floods its sql log with AUTOCOMMIT TRUE/FALSE	⬇	Closed

From 2.1.0 to 2.1.1

17th November 2009

This is a bugfix release.

The following issues are addressed by this release:

JIRA Issues (25 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-2710	NPE when setting up instance and only entering a fisheye license	⬇	Closed
CRUC-2707	Paging of commit-only activity stream can be very slow	⬇	Closed
CRUC-2705	Document Supported Database server versions	⬇	Closed
CRUC-2704	Upgrading to 2.1 doesn't work with Postgresql 7.x	❗	Closed
CRUC-2703	Changeset Navigation in manage files is broken	⬇	Closed
CRUC-2702	Plugins can't be enabled because of apache commons version constraints	⬆	Closed
	Don't try to create a revisions drop down on the review contents tab when the		

CRUC-2700	revision is in a light scm repo which doesn't implement DirectoryBrowser	↓	Closed
CRUC-2698	Lucene Index for Review State Changes	↓	Closed
CRUC-2645	something wrong with python syntax highlighting	↓	Closed
CRUC-2191	Document supported database versions	↑	Closed
CRUC-2118	Defect properties on comments are being lost	⤒	Closed
CRUC-2095	Activity stream says "finished" instead of "uncompleted"	↑	Closed
CRUC-2085	FE PERF - fixing the people page. Based on above perf data. Goal, all cases < 10s, ideally < 2s	↓	Resolved
CRUC-1818	Double slashes appended incorrectly at the end of the SVN URL	↓	Closed
CRUC-1687	upgrade to latest jquery	↑	Closed
CRUC-1673	GenericJDBCException thrown on atlaseye	↑	Closed
CRUC-1603	Due date display is too exact.	↓	Closed
CRUC-1601	find out the css differences between the review dir tree and the existing content dir tree	↑	Closed
CRUC-1599	display settings for reviews arent intuitive and arent sticky	↑	Closed
CRUC-1597	Need Performance Benchmarks for Crucible and Fisheye	↑	Closed
CRUC-1449	Largish review beachballs on resize and when expanding FRXs	↑	Closed
CRUC-1432	Page refreshes when preferences pane is closed, even if there has been no change	▼	Closed
CRUC-1385	Add "Change Diff" button for file revision chosen from Light SCM plugins similar to files chosen from Fisheye Repositories	↑	Closed
CRUC-1334	Hot Deployment of Plugin does not work	↑	Closed
CRUC-874	Documentation for Web Fragment Plugin Points	↑	Closed

Crucible 2.0 Release Notes

⚠ Crucible 2.1 has now been released. Read the Release Notes.

30 June 2009

Atlassian presents Crucible 2.0

Crucible 2.0 adds the all-new Iterative Reviews feature, enhanced JIRA integration and a brand new user interface.

Highlights of this release:

- Iterative Reviews
- New User Interface
- Read/Unread comments
- Enhanced JIRA Integration
- Keyboard Shortcuts
- Review Activity
- External Databases and Backup
- Plus numerous improvements and bug fixes

Thank you for your interest in Crucible 2.0.

DOWNLOAD
latest version

See the documentation on Upgrading to this version.

Installing Crucible 2.0

You can now download the Crucible 2.0 from here. See the documentation on Upgrading to this version.

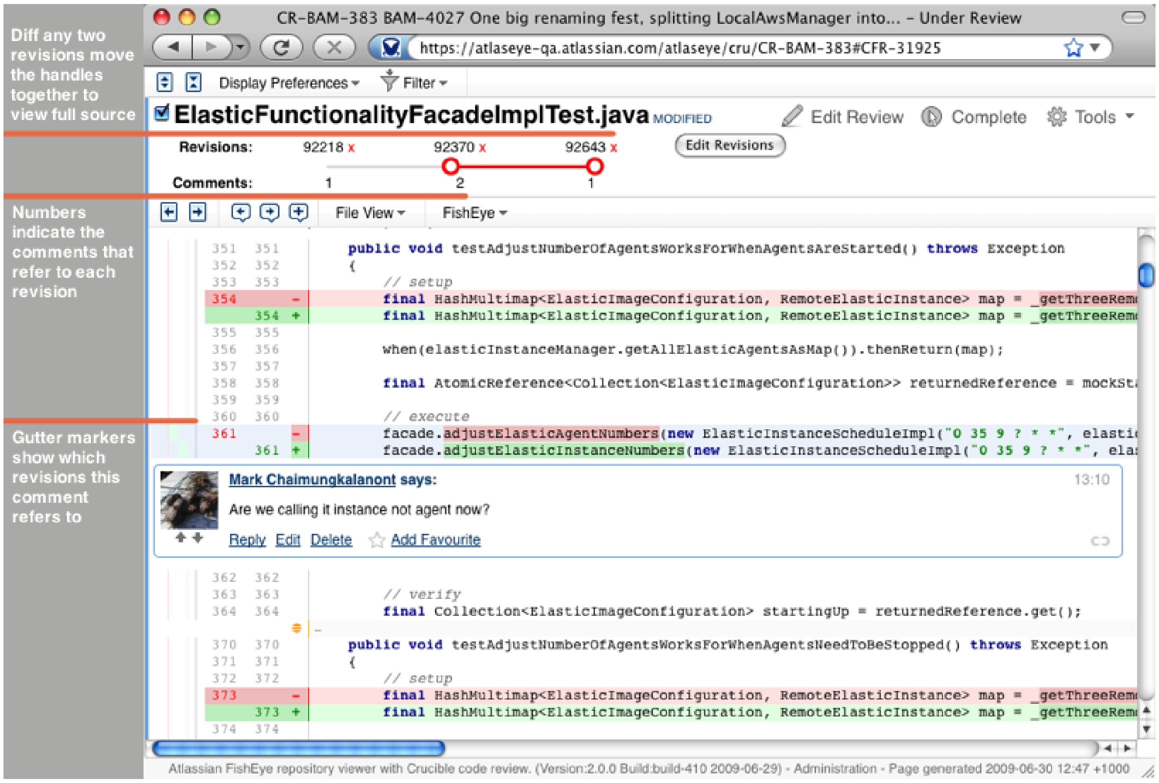
Highlights of Crucible 2.0

1

Iterative Reviews

Crucible now allows you to review several revisions of a file within one review, seamlessly switching between them. Comments are linked and relative to a specific revision. This allows you to review every change that has occurred on a code file within a given period of time and hence visualise its evolution in the context of the review.

Screenshot: Iterative Reviews



Diff any two revisions move the handles together to view full source

Numbers indicate the comments that refer to each revision

Gutter markers show which revisions this comment refers to

CR-BAM-383 BAM-4027 One big renaming fest, splitting LocalAwsManager into... - Under Review
https://atlaseye-qa.atlassian.com/atlaseye/cru/CR-BAM-383#CFR-31925
Display Preferences
Filter
ElasticFunctionalityFacadeImplTest.java MODIFIED
Edit Review
Complete
Tools
Edit Revisions
File View
FishEye

Revisions: 92218 x 92370 x 92643 x

Comments: 1 2 1

```
    public void testAdjustNumberOfAgentsWorksForWhenAgentsAreStarted() throws Exception
    {
        // setup
-       final HashMultimap<ElasticImageConfiguration, RemoteElasticInstance> map = _getThreeRem
+       final HashMultimap<ElasticImageConfiguration, RemoteElasticInstance> map = _getThreeRem

        when(elasticInstanceManager.getAllElasticAgentsAsMap()).thenReturn(map);

        final AtomicReference<Collection<ElasticImageConfiguration>> returnedReference = mockSt

        // execute
-       facade.adjustElasticAgentNumbers(new ElasticInstanceScheduleImpl("0 35 9 ? * *", elasti
+       facade.adjustElasticInstanceNumbers(new ElasticInstanceScheduleImpl("0 35 9 ? * *", ela
```

Mark Chaimungkalanont says: 13:10

Are we calling it instance not agent now?

Reply Edit Delete ☆ Add Favourite

```

        // verify
        final Collection<ElasticImageConfiguration> startingUp = returnedReference.get();
...
    public void testAdjustNumberOfAgentsWorksForWhenAgentsNeedToBeStopped() throws Exception
    {
        // setup
-       final HashMultimap<ElasticImageConfiguration, RemoteElasticInstance> map = _getThreeRem
+       final HashMultimap<ElasticImageConfiguration, RemoteElasticInstance> map = _getThreeRem

```

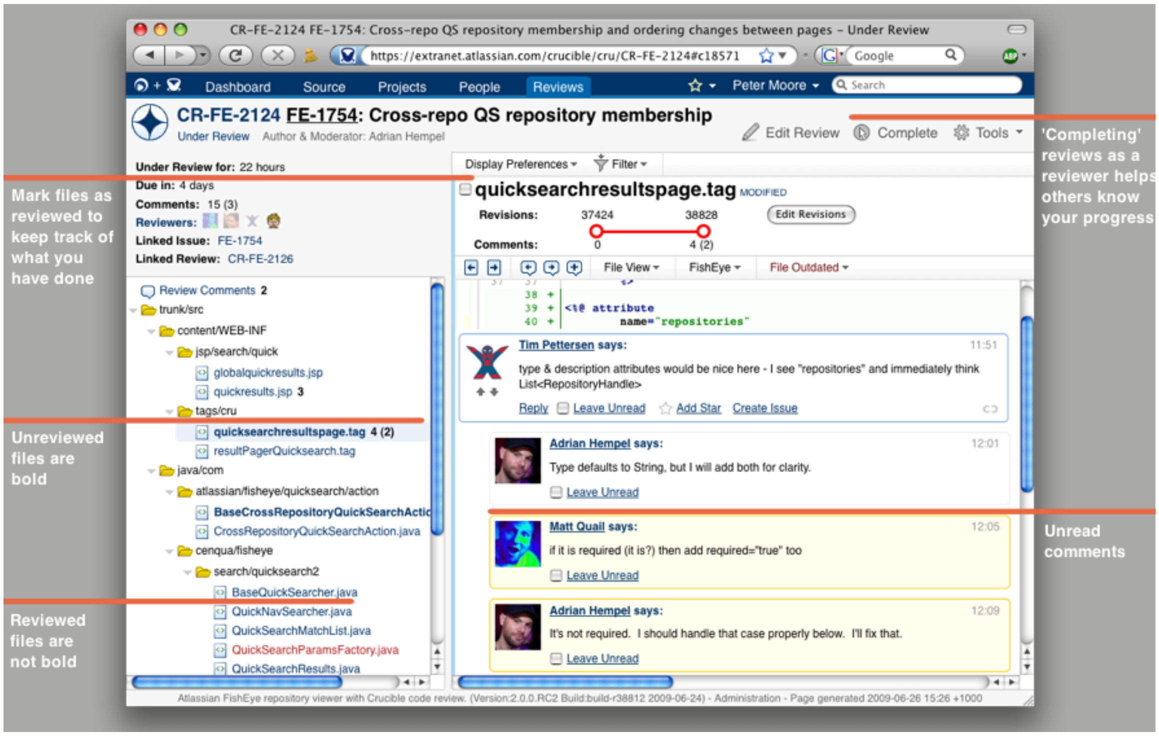
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 12:47 +1000

2

New User Interface

Taking on board wide-ranging feedback from customers, the Crucible team has completely revamped the user interface of the product, adding more views on your work and allowing you to access controls from multiple locations and allowing for different work styles. Files in review are arranged in a tree for easy navigation. New viewing modes for reviews support full screen view, side-by-side diff view and single or multiple file views.

Screenshot: Reviewed files



CR-FE-2124 FE-1754: Cross-repo QS repository membership

Under Review Author & Moderator: Adrian Hempel

Under Review for: 22 hours
Due in: 4 days
Comments: 15 (3)
Reviewers:
Linked Issue: FE-1754
Linked Review: CR-FE-2126

quicksearchresultspage.tag MODIFIED

Revisions: 37424 38828
Comments: 0 4 (2)

```
+
+ <%@ attribute
+         name="repositories"
```

Tim Pettersen says: 11:51
type & description attributes would be nice here - I see "repositories" and immediately think List<RepositoryHandle>

Adrian Hempel says: 12:01
Type defaults to String, but I will add both for clarity.
☐ Leave Unread

Matt Quail says: 12:05
if it is required (it is?) then add required="true" too
☐ Leave Unread

Adrian Hempel says: 12:09
It's not required. I should handle that case properly below. I'll fix that.
☐ Leave Unread

Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0.RC2 Build:build-r38812 2009-06-24) - Administration - Page generated 2009-06-26 15:26 +1000

Mark files as reviewed to keep track of what you have done

Unreviewed files are bold

Reviewed files are not bold

'Completing' reviews as a reviewer helps others know your progress

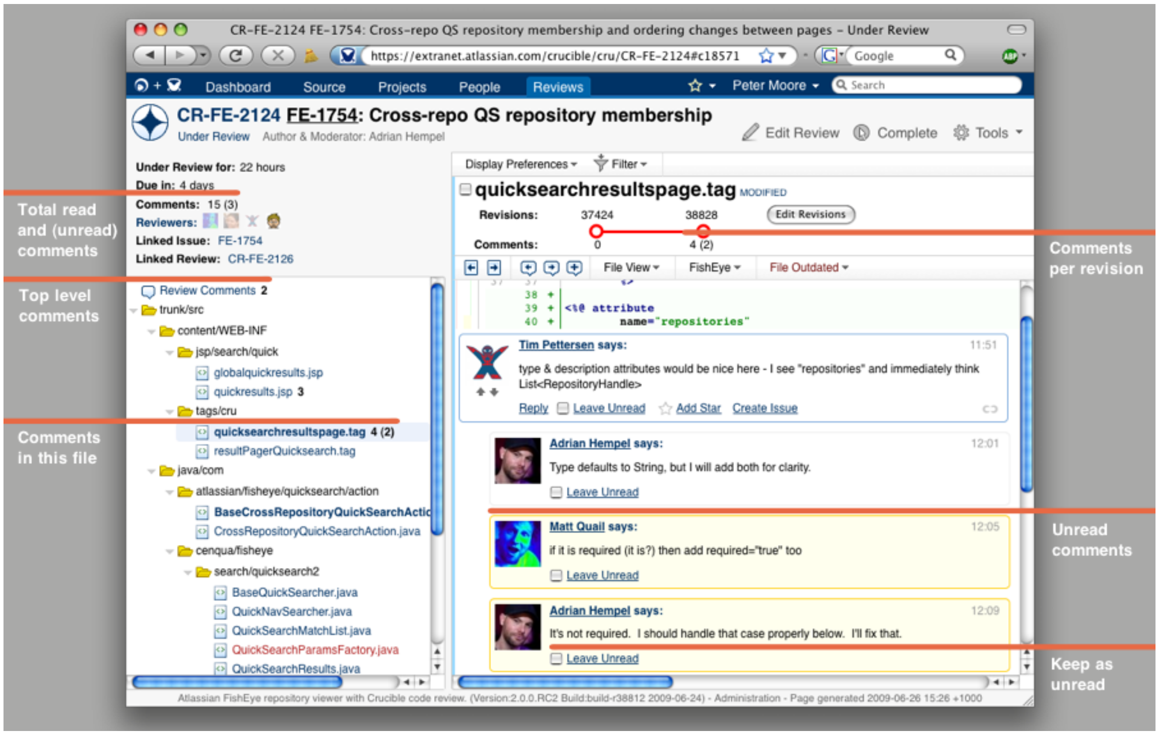
Unread comments

3

Read/Unread comments

As you move around a review, Crucible keeps track of which comments you've seen and marks them as read. When you see a comment that you want to come back to, check the 'leave as unread' box so you don't forget it. Unlike an email thread, new comments are rarely at the bottom. That's why its especially useful to filter and highlight just the new comments when you come back to a review that's underway.

Screenshot: Unread comments


CR-FE-2124 FE-1754: Cross-repo QS repository membership and ordering changes between pages – Under Review
https://extranet.atlassian.com/crucible/cru/CR-FE-2124#c18571
Google
Dashboard
Source
Projects
People
Reviews
Peter Moore
Search
CR-FE-2124 FE-1754: Cross-repo QS repository membership
Under Review
Author & Moderator: Adrian Hempel
Edit Review
Complete
Tools
Under Review for: 22 hours
Due in: 4 days
Comments: 15 (3)
Reviewers:
Linked Issue: FE-1754
Linked Review: CR-FE-2126
Display Preferences
Filter
quicksearchresultspage.tag MODIFIED
Revisions: 37424 38828
Edit Revisions
Comments: 0 4 (2)
File View
FishEye
File Outdated
Review Comments 2
trunk/src
content/WEB-INF
jsp/search/quick
globalquickresults.jsp
quickresults.jsp 3
tags/cru
quicksearchresultspage.tag 4 (2)
resultPagerQuicksearch.tag
java/com
atlassian/fisheye/quicksearch/action
BaseCrossRepositoryQuickSearchActio
CrossRepositoryQuickSearchAction.java
cenqua/fisheye
search/quicksearch2
BaseQuickSearcher.java
QuickNavSearcher.java
QuickSearchMatchList.java
QuickSearchParamsFactory.java
QuickSearchResults.java
Reply
Leave Unread
Add Star
Create Issue
Leave Unread
Leave Unread
Leave Unread
Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0.RC2 Build:build-r38812 2009-06-24) - Administration - Page generated 2009-06-26 15:26 +1000
Total read and (unread) comments
Top level comments
Comments in this file
Comments per revision
Unread comments
Keep as unread

```
38 +
39 +  <%@ attribute
40 +          name="repositories"
```

Tim Pettersen says: 11:51
type & description attributes would be nice here - I see "repositories" and immediately think List<RepositoryHandle>

Adrian Hempel says: 12:01
Type defaults to String, but I will add both for clarity.

Matt Quail says: 12:05
if it is required (it is?) then add required="true" too

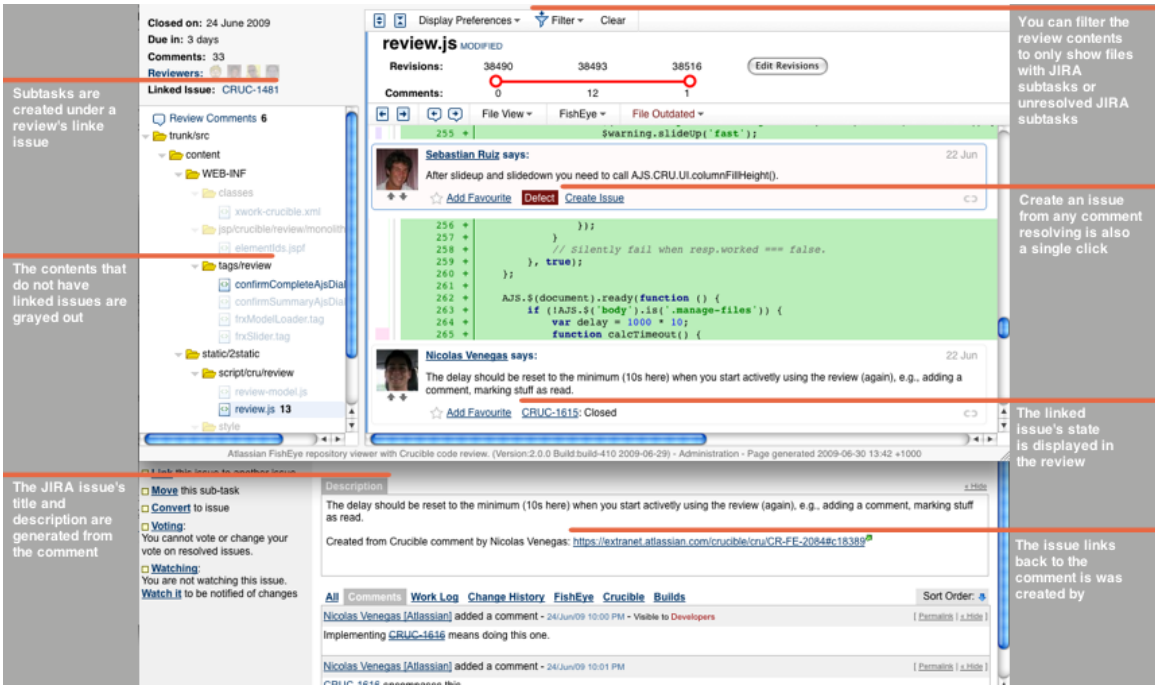
Adrian Hempel says: 12:09
It's not required. I should handle that case properly below. I'll fix that.

4

Enhanced JIRA Integration

Crucible now has better JIRA integration, allowing you to see regular JIRA updates in your Crucible dashboard and create JIRA sub-tasks inline, without leaving the Crucible interface. You can still click on issue names to visit the JIRA instance they belong to, also. See instructions for JIRA configuration.

Screenshot: Enhanced JIRA Integration



Closed on: 24 June 2009
Due in: 3 days
Comments: 33
Reviewers:
Linked Issue: CRUC-1481

review.js MODIFIED

Revisions: 38490 38493 38516
Comments: 0 12 1

```
255 +                      $warning.slideUp('fast');
```

Sebastian Ruiz says: 22 Jun
After slideup and slidedown you need to call AJS.CRU.UI.columnFillHeight().

```
256 +                    });
257 +                }
258 +                // Silently fail when resp.worked === false.
259 +            }, true);
260 +        };
261 +
262 +        AJS.$(document).ready(function () {
263 +            if (!AJS.$('body').is('.manage-files')) {
264 +                var delay = 1000 * 10;
265 +                function calcTimeout() {
```

Nicolas Venegas says: 22 Jun
The delay should be reset to the minimum (10s here) when you start activetly using the review (again), e.g., adding a comment, marking stuff as read.

CRUC-1615: Closed

Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0 Build:build-410 2009-06-29) - Administration - Page generated 2009-06-30 13:42 +1000

□ Link this issue to another issue
□ Move this sub-task
□ Convert to issue
□ Voting:
You cannot vote or change your vote on resolved issues.
□ Watching:
You are not watching this issue.
Watch it to be notified of changes

The delay should be reset to the minimum (10s here) when you start activetly using the review (again), e.g., adding a comment, marking stuff as read.

Created from Crucible comment by Nicolas Venegas: https://extranet.atlassian.com/crucible/cru/CR-FE-2084#c18389

Nicolas Venegas [Atlassian] added a comment - 24/Jun/09 10:00 PM - Visible to Developers
Implementing CRUC-1616 means doing this one.

Nicolas Venegas [Atlassian] added a comment - 24/Jun/09 10:01 PM
CRUC-1616 encompasses this

Subtasks are created under a review's linke issue

The contents that do not have linked issues are grayed out

The JIRA issue's title and description are generated from the comment

You can filter the review contents to only show files with JIRA subtasks or unresolved JIRA subtasks

Create an issue from any comment resolving is also a single click

The linked issue's state is displayed in the review

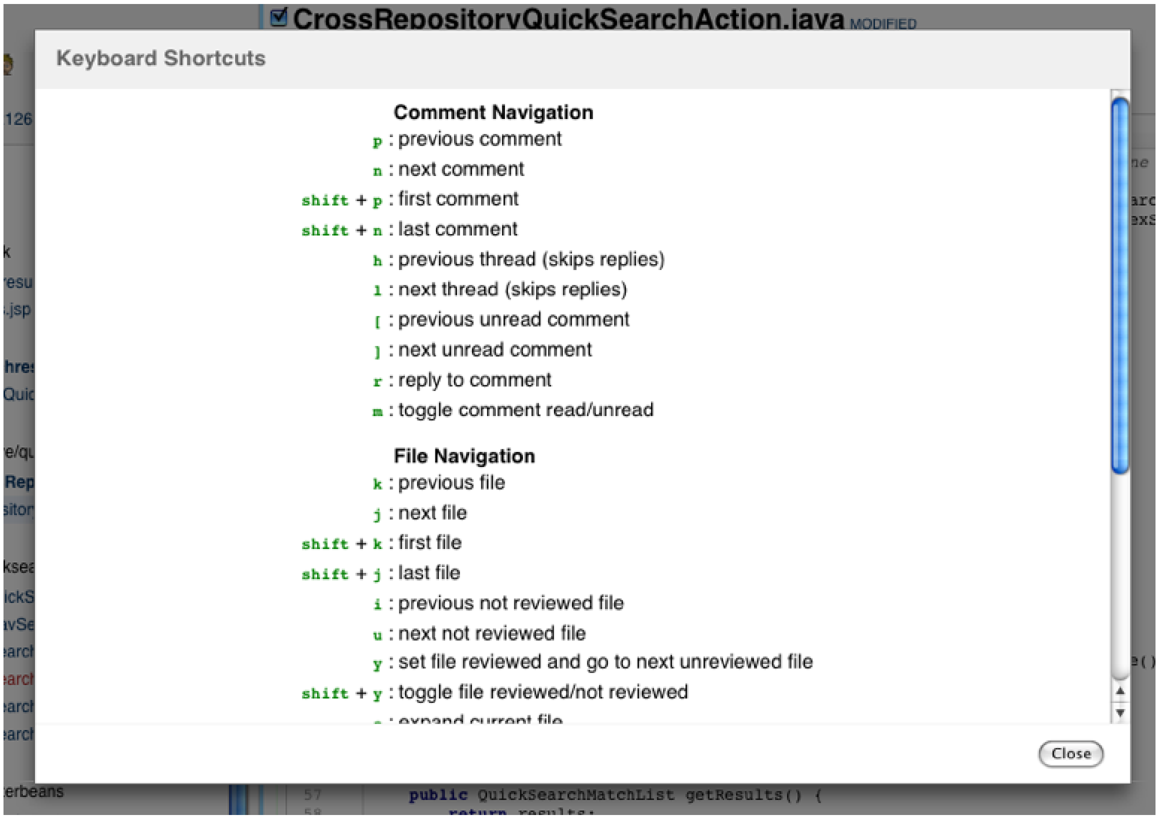
The issue links back to the comment is was created by

5

Keyboard Shortcuts

Crucible now has keyboard shortcuts for navigating around your reviews efficiently. No more repetitive strain injury from that mouse wheel.

Screenshot: Keyboard shortcuts



Comment Navigation

- `p` : previous comment
- `n` : next comment
- `shift` + `p` : first comment
- `shift` + `n` : last comment
- `h` : previous thread (skips replies)
- `l` : next thread (skips replies)
- `[` : previous unread comment
- `]` : next unread comment
- `r` : reply to comment
- `m` : toggle comment read/unread

File Navigation

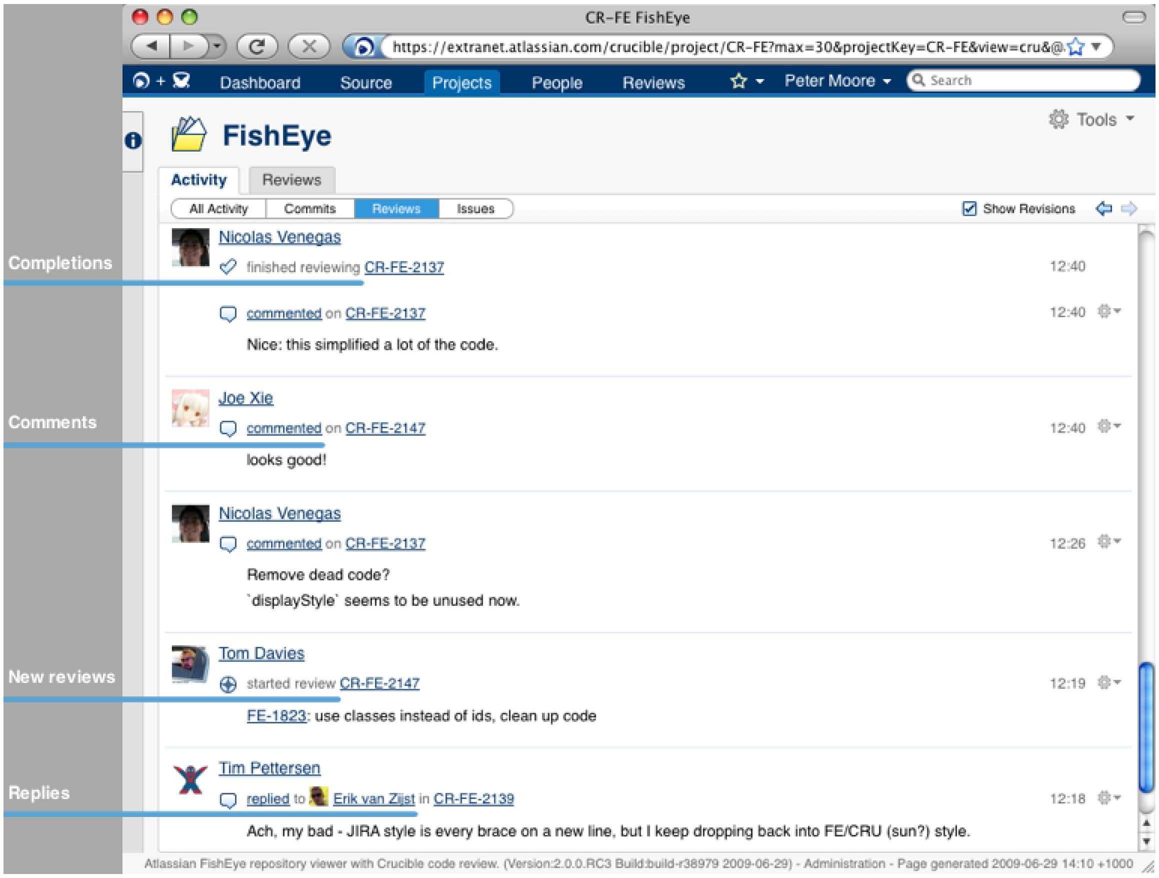
- `k` : previous file
- `j` : next file
- `shift` + `k` : first file
- `shift` + `j` : last file
- `i` : previous not reviewed file
- `u` : next not reviewed file
- `y` : set file reviewed and go to next unreviewed file
- `shift` + `y` : toggle file reviewed/not reviewed
- : expand current file

6

Review Activity

All the activity that happens in Crucible is available as an activity stream. Streams can be accessed per project as well as a personal stream that includes the activity from people, projects, reviews and even comments you favorite or are involved in.

Screenshot: Project review activity stream



Nicolas Venegas
finished reviewing CR-FE-2137 12:40

commented on CR-FE-2137 12:40
Nice: this simplified a lot of the code.

Joe Xie
commented on CR-FE-2147 12:40
looks good!

Nicolas Venegas
commented on CR-FE-2137 12:26
Remove dead code?
`displayStyle` seems to be unused now.

Tom Davies
started review CR-FE-2147 12:19
FE-1823: use classes instead of ids, clean up code

Tim Pettersen
replied to Erik van Zijst in CR-FE-2139 12:18
Ach, my bad - JIRA style is every brace on a new line, but I keep dropping back into FE/CRU (sun?) style.

Atlassian FishEye repository viewer with Crucible code review. (Version:2.0.0.RC3 Build:build-r38979 2009-06-29) - Administration - Page generated 2009-06-29 14:10 +1000

7

External Databases and Backup

Crucible now supports MySQL and Postgress in addition to the embeded HSQL database. Backup and restore capabilities have also been greatly enhanced.

8

Plus numerous improvements and bug fixes

Visit our issue tracker to see the full list of improvements and bug fixes.

Crucible 2.0 Changelog

On this page:

- From 2.0.5 to 2.0.6
- From 2.0.4 to 2.0.5
- From 2.0.3 to 2.0.4
- From 2.0.2 to 2.0.3
- From 2.0.1 to 2.0.2
- From 2.0 to 2.0.1

- From 2.0 Beta3 to 2.0
- From 2.0 Beta2 to 2.0 Beta3
- From 1.6.6 to 2.0 Beta2
-

From 2.0.5 to 2.0.6

8th October 2009

This is a bugfix release.

This release fixes a bug that affected Crucible-only installations.

From 2.0.4 to 2.0.5

6th October 2009

This is a bugfix and improvement release.

The following issues are addressed by this release:

JIRA Issues (9 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-2317	Linking to JIRA issues doesn't work when you customise the JIRA issue key format	↑	✓ Closed
CRUC-2295	"Open in Crucible" link always leads to EAC/CRU	⇑	✓ Closed
CRUC-2214	Transactions don't nest	⛔	✓ Closed
CRUC-2203	Allow detailed logging of requests to SCM plugins	↓	✓ Closed
CRUC-2135	Upgrade to Licensing 2.0	↓	➔ Resolved
CRUC-1856	Page title is doubly escaped	↓	✓ Closed
CRUC-1742	NPE when trying to load FRX	↑	✓ Closed
CRUC-1734	Doesn't pick up the changes made for the "Statement of Objectives" field in IE 7	⇑	✓ Closed
CRUC-269	ctempobj not well-handled (+S flag on perforce files)	▼	✓ Closed

From 2.0.3 to 2.0.4

8th September 2009

This is a bugfix and improvement release.

- **Crucible Light SCM Issues with Deleted Files**: See the JIRA issue for details.

The following issues are addressed by this release:

JIRA Issues (14 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-2160	add some reviewers in the dialog, hit 'start review' --> "This review has no reviewers"	⇑	✓ Closed
CRUC-2128	NPE when trying to create a review	↓	✓ Closed

CRUC-2112	NullPointerException when posting/updating comments through the SPI	↑	✓ Closed
CRUC-2105	"java.lang.RuntimeException: org.tigris.subversion.javahl.ClientException: svn: File not found:" thrown when adding a changeset that contains deleted files in light svn plugin	↑	✓ Closed
CRUC-2104	Crucible is still creating ghost draft comments	❗	✓ Closed
CRUC-2101	binary deleted revisions in reviews are being diffed as text in Cru	↓	✓ Closed
CRUC-1952	Iterative reviews with uploaded files is broken	❗	✓ Closed
CRUC-1919	NonUniqueResultException: query did not return a unique result	↑	✓ Closed
CRUC-1907	Warn before dropping database in restore	❗	✓ Closed
CRUC-1402	NPE When using eclipse plugin	⇑	✓ Closed
CRUC-1256	Crucible does not handle incomplete record data (via REST API) properly	↑	✓ Closed
CRUC-859	NPE when loading full history for file in lightSVN	↑	✓ Closed
CRUC-794	SVN LightSCM plugin tries to get revision which don't exist when copies are made	↑	✓ Closed
CRUC-761	Add warning that if user does not enter fisheye license they will not have access to fisheye capabilites	⇑	✓ Closed

From 2.0.2 to 2.0.3

18th August 2009

This is a bugfix release which includes the following issues:

JIRA Issues (14 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-2051	ie7: can't view draft reviews	⇑	✓ Closed
CRUC-2050	java.util.ConcurrentModificationException thrown during when removing participants	⇑	✓ Closed
CRUC-2015	RSS should have absolute URLs, not relative	↓	✓ Closed
CRUC-1940	Frequent Error When Starting A Review : error.unauthorized.stateChange.description	⇑	✓ Closed
CRUC-1927	The copied reviewed file should be shown in orange color	↓	✓ Closed

CRUC-1911	Case sensitivity in username	⬇	Closed
CRUC-1906	[git] annotation view skips chunks	❗	Closed
CRUC-1899	Review updated ajax calls should be chained	⬆	Closed
CRUC-1845	Cannot delete abandoned review	⬇	Closed
CRUC-1833	NPE thrown in getIcDOFromComment	⬆	Closed
CRUC-1828	Make the number of connections to a db easily configurable	⬆	Closed
CRUC-1814	Review Deletion is not working	⬆	Closed
CRUC-1094	Strange problems while trying to use SCM plugin for SVN with multiple repositories	⬆	Closed
CRUC-854	Author Mapping doesn't always work	⬆	Closed

From 2.0.1 to 2.0.2

24th July 2009

This is a bugfix release which includes the following issues:

JIRA Issues (15 issues)

Key	Summary	Priority	Status
CRUC-1886	UTF-8 encoding of forms not being handled	⬆	Closed
CRUC-1831	various misc wording improvements in Crucible	⬆	Closed
CRUC-1791	when creating a review from a changeset in the activity stream, don't add the revision number to the objectives	⬆	Resolved
CRUC-1781	[Security Vulnerability] - REST API is working even if Remote API is OFF	⬆	Closed
CRUC-1780	Invalid filter parameter could lead to security issues	⬆	Closed
CRUC-1779	stored paths can be invalid when creating reviews at the same time	⬆	Closed
CRUC-1771	Switching between single file and multi-file should be hoisted to the tool bar	⬆	Resolved
CRUC-1770	Switching to fullscreen should be hoisted to an icon on the tool bar	⬆	Resolved

CRUC-1747	NPE in atlaseye for lauchlan	⬆	Closed
CRUC-1746	Adding revisions iteratively breaks in litesvn	⇪	Resolved
CRUC-1739	change 'summarise' to 'summarize' on review age chart	⬆	Closed
CRUC-1729	bad wording of comment activity tool menu 'add favourites'	⬆	Closed
CRUC-1720	Error logging in if the project or repository added as a favourite has been deleted	⬆	Closed
CRUC-1715	links to review inline comments dont work properly	⬆	Closed
CRUC-1384	Review comments and replies added via REST API do not trigger e-mail notifications	⇪	Closed

From 2.0 to 2.0.1

14th July 2009

This is a bugfix release which includes the following issues:

JIRA Issues (29 issues)

Key	**Summary**	**Priority**	**Status**
CRUC-1743	Select Reviewers textbox on the Edit Review->Review Details screen will not add user to a review if the username contains "." or"@" symbols	⬇	Closed
CRUC-1716	[Regression] Suggested reviewers are not appearing in the list of reviewers when they are added from the pop-up	⬆	Closed
CRUC-1703	Arrow "Previous Comment Thread" in Crucible Review throws script exception "CRUCOMMENT is undefined"	⬇	Closed
CRUC-1701	Fail to add user as reviewer in Crucible if the user's username contains dot (.)	⬇	Closed
CRUC-1691	Poll for state changes	⬆	Closed
CRUC-1690	User and Crucible links don't have hovers in Review Comment search results	⬆	Closed
CRUC-1686	Editing Permission Schemes throws javax.el.PropertyNotFoundException	❗	Closed
CRUC-1683	Non-empty folder doesn't have comment count in tree	⬆	Closed
CRUC-1682	Opening a review with a 2mb patch causes the review to take 1 minute to open	⬆	Closed
CRUC-1676	Empty folders are incorrectly ordered	⬆	Closed
CRUC-1672	"Create Review" on browse page does not add-to-review, but always creates a new review	⬆	Closed

CRUC-1671	Empty folder doesn't have folder icon in tree	↑	Closed
CRUC-1670	FRXs with the same path are collapsed in the navigation tree	↑	Closed
CRUC-1669	Patch reviews shouldn't allow deletion of frxrevisions	↑	Closed
CRUC-1664	Backup admin page includes all items when de-selecting all	↑	Closed
CRUC-1657	draft comments are shown as unread from the reviews list	↑	Closed
CRUC-1655	inline comments need revisiting in side-by-side diff mode	↑	Closed
CRUC-1654	Restore user to previous position in review when reloading from a review update	↑	Closed
CRUC-1653	Show "reviews to do" table when closing a review	↑	Closed
CRUC-1650	edit revisions allows you to try and add a revision already in the review, then complains	↑	Closed
CRUC-1638	terminology - should use "unreviewed" or "not reviewed"	↓	Closed
CRUC-1636	inconsistent state names	↓	Closed
CRUC-1630	Grey out menu items when not applicable	↑	Closed
CRUC-1619	due dates arent updated when new revisions are added	↑	Closed
CRUC-1606	Sort indicator on "Age" column of reviews dashboard points in the wrong direction	↓	Closed
CRUC-1598	Add linked reviews to the review sidebar.	↑	Closed
CRUC-1579	Editing a User in crucible standalone will render an sterisk next to their name	↑	Closed
CRUC-1567	Comment hovers only on text in IE8	↑	Closed
CRUC-1443	Safari issue - Tools dropdown causes 'invisible scrollbars'	↓	Closed

From 2.0 Beta3 to 2.0

30th June 2009

Full list of issues in this release:

JIRA Issues (200 issues)			

Key	Summary	Priority	Status
CRUC-1749	XSS issue in filter parameter	↑	✓ Closed
CRUC-1661	File marked as unreviewed for defect author	↑	✓ Closed
CRUC-1656	RC3 - NPE when trying to create first project on empty CRU install	⇑	✓ Closed
CRUC-1651	clicking [add] branch in SVN admin symbolic screen causes page-pop and 404	↑	✓ Closed
CRUC-1649	toggling Display Preferences > Show Source in a review does nothing	↑	✓ Closed
CRUC-1644	Some display preferences don't update when clicked	↑	✓ Closed
CRUC-1641	error mapping JIRA servers	↑	✓ Closed
CRUC-1623	Admin: Default reviewers are not re-saved	↑	✓ Closed
CRUC-1616	Tune polling period.	❗	✓ Closed
CRUC-1615	Reset the delay when activetly using the review	❗	✓ Closed
CRUC-1612	Move the warning to a Gmail-style overlay	❗	✓ Closed
CRUC-1611	Commenting doesn't work in Safari 4	↑	✓ Closed
CRUC-1609	CAC documentation for iterative review info from CRUC-1536	↑	✓ Closed
CRUC-1481	Warning dialogs and polling when new stuff added since last refresh	↑	✓ Closed
CRUC-1445	STUPID: old style error page when trying to access fisheye pages	▼	✓ Closed
CRUC-1442	Crucible Standalone - Logout screen has FishEye icon & link	▼	✓ Closed
CRUC-1413	Adding all changesets for new files with multiple commits should show file as "Added", not "Modified"	↓	✓ Closed
CRUC-1412	"Earlier changeset" and "Later changeset" links should be grayed out when earliest/latest already listed.	↓	✓ Closed
CRUC-1408	Images in review blend in to crucible	↓	✓ Closed
CRUC-1404	Review throws 500 error NPE	↑	✓ Closed

CRUC-1403	Rename "crucibledb" to "sql"	⬆	Closed
CRUC-1399	Default Reviewer for project is not exposed through the SPI	⬆	Closed
CRUC-1398	overdue but review has not been started. Setting a due date in the past causes this.	⬆	Closed
CRUC-1393	Refactor Email Comments Page into a Dialog	⬆	Resolved
CRUC-1392	fix remove file behaviour of current content panel	⬆	Closed
CRUC-1391	Use newDirTreeBox in the files pane	⬆	Closed
CRUC-1390	Fix Uploads	⬆	Closed
CRUC-1388	Missing css files on Send Email page / Layout needs love	⬆	Closed
CRUC-1386	fix suggest reviewer ui	⬆	Closed
CRUC-1383	suggest reviewers - turn blame calculation back on	⬆	Closed
CRUC-1382	defect metrics charts are broken	⬆	Closed
CRUC-1381	.png file not detected as binary	⬆	Closed
CRUC-1380	Email Review button from the dashboard doesnt work / Email review in tols menu gives a 404 - permaid is undefined	⬆	Closed
CRUC-1379	clicking HELP opens help twice / Help in Tools menu will open a new window and change your window	⬆	Closed
CRUC-1378	page load error "The requested URL /atlaseye/static/kidi0j/2static/script/fe/fisheye-profile.js was not found on this server."	⬆	Closed
CRUC-1377	wrong review actions (comments.txt and Email Review) are visible	⬆	Closed
CRUC-1376	we need to handle multiple errors in a smarter way	⬆	Closed
CRUC-1375	There is no jira quick link in the edit review dialog	⬆	Closed
CRUC-1374	Review info popup on the dashboard shouldn't show when clicking on a link such as the review permaId	⬆	Closed
CRUC-1373	Page load error "The requested URL /atlaseye/static/kidi0j/2static/images/sprite_arrows.png was not found on this server"	⬆	Closed

CRUC-1371	External dialog 'View ...' links need to redirect to review and trigger filter	⬆	Closed
CRUC-1370	Show current content panel in manage files	⬆	Closed
CRUC-1369	make the side filters like the fisheye tabs	⬆	Closed
CRUC-1368	bring back the custom filter	⬆	Closed
CRUC-1367	make the pages consistent	⬆	Closed
CRUC-1366	Add warnings for incomplete FRXs and unresolved JIRAs to dialogs	⬆	Closed
CRUC-1365	Move review summary form into a dialog.	⬆	Closed
CRUC-1364	Move creation-time review suggestions into a dialog	⬆	Closed
CRUC-1363	Convert project selection screen to a dialog	⬆	Closed
CRUC-1362	Backup broke Crucible Blitz	⬆	Closed
CRUC-1361	General Issues	⬆	Resolved
CRUC-1360	Manage Files	⬆	Resolved
CRUC-1359	Edit review form	⬆	Resolved
CRUC-1358	Interstitials/dialogs	⬆	Resolved
CRUC-1357	Dashboard Fixes	⬆	Resolved
CRUC-1356	Remove driver url parameter from Restore and use new DBInfo properties	⬆	Closed
CRUC-1354	Refactor programatic frx filtering to not invoke click events	⬆	Resolved
CRUC-1351	Add path and base filename to backup UI	⬆	Resolved
CRUC-1345	Extend QuartzManager to support persistent configuration data per job	⬆	Closed
CRUC-1344	Rest Api returns the wrong URL for download Patch	⬆	Closed
CRUC-1332	Going to full context doesn't work	⬆	Closed

CRUC-1330	Put custom quartz job scheduling in a separate tab	↑	Closed
CRUC-1329	Dynamically update general comment counts	↑	Closed
CRUC-1328	Add general comments to the tree	↑	Closed
CRUC-1327	Add the BackupItem selection to the scheduled backup	↑	Closed
CRUC-1325	Add file type css place holders	↑	Closed
CRUC-1324	FishEye only indicate Crucible review for a file with latest revision	↓	Closed
CRUC-1319	Update doco for admin backup page	↑	Resolved
CRUC-1317	Dynamically update comment counts	↑	Closed
CRUC-1316	Add comment counts to the frx tree	↑	Closed
CRUC-1305	Springify Backup Manager	↑	Closed
CRUC-1303	Write unit test	↑	Closed
CRUC-1294	Check start/stop no longer needed in restore	↑	Closed
CRUC-1293	Render maintenance page for non-admin pages when a backup is being made	↑	Closed
CRUC-1291	Add "Change Diff" dropdown to manipulate both fromRev and toRev independently	↑	Resolved
CRUC-1290	Collapse common directories	↑	Closed
CRUC-1289	Apply tree html structure and css styling to list	↑	Closed
CRUC-1288	Treeify FRX list pane	↑	Resolved
CRUC-1286	add other filters	↑	Closed
CRUC-1285	add filter - with comments	↑	Closed
CRUC-1284	add UI for dropdown and expand/collapse	↑	Closed
CRUC-1283	Drop-down filter for frx tree	↑	Resolved

CRUC-1275	Convert stored dates to long	⬆	Closed
CRUC-1274	Create FRX anchors and adjust the browser location as you navigate	⬆	Resolved
CRUC-1273	FRXs should start expanded and maintain position when loading frxs above	⬆	Closed
CRUC-1272	Told that other participants have drafts, but also that all other participants have zero drafts	⬆	Closed
CRUC-1271	Move edit review box into AJS.Dialog	⬆	Closed
CRUC-1270	Integrate the summarise and close pages	⬆	Closed
CRUC-1254	Rearrange content	⬆	Closed
CRUC-1253	Review meta information sub pane	⬆	Closed
CRUC-1252	Move manage files tabs into AJS.Dialog	⬆	Closed
CRUC-1251	scroll frx pane when clicking on frx list item	⬆	Closed
CRUC-1250	create floating frx info subpane which updates on scroll	⬆	Closed
CRUC-1248	Modify overview header	⬆	Closed
CRUC-1247	3 Pane UI	⬆	Resolved
CRUC-1242	Delete the old backup mechanism	⬆	Resolved
CRUC-1241	Rewire the quartz job	⬆	Closed
CRUC-1235	Revision slider hardening/tweaking	⬆	Resolved
CRUC-1233	Wrap option	⬆	Closed
CRUC-1232	Delete comment refactor	⬆	Closed
CRUC-1215	Quartz backup job must invoke the new xml-based backup	⬆	Resolved
CRUC-1208	reorganise contents of static directory	⬆	Closed
CRUC-1206	add new preference for new frxrevisions	⬆	Closed

CRUC-1205	add new notification for adding frxRevisions	⇧	Closed
CRUC-1201	displaying frxs does not take into account showAsDiff flag	⇪	Closed
CRUC-1179	Sort FRX - alpha server side sort	⇧	Resolved
CRUC-1174	Add banner warning "you have unresolved jiras" to page	⇧	Closed
CRUC-1173	Add resolve all JIRAs button & action to summarize page	⇧	Closed
CRUC-1171	XML-RPC bollocks	⇧	Closed
CRUC-1170	Delete a defect's linked JIRA subtask when deleting the defect	⇧	Resolved
CRUC-1169	ui & action	⇧	Closed
CRUC-1168	XML-RPC bollocks	⇧	Closed
CRUC-1167	Resolve a defect's linked JIRA subtask from Crucible	⇧	Resolved
CRUC-1166	XML-RPC bollocks	⇧	Closed
CRUC-1165	UI display of jira link and status + post error creating	⇧	Closed
CRUC-1164	schema	⇧	Closed
CRUC-1163	Automatically add defects as subtasks to JIRA & link to them	⇧	Resolved
CRUC-1162	guessing + jira validation	⇧	Closed
CRUC-1161	edit details ui + actions	⇧	Closed
CRUC-1160	schema	⇧	Closed
CRUC-1159	Link a review to a single JIRA and enable auto defect creation	⇧	Resolved
CRUC-1158	just do it	⇧	Closed
CRUC-1157	Configure a JIRA instance for use by Crucible	⇧	Resolved
CRUC-1156	admin UI + create review		

		⬆	Closed
CRUC-1155	schema	⬆	Closed
CRUC-1154	Configure a default review duration for a project in week days so the due date can be pre-populated	⬆	Resolved
CRUC-1153	highlighting overdue	⬆	Closed
CRUC-1152	sorting in the lists	⬆	Closed
CRUC-1151	Order any list of reviews by due date & highlight any that are overdue	⬆	Resolved
CRUC-1150	due date field in UI	⬆	Closed
CRUC-1149	schema	⬆	Closed
CRUC-1148	Set or change a due date on a review	⬆	Resolved
CRUC-1143	just do it	⬆	Closed
CRUC-1142	Combine diff and annotation views	⬆	Resolved
CRUC-1141	just do it	⬆	Closed
CRUC-1140	show the reviews you can't update without buttons	⬆	Resolved
CRUC-1139	Author colours	⬆	Closed
CRUC-1138	Invisible markers (between lines)	⬆	Closed
CRUC-1137	Mouseover	⬆	Closed
CRUC-1136	Improve comment marker behaviour (tetris bar)	⬆	Resolved
CRUC-1134	replace yahoo shite if we can	⬆	Closed
CRUC-1133	event binding	⬆	Closed
CRUC-1132	error handling	⬆	Closed
CRUC-1131	namespacing & multiple files	⬆	Closed

CRUC-1130	remove prototype	↑	Closed
CRUC-1129	client side js model	↑	Closed
CRUC-1128	refactor commentator.js and friends	↑	Resolved
CRUC-1127	just do it	↑	Closed
CRUC-1126	handle modifier keypress	↑	Resolved
CRUC-1125	just do it	↑	Closed
CRUC-1124	Popup a keyboard shortcut help overlay when i press '?'	↑	Resolved
CRUC-1123	add user pref	↑	Closed
CRUC-1122	add complete status resetter to the defect notification	↑	Closed
CRUC-1121	Reset my complete status when new defects are added to a review	↑	Resolved
CRUC-1119	Tweak comment scroll tracker behaviour	↑	Resolved
CRUC-1118	Improve revision slider behaviour	↑	Resolved
CRUC-1117	add unread comment counts per revision	↑	Closed
CRUC-1114	fix rendering/script issue in IE	↑	Closed
CRUC-1113	make the slider handles move in unison	↑	Closed
CRUC-1108	Mark comment read when it hits top of the viewport	↑	Closed
CRUC-1105	adding to review doesnt work properly when you add the first file after a branch	●	Closed
CRUC-1097	Remove the use of half closed TCP connections	↑	Resolved
CRUC-1096	Change SPI to support multiple revisions per FRX	↑	Closed
CRUC-1092	Show user's lastname and firstname when selecting a reviewer	↓	Closed
CRUC-1089	Crucible Project should move from Global Settings to own Section	↓	Closed

CRUC-1083	render author + participant results	⬆	Resolved
CRUC-1062	api changes for multi-revisions and revision comments	⬆	Closed
CRUC-1060	Comment-counts displayed are not taking replies into consideration	⬆	Closed
CRUC-1057	Expand Review Summary Email	⬆	Closed
CRUC-1048	Change the admin backup page to use the new backup/restore implementation	⬆	Resolved
CRUC-1044	When upgrading from fisheye to crucible, the UI does not allow you to enter the Crucible license	⬆	Closed
CRUC-977	issues comments added on changed and unchanged code don't show up none or less context	⬆	Closed
CRUC-971	CRU Defect-> JIRA Subtask creation	⬆	Closed
CRUC-940	Comments made on a line not in the diff is not shown	⬆	Closed
CRUC-927	Add a warning in the product that re-sync in large LDAP servers can cause performance issue	⬆	Closed
CRUC-922	Admin screens with config.xml write and create/migrate workflow	⬆	Closed
CRUC-921	Set up maintenance Mode	⬆	Closed
CRUC-920	set up tests to compare upgrades of each database type	⬆	Closed
CRUC-918	Write tests to determine/compare expected results from database tables	⬆	Closed
CRUC-915	Usage Documentation for web-based backup/restore feature	⬆	Closed
CRUC-912	Integration Tests	⬆	Closed
CRUC-910	Admin Backup Page	⬆	Closed
CRUC-909	Backup Status Page	⬆	Closed
CRUC-907	Online Backup/Restore	⬆	Closed
CRUC-882	REST API return other people's drafts	⬆	Closed
CRUC-861	Allow soft wrap on diff and annotated display	⬆	Closed

CRUC-860	Autosave has issues	↑	Closed
CRUC-858	"Lines are missing" divider graphics are no longer there	↓	Closed
CRUC-845	Describe this in the CAC docs.	↑	Closed
CRUC-825	remove abandon button from edit details section	↑	Closed
CRUC-824	If we haven't already we should automatically create a backup of our CRUDB (and other configuration files) before upgrading	⇑	Closed
CRUC-823	Add an Expand Unchecked Files option	↑	Closed
CRUC-813	Side-by-side diff mode is too wide	↑	Closed
CRUC-770	Support file upload via REST	↑	Closed
CRUC-745	Setup page doesn't warn of invalid fe licence when valid cru licence is provided	↓	Closed
CRUC-741	Change moderator via REST	↑	Closed
CRUC-734	"Side-by-side" Diffs option does not update upon refreshing the Firefox browser	↓	Closed
CRUC-603	Support Anonymous Users Properly	↑	Closed
CRUC-581	Include Light SCM Repos in information returned by REST API	↑	Closed
CRUC-537	Summary Report & Searching by File	↓	Closed

From 2.0 Beta2 to 2.0 Beta3

5th June 2009

Full list of issues in this release:

JIRA Issues (35 issues)

Key	Summary	Priority	Status
CRUC-1524	review actions from hovers sometimes aren't bound and dont work.	↑	Closed
CRUC-1515	File time stamps are not preserved in the backup	↑	Closed
CRUC-1513	Permalinks to hidden comments are broken	↑	Closed

CRUC-1512	Crucible user preferences should be saved automatically when changed	⬆	Closed
CRUC-1508	Pressing 'y' in fullscreen mode skips an frx	⬆	✓ Closed
CRUC-1507	Next/Prev FRX buttons	⬆	✓ Closed
CRUC-1506	Pressing 'y' in review comments doesn't do anything.	⬆	✓ Closed
CRUC-1505	Marking a file as reviewed should mark all frx comments as read	⬆	✓ Closed
CRUC-1504	Error dialog box background css shows entire sprite	⬆	✓ Closed
CRUC-1503	Defect subtasks have their assignee set to default assignee when resolved from Crucible	⬆	✓ Closed
CRUC-1501	1.6 -> 2.0 beta upgrade truncates timestamps	❗	✓ Closed
CRUC-1499	IDEA direct link port changed from 6666 to 51234	⬆	✓ Closed
CRUC-1496	FRX reloads cause review model sort problems	❗	✓ Closed
CRUC-1493	Show a message in the frx pane when all files are filtered	❗	✓ Closed
CRUC-1492	Files from different repos breaks Manage Files	⬆	✓ Closed
CRUC-1491	I think it should be sticky. i.e. set your preference each time you toggle it.	❗	✓ Closed
CRUC-1487	Need to show initial sort order on the Review dashboard	⬆	✓ Closed
CRUC-1486	Adding a changeset to a review results in duplicate FRXs	⬆	✓ Closed
CRUC-1484	Linked subtask in comment doesn't have a JIRA hover	⬆	✓ Closed
CRUC-1480	Revision slider doesn't resize properly when you edit revisions	⬆	✓ Closed
CRUC-1477	Clicking on a source line when creating a file comment kills the file comment	⬆	✓ Closed
CRUC-1475	Moving slider on image diff causes NPE	⬆	✓ Closed
CRUC-1474	Suggested reviews dialog uses old style inline review dialog	⬆	✓ Closed
CRUC-1473	Review actions missing from inline review dialog	⬆	✓ Closed

CRUC-1469	"You do not have permission to see all the search results." Looks crap.	⬆	Closed
CRUC-1461	Unable to add source file to review	⬆	Closed
CRUC-1460	fix styling of crucible project side bar	⬆	Closed
CRUC-1446	FishEye activity in activity stream when in CruOnly mode	⬆	Closed
CRUC-1438	Review Layout Issues	⬆	Closed
CRUC-1434	Toolbar buttons don't hover and they dont show a cursor	⬇	Closed
CRUC-1429	Tooltip over the top of Review hover	▼	Closed
CRUC-1425	Folders show in review file list	⬆	Closed
CRUC-1323	Bug in adding iterative revisions	⬆	Closed
CRUC-1315	Suggestions are not accurate	⬆	Closed
CRUC-1224	keep slider visible during frx reload	⬆	Closed

From 1.6.6 to 2.0 Beta2

Full list of issues in this release:

JIRA Issues (13 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-1488	Improve user interface in the frx tree	⬆	Closed
CRUC-1478	Single frx visible mode	⬆	Closed
CRUC-1476	File revision comments aren't autosaved	⬆	Closed
CRUC-1465	Synchronise comment buttons when states are changed	⬆	Closed
CRUC-1464	Fix rev slider labels so that slider can snap	⬆	Closed
CRUC-1463	User has no display name	⬆	Closed
CRUC-1462	Selecting changeset in Manage Files fails	⬆	Closed
CRUC-1458	Account signup pages still use old styles	⬆	Closed
CRUC-1454	Timezone problem in review popup	⬆	Closed
CRUC-1439	Crucible Standalone - Manage Files links to FishEye	⬆	Closed
CRUC-1427	Old error page for disabled FishEye Urls	▼	Closed
CRUC-1396	Empty reviews cannnot be abandoned/closed/viewed	⬆	Closed

CRUC-1116	No notifications sent for comments created through REST	↑	Closed

Crucible 2.0 Upgrade Notes

This page contains information about upgrading to Crucible 2.0.

On this page:

- Browsers
- Known Issues
    - MySQL Database Issues
    - Problems with Crucible Freezing Unexpectedly

Browsers

Crucible 2.0 now supports the following browsers:

- Safari 3 (or later)
- FireFox 3 (or later)
- Internet Explorer 7 (or later)

⚠ Internet Explorer 6 is no longer supported.

Known Issues

MySQL Database Issues

When migrating your database to MySQL, you may encounter problems with very long comments in MySQL.

Problems with Crucible Freezing Unexpectedly

A known issue may cause Crucible 2.0 to freeze unexpectedly.

Crucible 2.0 Beta Release Notes

Crucible 2.0 Beta is a public development release leading up to **Crucible 2.0**. For all production use and testing of Crucible, please use the latest official release.

This page refers to an updated version of the Beta (Beta 3). We strongly recommend all beta users upgrade to this release.

> Do not use in production.
> Beta releases should not be used in production environments.

⚠ Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing Crucible development process. As such:
  - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
  - Features in development releases may be incomplete, may change or be removed before the next full release.
  - FireFox 3 and Safari are the only browsers supported.

5 June 2009

Atlassian presents Crucible 2.0 Beta

Crucible 2.0 adds the all-new Iterative Reviews feature, enhanced JIRA integration and a brand new user interface.

Highlights of this release:

- Iterative Reviews
- Enhanced JIRA Integration
- Stars Feature
- Unique Avatars
- People View
- New User Interface
- Plus numerous improvements and bug fixes

Thank you for your interest in Crucible 2.0 Beta.

DOWNLOAD latest version

See the documentation on Upgrading to this version.

Installing Crucible 2.0 Beta

You can now download the Crucible 2.0 Beta from here. See the documentation on Upgrading to this version.

Highlights of Crucible 2.0 Beta

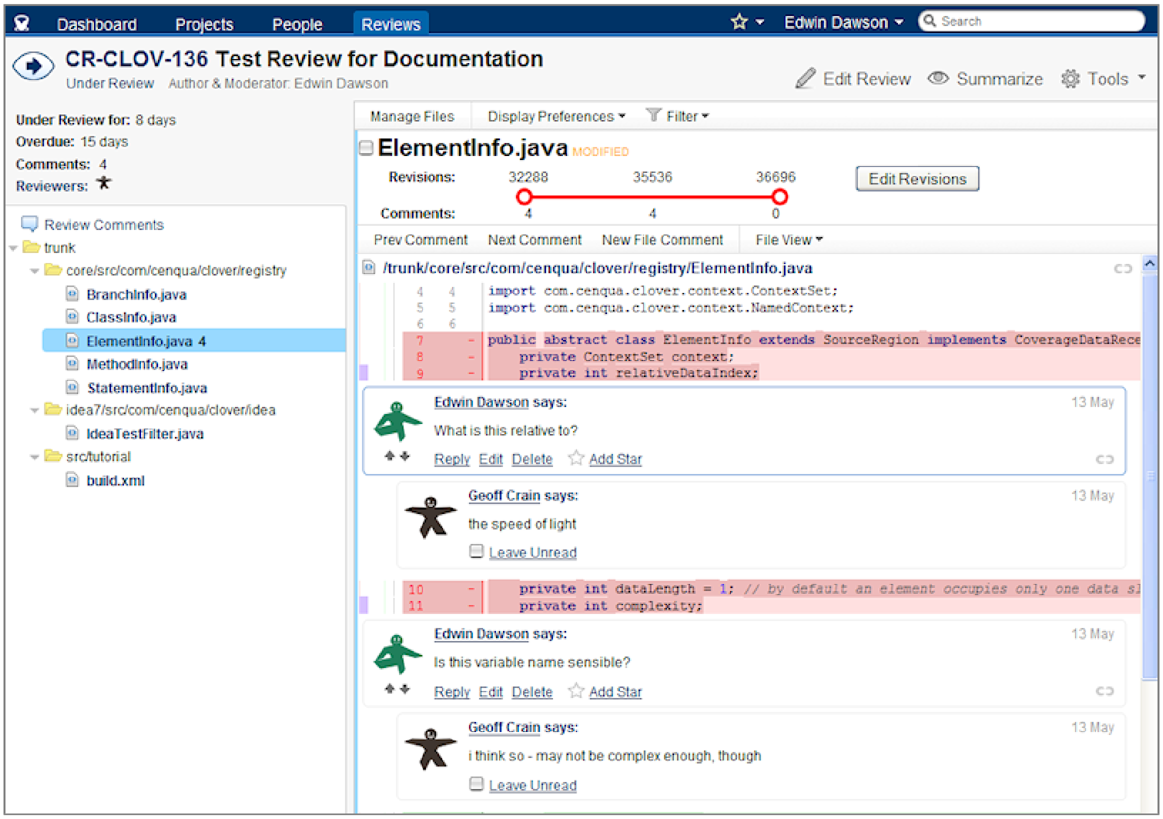
1

Iterative Reviews

Crucible now allows you to review several revisions of a file within one review, seamlessly switching between them.

Comments are persistent and relative to a specific revision. This allows you to review every change that has occurred on a code file within a given period of time and hence visualise its evolution in the context of the review.

Screenshot: Iterative Reviews



CR-CLOV-136 Test Review for Documentation

Under Review   Author & Moderator: Edwin Dawson

Under Review for: 8 days
Overdue: 15 days
Comments: 4
Reviewers:

ElementInfo.java MODIFIED

Revisions:	32288	35536	36696
Comments:	4	4	0

/trunk/core/src/com/cenqua/clover/registry/ElementInfo.java

```
import com.cenqua.clover.context.ContextSet;
import com.cenqua.clover.context.NamedContext;

-public abstract class ElementInfo extends SourceRegion implements CoverageDataRece
-    private ContextSet context;
-    private int relativeDataIndex;
```

Edwin Dawson says: 13 May
What is this relative to?

Geoff Crain says: 13 May
the speed of light

```
-    private int dataLength = 1; // by default an element occupies only one data sl
-    private int complexity;
```

Edwin Dawson says: 13 May
Is this variable name sensible?

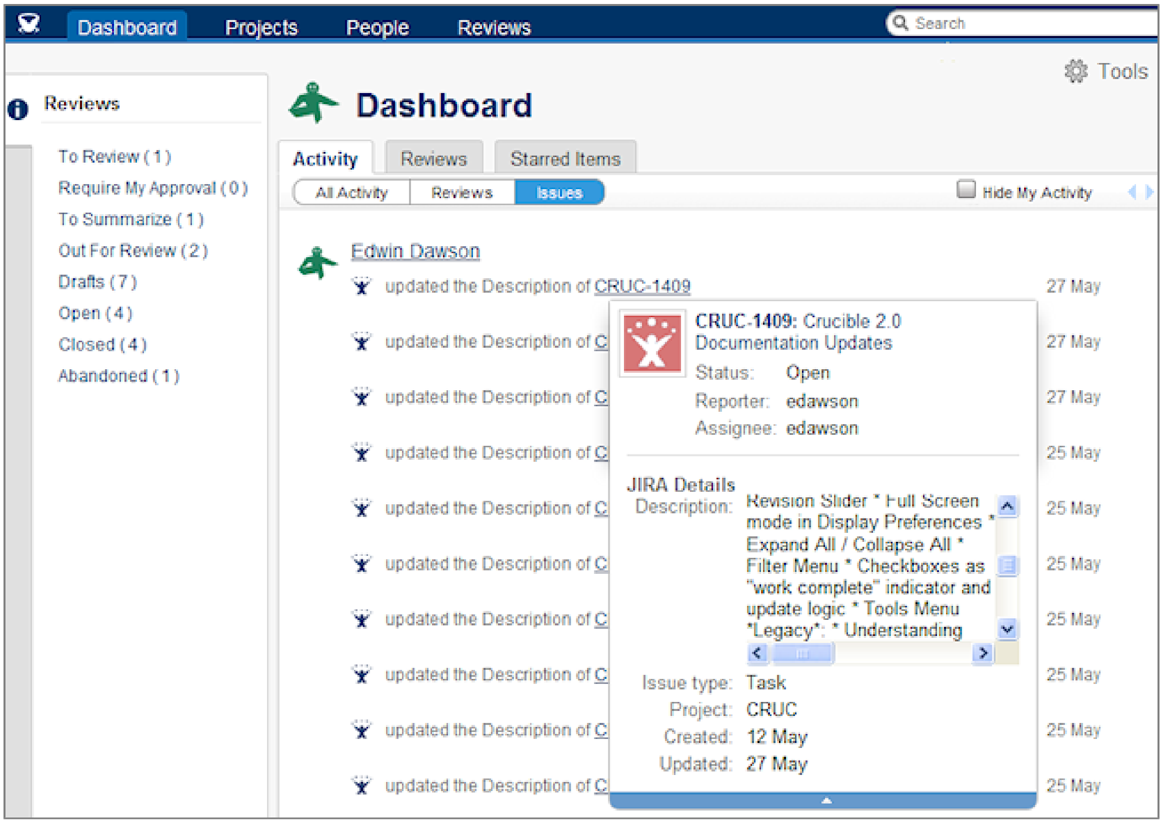
Geoff Crain says: 13 May
i think so - may not be complex enough, though

2

Enhanced JIRA Integration

Crucible now has better JIRA integration, allowing you to see regular JIRA updates in your Crucible dashboard and create JIRA sub-tasks inline, without leaving the Crucible interface. You can still click on issue names to visit the JIRA instance they belong to, also. See instructions for JIRA configuration.

Screenshot: Enhanced JIRA Integration



Edwin Dawson

updated the Description of CRUC-1409 27 May

updated the Description of C 27 May

updated the Description of C 27 May

updated the Description of C 25 May

updated the Description of C 25 May

updated the Description of C 25 May

updated the Description of C 25 May

updated the Description of C 25 May

updated the Description of C 25 May

updated the Description of C 25 May

CRUC-1409: Crucible 2.0 Documentation Updates

Status: Open
Reporter: edawson
Assignee: edawson

JIRA Details
Description: Revision Slider * Full Screen mode in Display Preferences * Expand All / Collapse All * Filter Menu * Checkboxes as "work complete" indicator and update logic * Tools Menu *Legacy*: * Understanding

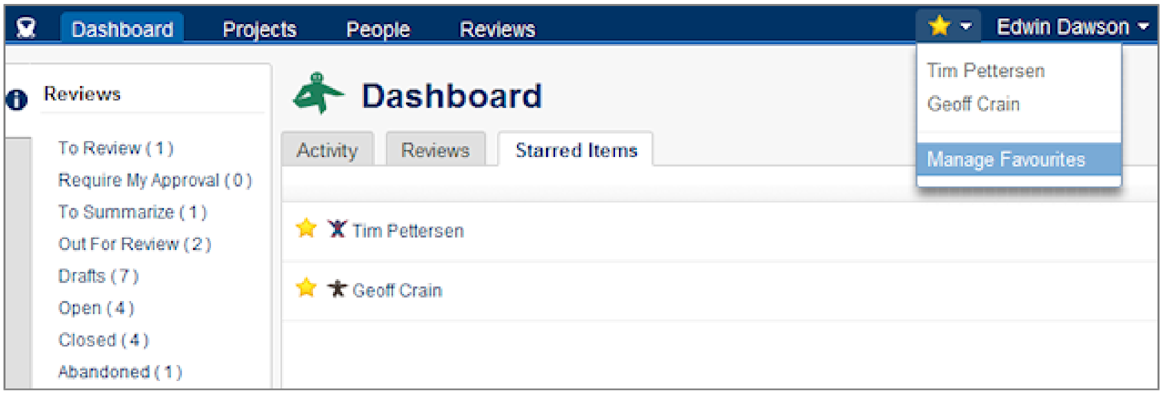
Issue type: Task
Project: CRUC
Created: 12 May
Updated: 27 May

3

Stars Feature

Crucible now allows you to create a list of "Starred" favourite items that includes reviews, JIRA issues, colleagues and more. These favourites can be viewed together in aggregate as a stream of active work you are doing.

Screenshot: Stars

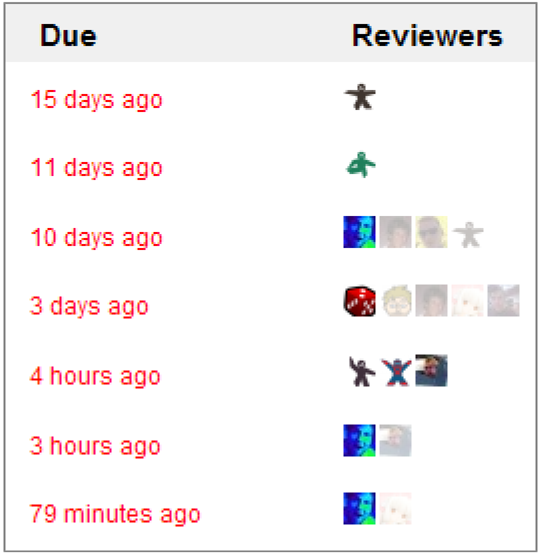


4

Unique Avatars

Crucible will now generate a unique "Charlie" image that will be used as your avatar in the system. These avatars create a new visual linkage to the personalities working on various items and are used as a visual shorthand to show user involvement on items in menu screens and dialogs.

Screenshot: Unique Avatars



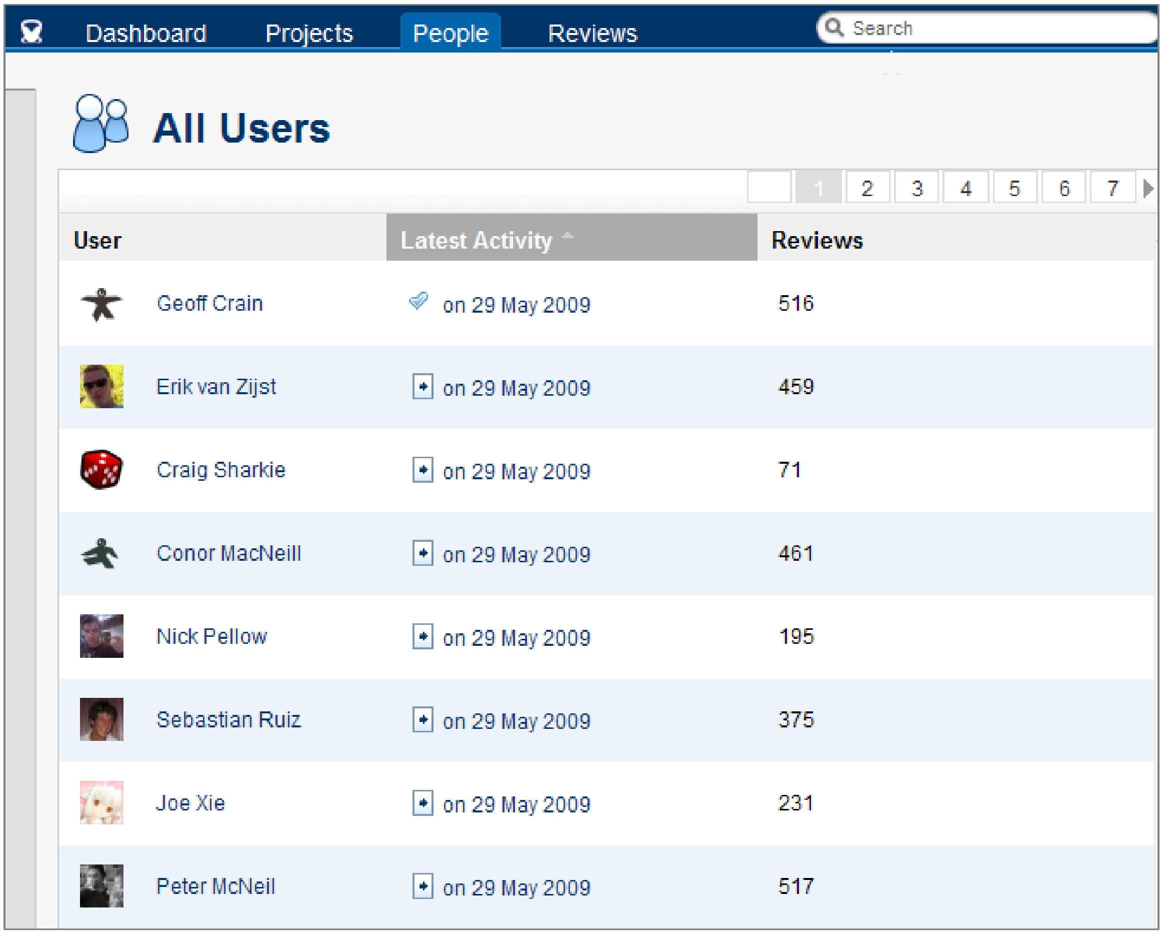
Due	Reviewers
15 days ago	
11 days ago	
10 days ago	
3 days ago	
4 hours ago	
3 hours ago	
79 minutes ago	

5

People View

You can now view detailed charts and activity statistics people who use your Crucible instance. You can compare number of reviews complete and other metrics charted over time.

Screenshot: People View



All Users

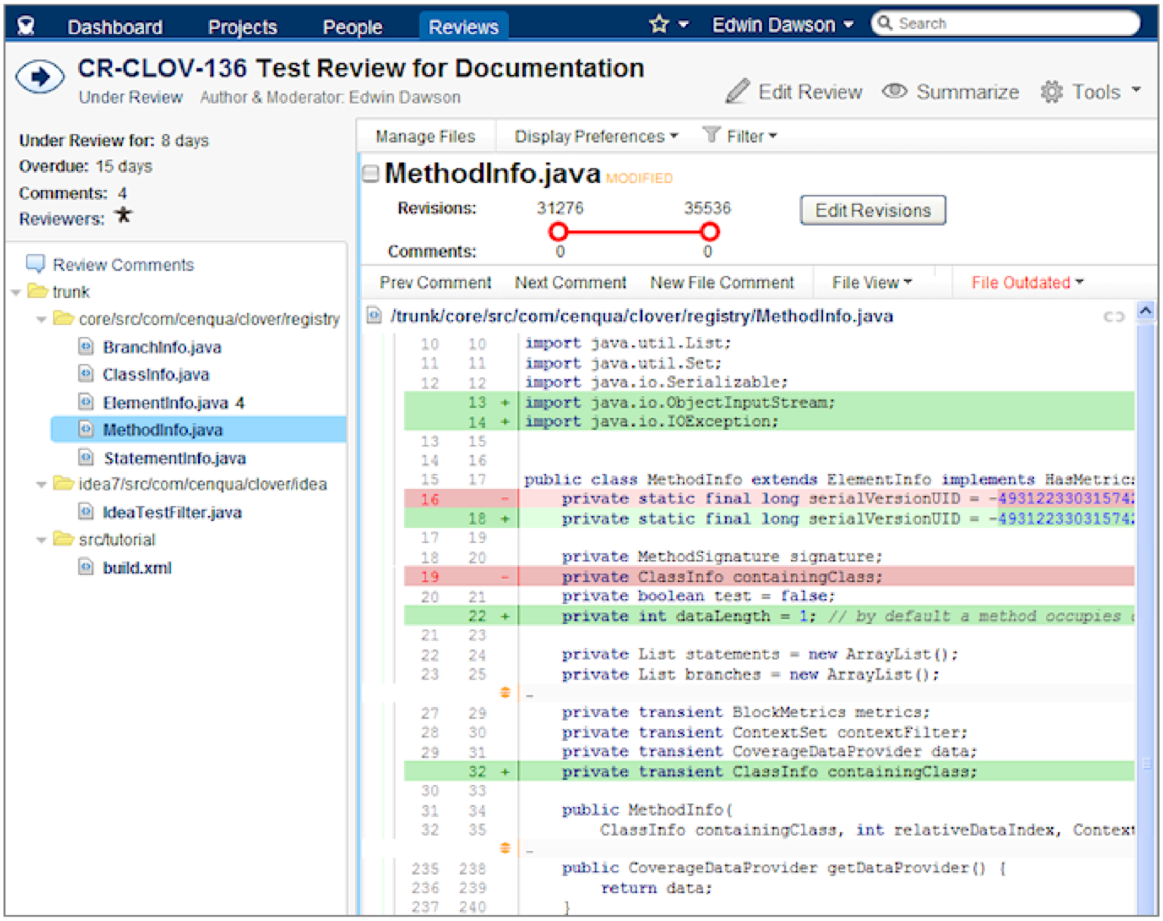
User	Latest Activity	Reviews
Geoff Crain	on 29 May 2009	516
Erik van Zijst	on 29 May 2009	459
Craig Sharkie	on 29 May 2009	71
Conor MacNeill	on 29 May 2009	461
Nick Pellow	on 29 May 2009	195
Sebastian Ruiz	on 29 May 2009	375
Joe Xie	on 29 May 2009	231
Peter McNeil	on 29 May 2009	517

6

New User Interface

Taking on board wide-ranging feedback from customers, the Crucible team has completely revamped the user interface of the product, adding more views on your work and allowing you to access controls from multiple locations and allowing for different work styles. Files in review are arranged in a tree for easy navigation. New viewing modes for reviews support full screen view, side-by-side diff view and single or multiple file views.

Screenshot: New User Interface



CR-CLOV-136 Test Review for Documentation

Under Review   Author & Moderator: Edwin Dawson

Under Review for: 8 days
Overdue: 15 days
Comments: 4
Reviewers:

MethodInfo.java MODIFIED

Revisions: 31276 35536
Comments: 0 0

/trunk/core/src/com/cenqua/clover/registry/MethodInfo.java

```
  import java.util.List;
  import java.util.Set;
  import java.io.Serializable;
+ import java.io.ObjectInputStream;
+ import java.io.IOException;


  public class MethodInfo extends ElementInfo implements HasMetric:
-     private static final long serialVersionUID = -49312233031574:
+     private static final long serialVersionUID = -49312233031574:

      private MethodSignature signature;
-     private ClassInfo containingClass;
      private boolean test = false;
+     private int dataLength = 1; // by default a method occupies :

      private List statements = new ArrayList();
      private List branches = new ArrayList();
  -
      private transient BlockMetrics metrics;
      private transient ContextSet contextFilter;
      private transient CoverageDataProvider data;
+     private transient ClassInfo containingClass;

      public MethodInfo(
          ClassInfo containingClass, int relativeDataIndex, Context
  -
      public CoverageDataProvider getDataProvider() {
          return data;
      }
```

7

Plus numerous improvements and bug fixes

Visit our issue tracker to see to see the full list of improvements and bug fixes between Beta 2 and Beta 3. We strongly recommend all beta users upgrade to the latest beta release.

See the Beta Reviewer's Guide for a list of known issues and guidance on the beta experience.

Crucible 2.0 Beta Upgrade Notes

Crucible 2.0 Beta is a public development release leading up to **Crucible 2.0**. For all production use and testing of Crucible, please use the latest official release.

> ⛔ Do not use in production.
> Beta releases should not be used in production environments.

⚠ Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing Crucible development process. As such:
    - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
    - Features in development releases may be incomplete, or may change or be removed before the next full release.
    - There will be an upgrade path from the 2.0 Beta to the final release.

This page contains instructions on how to upgrade your Crucible instance to the Crucible 2.0 Beta.

ℹ Read about how your Crucible installation works with FishEye.

Before you Start

- Before upgrading you should always read the Release Notes for the version you are upgrading to, as well as any versions you are skipping.
- **We strongly recommend you make a backup of your data before upgrading Crucible**. Simply make a copy of your `crucible_install_dir/var/data/` directory.
- Download the Crucible zip file.

Upgrade Procedure

Method 1 - Using a FISHEYE_INST Directory

If you have Crucible configured to use a `FISHEYE_INST` directory, then simply:

1. Shutdown your existing fisheye server
2. Make a backup of your FISHEYE_INST directory
3. Extract the new Crucible version to a directory.
4. Leave your `FISHEYE_INST` environment variable set to its existing location.
5. Start Crucible from the new installation.
6. Follow any version-specific instructions found in the Release Notes.

ℹ Read more about the `FISHEYE_INST` environment variable.

Method 2 - Without a FISHEYE_INST Directory

If you are not using `FISHEYE_INST`, you will need to copy some files from your old Crucible installation to your new one.

1. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
2. Delete the `/NEW_FISHEYE/var` directory.
3. Shut down the old Crucible instance if it is running.
4. Copy `/OLD_FISHEYE/config.xml` to `/NEW_FISHEYE/`.
5. Copy (or move) the `/OLD_FISHEYE/var` directory to `/NEW_FISHEYE/var`.
6. If you have a Cenqua-issued Crucible license, copy all `/OLD_FISHEYE/*.license` files to `/NEW_FISHEYE/`. (Atlassian-issued licenses are included within `config.xml`.)
7. Follow any version-specific instructions found in the Release Notes.

Method 3 - Without a FISHEYE_INST Directory, but would like to set one up

1. Shut down the old FishEye instance if it is running.
2. Set up the `FISHEYE_INST` environment variable, then create the FISHEYE_INST directory on your filesystem.
3. Copy the `/OLD_FISHEYE/config.xml` to `/FISHEYE_INST`.
4. Copy the `/OLD_FISHEYE/var` directory to `/FISHEYE_INST`.
5. Download Crucible.
6. Extract the new Crucible archive into a directory such as `/NEW_FISHEYE/`.
7. Start Crucible from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
8. Follow the initial configuration steps outlined below.
9. If you configuration is not automatically picked up and you cannot see your existing repositories, check your Administration > Sys-Info page, where you will see information about FISHEYE_HOME and FISHEYE_INST. Check your FISHEYE_INST is pointing to the right directory.

Crucible 2.0 Beta Reviewer's Guide

Crucible 2.0 Beta is a public development release leading up to **Crucible 2.0**. For all production use and testing of Crucible, please use the latest official release.

⛔ Do not use in production.
Beta releases should not be used in production environments.

⚠ Please also take note of the following information:

- Beta releases are not safe — Beta releases are snapshots of the ongoing Crucible development process. As such:
  - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
  - Features in development releases may be incomplete, or may change or be removed before the next full release.
  - FireFox 3 and Safari are the only browsers supported.

Thank you for your interest in the Crucible 2.0 Beta. This page contains some direction on what is ready for testing, what the known issues are and how you can submit feedback.

Known Issues

This is a list of known issues with the Crucible 2.0 Beta; please do not raise requests related to these as solutions for them are already under way.

- REST API updates are not fully functional; you can't access various features from the IDE Connectors or other technology that depends on REST. The features affected include marking comments as read, Iterative Reviews and Activity Streams. Also, JSON support is alpha at this stage.
- JIRA sub-tasks integration will not work on JIRA instances where the JIRA General Setting "allow unassigned issues" is set to 'OFF'.
- Sorting functionality is only partially functional; when you are trying to sort content — such as review lists on the reviews page, content on the dashboard and project dashboard — the sorting of content by name, date and so on only applies to the current page of results. To mitigate this, narrow down your searches so that you only have one page of results. This will be revised to have server-side sorting.
- You can't create reviews with files or changesets from multiple repositories.

Features Ready For Testing

The following features in the Crucible 2.0 Beta are relatively hardened and using these thoroughly will help contribute to the final product.

- Iterative Reviews; create a review on multiple revisions of a changeset, to see the history and evolution of the code in-line.
- Activity Streams; see review activity, code commits and JIRA activity (when properly configured) on the Dashboard.
- External Database Support; You can now store Crucible's internal data in a MySQL or PostgreSQL database, as an alternative to the built-in HSQLDB.
- Crucible can assist you in updating reviews by suggesting changesets that are likely related to your reviews.
- Reviews can now have a Due Date. Reviews that are overdue will show up in red on the reviewer's dashboards to minimise the chance of reviews getting stale.
- Stars; add colleagues, reviews and files to your favourites list, then view updates related to them as a feed.
- Charlietars; the automatically generated Crucible avatars should work smoothly. Also, you can sign up to Globally Recognised Avatars (http://www.gravatar.com) to upload a profile image and use that instead of the Charlie image.
- Scheduled Backups; you can now easily set Crucible to backup data periodically using a simple calendar function in the user interface.

Submitting feedback

To submit feedback on the Crucible 2.0 Beta, please use the Crucible Forums.

JIRA Integration in Crucible 2.0 Beta

Crucible 2.0 Beta is a public development release leading up to **Crucible 2.0**. For all production use and testing of Crucible, please use the latest official release.

> Do not use in production.
> Beta releases should not be used in production environments.

> Please also take note of the following information:
>
> - Beta releases are not safe — Beta releases are snapshots of the ongoing Crucible development process. As such:
>   - While we try to keep these releases stable, they have not undergone the same degree of testing as a full release.
>   - Features in development releases may be incomplete, or may change or be removed before the next full release.

This page contains instructions for setting up JIRA integration in Crucible.

JIRA is Atlassian's issue tracking product, which can be used to manage projects and associated work.

Before you begin: Ensure that you configure your JIRA instance to enable sub-tasks, enable unassigned issues and allow Remote API access. The instructions on this page have been tested with JIRA 3.13.4.

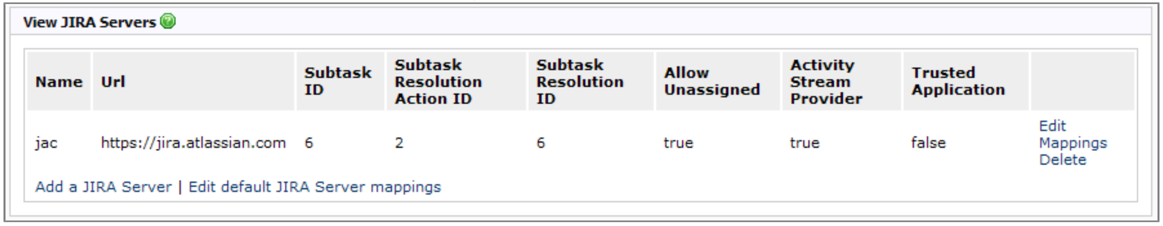
On this page:

JIRA issues can be viewed in the main Dashboard view in Crucible. This requires you to enter details on the required JIRA server(s) via the Crucible administration screens.

Opening the Administration Screen for JIRA Integration

To set up JIRA integration, open the Administration screen and then click '**JIRA Servers**' under the '**Global Settings** ' sub-menu on the left navigation bar. The '**View JIRA Servers**' administration page opens.

Screenshot: The View JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Add a JIRA Server | Edit default JIRA Server mappings

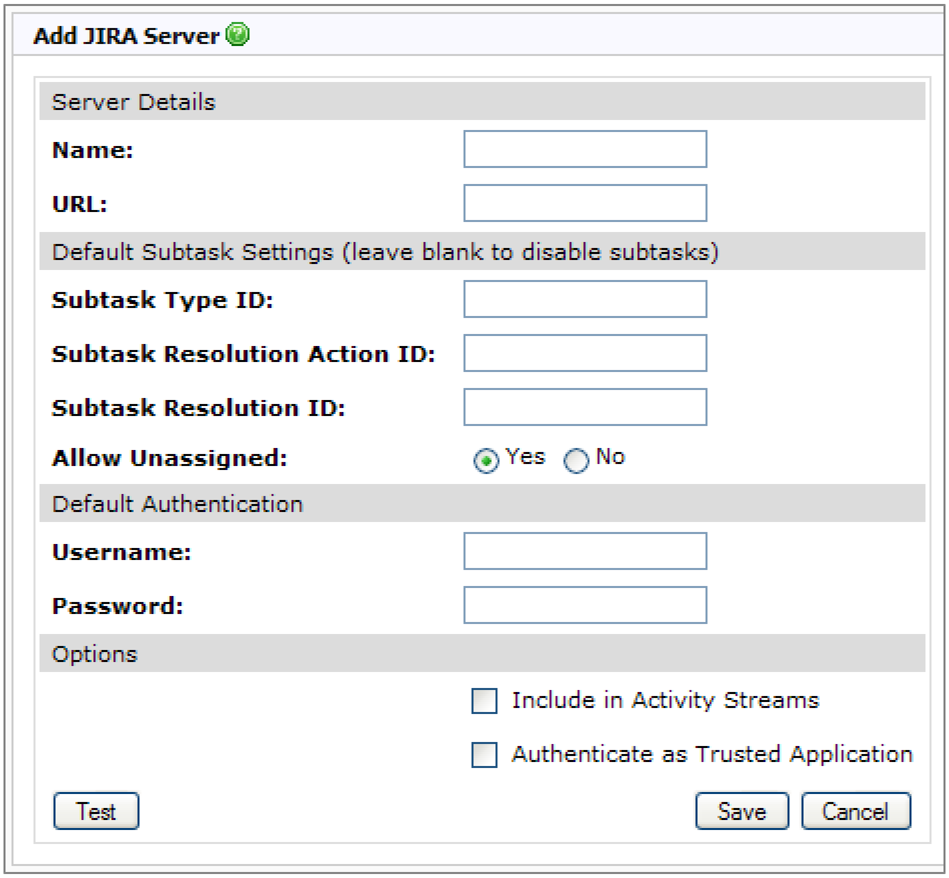
On the View JIRA Servers page, you can carry out a number of operations as listed on this page.

Adding a New JIRA Server

To add a new JIRA server from the View JIRA Servers page, click '**Add JIRA Server**'.

The '**Add JIRA Server**' page opens.

Screenshot: The Add JIRA Server Page



Add JIRA Server

Server Details

Name:

URL:

Default Subtask Settings (leave blank to disable subtasks)

Subtask Type ID:

Subtask Resolution Action ID:

Subtask Resolution ID:

Allow Unassigned: ● Yes ○ No

Default Authentication

Username:

Password:

Options

□ Include in Activity Streams

□ Authenticate as Trusted Application

A number of fields and options must be filled out or selected on this page. See the table below for information on each field.

Option	Type	Description	Required
Name	Text Field	A descriptive name for the JIRA server.	Yes
URL	Text Field	The Internet address of the JIRA server.	Yes
Subtask Type ID	Number	This is required to enable creating issues from a Crucible comment.	No
Subtask Resolution Action ID	Number	This is required to enable creating issues from a Crucible comment. See below for instructions on obtaining this number.	No
Subtask Resolution ID	Number	This is required to enable creating issues from a Crucible comment. See below for instructions on obtaining this number.	No
Allow Unassigned	True/False Button	Allow unassigned sub-tasks.	No
Username	Text Field	The username of an account on the JIRA instance (All activity that takes place will be attributed to this user, unless using the Trusted Application setting).	Yes
Password	Text Field	The password for the account on the JIRA instance.	Yes

Include in Activity Streams	Check Box	Allows JIRA information to appear on the Dashboard.	No
Authenticate as Trusted Application	Check Box	Allows the system to interface with JIRA and let users log on with their own accounts (and use their own accounts on the JIRA server. See complete FishEye documentation and complete JIRA documentation.	No

Once you've filled out the necessary fields, click '**Test**' to ensure that your details are correct. If you have a positive message return from the test, click '**Save**'.

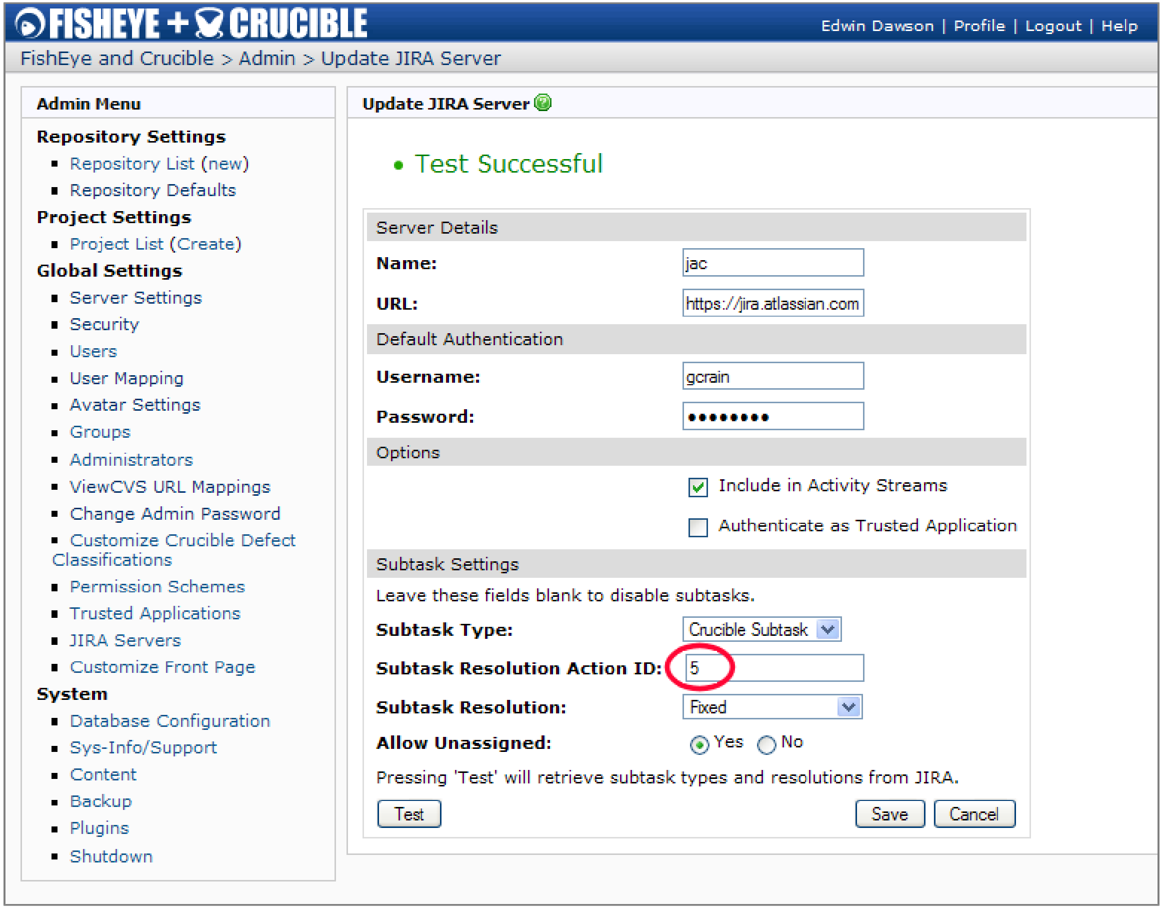
Obtaining the Subtask Type ID

ℹ This value is required (along with the Subtask Resolution ID and Subtask Resolution Action ID) to enable creating issues from a Crucible comment. This is the subtask type that will be created when you create a JIRA subtask in Crucible.

To obtain this value, carry out the following steps.

1. Enable sub-tasks on your JIRA instance from the '**JIRA Administration' > '*Sub-Tasks**' page. See the JIRA documentation for details on this step.
2. Return to the FishEye Administration screen and then click '**JIRA Servers**' under the '**Global Settings**' sub-menu on the left navigation bar. Click '**Edit**' next to the JIRA server you have configured.
3. Your JIRA server's basic details should appear. Click '**Test**' once again. The field for Subtask Type ID will change to a drop-down menu, showing the available subtask types. Choose the correct one.
4. Save your Crucible configuration settings.

Screenshot: Obtaining the Subtask Type ID



Update JIRA Server

- Test Successful

Server Details

Name:	jac
URL:	https://jira.atlassian.com

Default Authentication

Username:	gcrain
Password:	••••••••

Options

☑ Include in Activity Streams

☐ Authenticate as Trusted Application

Subtask Settings

Leave these fields blank to disable subtasks.

Subtask Type:	Crucible Subtask
Subtask Resolution Action ID:	5
Subtask Resolution:	Fixed
Allow Unassigned:	● Yes ○ No

Pressing 'Test' will retrieve subtask types and resolutions from JIRA.

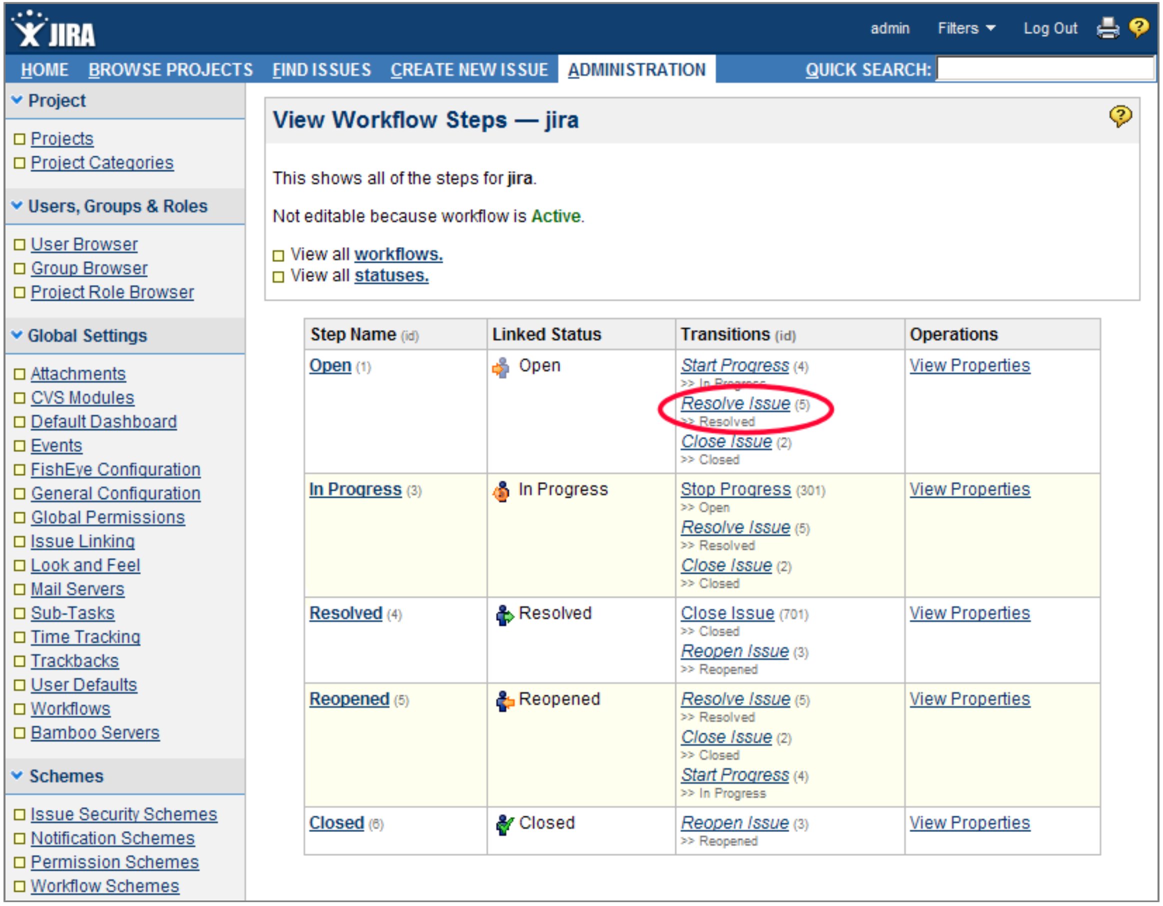
Obtaining the Subtask Resolution ID and Subtask Resolution Action ID

These values are required (along with the Subtask Type ID) to enable creating issues from a Crucible comment.

To obtain these values, carry out the following steps.

1. Open your JIRA instance and go to '**Administration**' > '**Workflows**'. The '**Workflows**' screen opens. By default, the '**JIRA**' workflow is shown on screen in a table.
2. Click the '**Steps**' link in the far right table cell. The '**View Workflow Steps — JIRA**' page opens.
3. The '**Subtask Resolution Action ID**' is in the '**Open**' row, under the '**Transitions**' column. Look at the link in that cell named '**Resolve Issue**'. The ID number is shown in brackets next to that heading '**Resolve Issue**' (shown in the screenshot below as 5).
4. Save your Crucible configuration settings.
5. The '**Subtask Resolution ID**' is the '**Resolved ID**' on this page. The ID number is shown in brackets next to the heading '**Resolved**' (shown in the screenshot below as '4'). Note it down and enter it into the Crucible configuration screen.
6. Save your Crucible configuration settings.

Screenshot: Obtaining the Subtask Resolution ID & Subtask Resolution Action ID



View Workflow Steps — jira

This shows all of the steps for **jira**.

Not editable because workflow is Active.

□ View all **workflows.**
□ View all **statuses.**

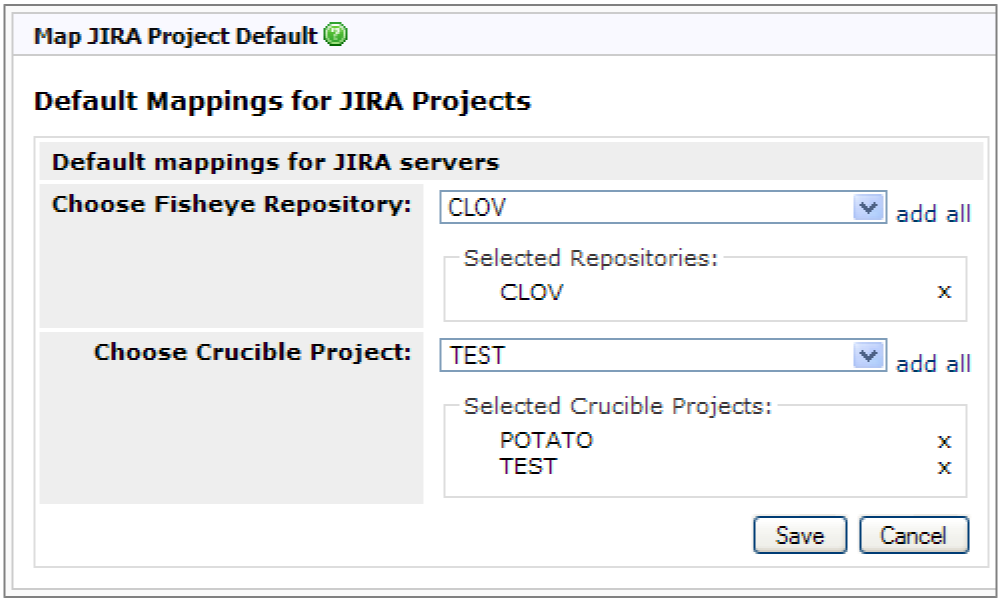
Step Name (id)	Linked Status	Transitions (id)	Operations
Open (1)	Open	Start Progress (4) >> In Progress Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
In Progress (3)	In Progress	Stop Progress (301) >> Open Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
Resolved (4)	Resolved	Close Issue (701) >> Closed Reopen Issue (3) >> Reopened	View Properties
Reopened (5)	Reopened	Resolve Issue (5) >> Resolved Close Issue (2) >> Closed Start Progress (4) >> In Progress	View Properties
Closed (6)	Closed	Reopen Issue (3) >> Reopened	View Properties

Editing Default JIRA Server Mappings

This setting enables the Crucible feature that shows JIRA information in a dynamic window when you hover the mouse over a JIRA issue key in Crucible. It will also turn every issue key into a hyperlink to that issue in Crucible.

To enable this feature, click '**Edit Default JIRA Server Mappings**' from the View JIRA Servers page. The '**Map JIRA Project Default**' page opens.

Screenshot: The Default JIRA Server Mappings Page



Map JIRA Project Default

Default Mappings for JIRA Projects

Default mappings for JIRA servers

Choose Fisheye Repository: CLOV

Selected Repositories:
CLOV

Choose Crucible Project: TEST

Selected Crucible Projects:
POTATO
TEST

On this page, select the FishEye repositories or Crucible Projects that you wish to associate with all the JIRA servers you have configured for use in Crucible. You can click '**add all**' to quickly include them all in this category. You can remove individual items by clicking the small 'X' marks.

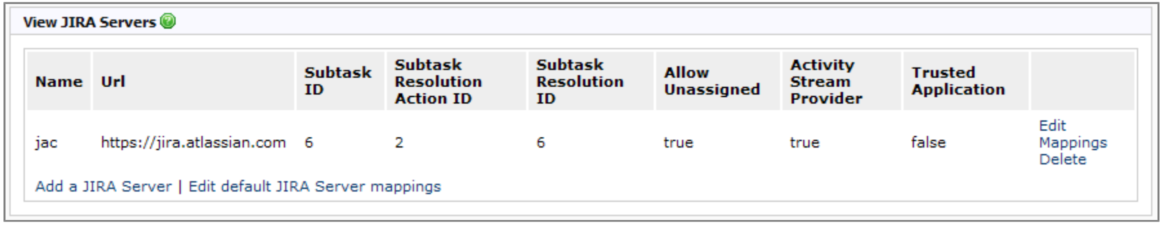
Once you've finished, click '**Save**'.

⚠ You should disable any existing Crucible linkers you have set up for JIRA, as they will override this feature and prevent the dynamic dialog box from appearing when you mouse over an issue.

Operations on Existing Servers

Once you have configured an existing JIRA server, there are three main operations you can carry out on it: '**Edit**', '**Mappings**' and '**Delete**'. These options appear on the far right of the screen.

Screenshot: Operations in the JIRA Servers Page



View JIRA Servers

Name	Url	Subtask ID	Subtask Resolution Action ID	Subtask Resolution ID	Allow Unassigned	Activity Stream Provider	Trusted Application	
jac	https://jira.atlassian.com	6	2	6	true	true	false	Edit Mappings Delete

Edit settings for an existing JIRA server

When you click '**Edit**', you can adjust any of the general settings you configured when you first added the server.

Edit mappings for an existing JIRA server

When you click '**Mappings**', a page is loaded that is almost identical to the '**Default Mapping**' screen, but allows you to choose mappings only for that specific JIRA server.

Delete an existing JIRA server

Clicking '**Delete**' will remove the server from the list.

Crucible 1.6 Release Notes

23 September 2008

Atlassian presents Crucible 1.6

Crucible release 1.6 makes it easier to review content that is not in FishEye. Furthermore, Crucible 1.6 can be deployed without FishEye for the first time. Through Crucible's new 'Light SCM' plugins, you can include content in reviews that are not associated with FishEye or even a source control repository. For example, you can review pages directly from Confluence, files on any file system connected to the machine FishEye is running on, and Subversion repositories not connected to FishEye. Crucible now has better support for uploading files for pre-commit review, in addition to the existing support for patches.

Highlights of this release:

- Support for Non-FishEye Repositories
- Confluence Page Reviews
- Shared File System Repositories
- Enhanced Pre-commit Reviews & Image Support
- Multiple Admin Users
- Expanded API
- Plus numerous improvements and bug fixes

DOWNLOAD
latest version

Upgrading to Crucible 1.6

You can now download Crucible from here. If upgrading from a previous version, please follow the Upgrade Guide.

Highlights of Crucible 1.6

1

Support for Non-FishEye Repositories

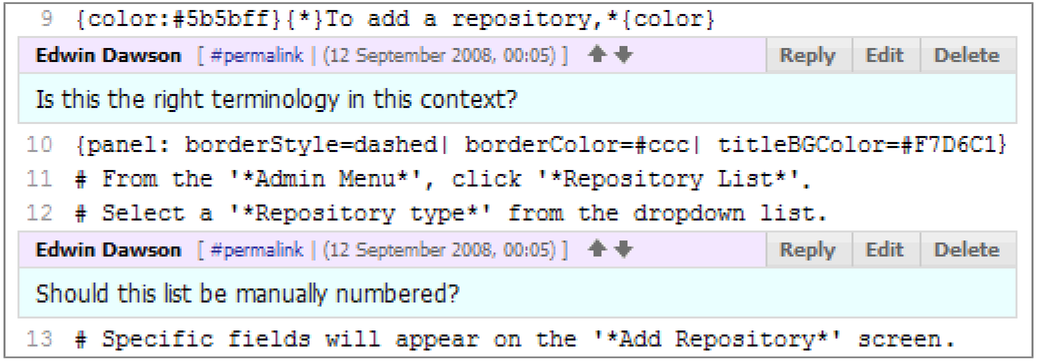
Crucible can now be deployed as a stand-alone application for the first time. With Crucible 1.6, you no longer need a FishEye license or even a source-control repository. Crucible's new *Light SCM plugin* infrastructure already supports Confluence, server file systems, and Subversion repository types. We will be adding GIT and ClearCase in the near future. The Light SCM interface is public and the shipped plugins are open source. As a result, you can extend these plugins or even write your own — great news for plugin developers.

2

Confluence Page Reviews

Crucible 1.6 allows you to select Atlassian Confluence as a source of material for reviews. In this way, you can use Crucible to easily review the Wiki Markup of pages created in Confluence. Read more.

Screenshot: Confluence Page Reviews in Crucible



```
{color:#5b5bff}{*}To add a repository,*{color}
```

Edwin Dawson [#permalink | (12 September 2008, 00:05)]

Is this the right terminology in this context?

```
{panel: borderStyle=dashed| borderColor=#ccc| titleBGColor=#F7D6C1}
# From the '*Admin Menu*', click '*Repository List*'.
# Select a '*Repository type*' from the dropdown list.
```

Edwin Dawson [#permalink | (12 September 2008, 00:05)]

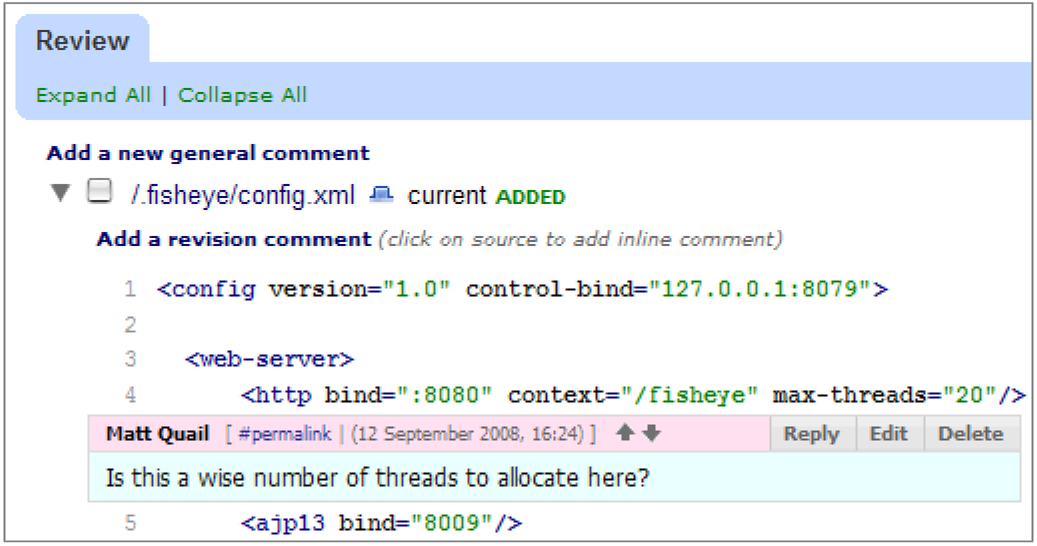
Should this list be manually numbered?

```
# Specific fields will appear on the '*Add Repository*' screen.
```

Shared File System Repositories

You can create a 'repository' for a local or remote directory on the server file system. Teams that are managing documents through a shared file system instead of a source control system can still benefit from peer reviews. Read more.

Screenshot: Crucible Reviews on the File System



/.fisheye/config.xml current ADDED

(click on source to add inline comment)

```
<config version="1.0" control-bind="127.0.0.1:8079">

  <web-server>
      <http bind=":8080" context="/fisheye" max-threads="20"/>
```

Matt Quail [#permalink | (12 September 2008, 16:24)]

Is this a wise number of threads to allocate here?

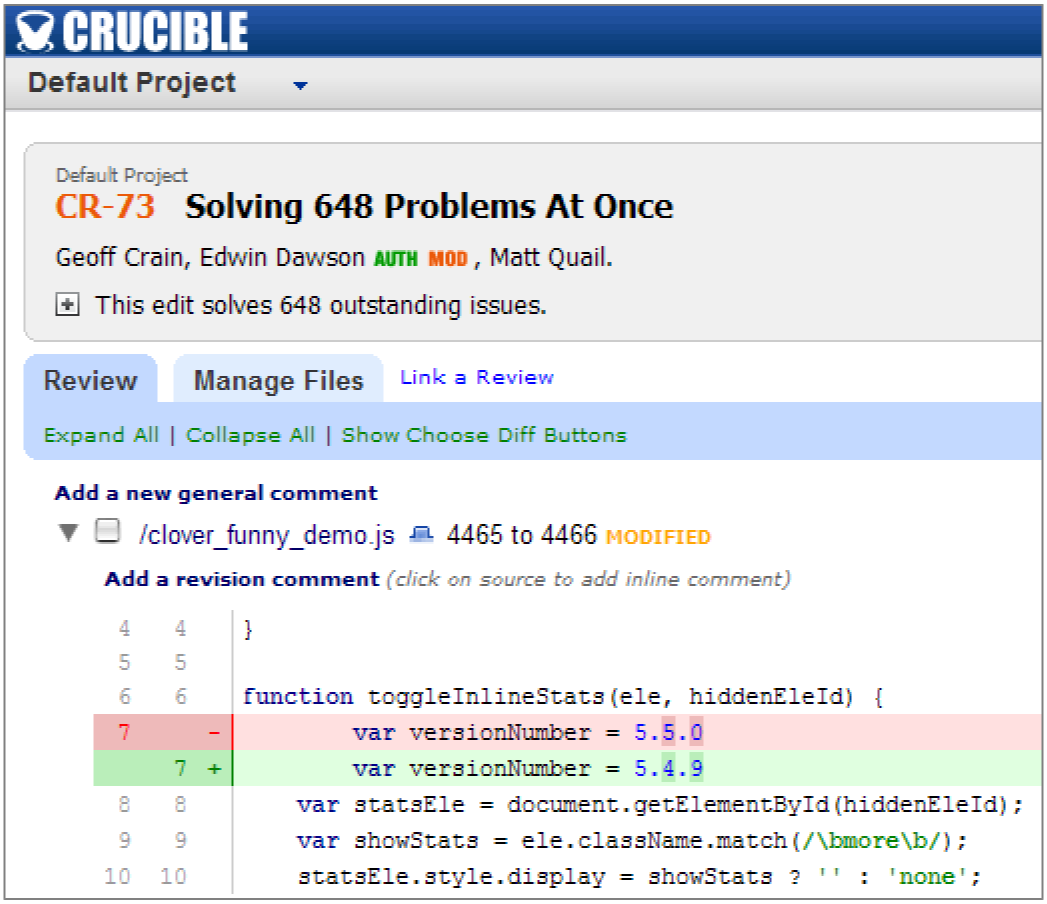
```
      <ajp13 bind="8009"/>
```

Enhanced Pre-commit Reviews & Image Support

In addition to Crucible's patch support, 1.6 enables any file to be uploaded for review. The new upload functionality enables two files to be uploaded and compared in the review window, with diff highlighting. Read more.

Crucible now supports before and after inline image previews.

Screenshot: Crucible Review of Uploaded Files



Default Project

CR-73 Solving 648 Problems At Once

Geoff Crain, Edwin Dawson AUTH MOD , Matt Quail.

⊞ This edit solves 648 outstanding issues.

Review
Manage Files
Link a Review
Expand All | Collapse All | Show Choose Diff Buttons
Add a new general comment

▼ □ /clover_funny_demo.js 4465 to 4466 MODIFIED

Add a revision comment (click on source to add inline comment)

```
}

function toggleInlineStats(ele, hiddenEleId) {
-        var versionNumber = 5.5.0
+        var versionNumber = 5.4.9
    var statsEle = document.getElementById(hiddenEleId);
    var showStats = ele.className.match(/\bmore\b/);
    statsEle.style.display = showStats ? '' : 'none';
```

5

Multiple Admin Users

Crucible now allows the Administrator to grant other users administration privileges. Admin Users can be individually assigned or given privileges through local or remote directory groups. Read more.

6

Expanded API

The Crucible API now allows programmable review creation, along with a host of other additions. Read more.

7

Plus numerous improvements and bug fixes

JIRA Issues (109 issues)			
Key	Summary	Priority	Status
			✓

CRUC-668	We don't update lite SCM revision details if the project does not store revisions	↑	Closed
CRUC-664	remove unecessary transactions from read-only REST requests	↑	✓ Closed
CRUC-663	blank page when a non creator/author/moderator tries to preview a draft review	↑	✓ Closed
CRUC-662	Updating general comments in the REST api has a permission flaw	↓	✓ Closed
CRUC-661	On the Crucible/Fisheye dashboard Projects are displayed even if the user has no access to enter those projects (e.g. anonymous users).	↥	✓ Closed
CRUC-659	Email notifications should be more easily threaded by mail clients	↑	✓ Closed
CRUC-654	improve webwork/jsp error handling	↑	✓ Closed
CRUC-647	improve ReviewManager.countStatesOn query	↑	✓ Closed
CRUC-646	Disabling and reenabling a plugin gives 'javax.el.ELException: org.springframework.osgi.service.importer.ServiceProxyDestroyedException: service proxy has been destroyed'	↥	✓ Closed
CRUC-645	Optimise DB queries	↑	✓ Closed
CRUC-640	Shutdown server documentation	↑	✓ Closed
CRUC-639	Admin password documentation	↑	✓ Closed
CRUC-635	Can't save changes to statement of objectives when at .../{id}/confirmApprove	↑	✓ Closed
CRUC-634	ReviewData should not include repoName	↑	✓ Closed
CRUC-632	Help text not updated after saving a review...	↑	✓ Closed
CRUC-626	Debug Logging Documentation	↑	✓ Closed
CRUC-623	SvnChangeLogBrowser.listChanges() in LiteSVN fails when processing revisions that are above the repo's configured base path	↑	✓ Closed
CRUC-622	Improve performance of querying for review details	↑	✓ Closed
CRUC-615	Make sure everything has a help link	↑	✓ Closed
CRUC-610	Removing revisions from review that have draft comments fails without an explanation	↑	✓ Closed
CRUC-608	Show a list of LightSCM repositories in admin	↥	✓ Closed
	Improve the efficiency of checking whether a review has been completed by all		✓

CRUC-602	reviewers	⬆	Closed
CRUC-600	REST API should return error when non-existent filter used	⬆	Closed
CRUC-598	Emails always include a "Summary" section	⬆	Closed
CRUC-597	Extend the Remote API to allows reviews to be deleted	⬆	Closed
CRUC-594	Review description preview on dashboard swallows new lines from the review description	⬆	Closed
CRUC-592	Filter names used in Cru do not match the menus	⬇	Closed
CRUC-591	renaming a confluence repository adds a new one instead of editing	⬆	Closed
CRUC-589	Change revisionData getAdded/Removed lines an Integer	⬆	Closed
CRUC-588	Search Subtab in Review->Manage Files throws PropertyNotFoundException for all searches	⬆	Closed
CRUC-586	Remove Delete option from project drop down	⬆	Closed
CRUC-579	Changing repository when certain review tabs are selected causes NPE/bad redirect	⬆	Closed
CRUC-574	Allowed Review Participants left blank means what?	⬇	Closed
CRUC-566	lightscm package should not be "fisheye"	⬆	Closed
CRUC-565	need get bulk version of getRevisionData() in SCMRepository	⬆	Closed
CRUC-563	Toggle side-by-side and show per frx diff options on review page	⬆	Closed
CRUC-562	REST API: createReview does not set Moderator and author correctly	⬆	Closed
CRUC-561	anchors to some comment types don't work	⬆	Closed
CRUC-560	REST-API: provide comit type information - esp. deleted file status	⬆	Closed
CRUC-556	Next/previous comment arrows do not jump between file and general comments	⬆	Closed
CRUC-554	Upgrade to non-beta atlassian-plugins	⬆	Closed
CRUC-551	Confluence Light SCM plugin dependencies belong in plugin	⬆	Closed
CRUC-549	Adjust IFRAME size when the admin page is resized		

		↑	Closed
CRUC-548	REST API: Handle allowReviewersToJoin flag on review	↓	✓ Closed
CRUC-547	Make Repository source check if the engine is available.	↑	✓ Closed
CRUC-545	NPE Closing Review	↑	✓ Closed
CRUC-542	Use content manager to retrieve detailed file revisions	↑	✓ Closed
CRUC-540	make metrics-config.xsd available online	↓	✓ Closed
CRUC-539	Serious: "Next comment" does not always work right	⇑	✓ Closed
CRUC-526	Space character in installation path to Crucible disables remote API	↑	✓ Closed
CRUC-519	Added files display "no change"	↑	✓ Closed
CRUC-518	Crucible does not seem to be incrementally updating the FishEye cache	↑	✓ Closed
CRUC-516	REST API: Return detailed information on review in one call - items, comments, reviewers, available actions	↑	✓ Closed
CRUC-511	Added files when stored show "No Change"	↑	✓ Closed
CRUC-507	Confluence: implement change set paging	↑	✓ Closed
CRUC-505	Add "Search for Review" functionality to Crucible remote API	↑	✓ Closed
CRUC-504	Confluence: get revision comments	↑	✓ Closed
CRUC-499	Create separate API jar	↑	✓ Closed
CRUC-496	Document store revisions with review	↑	✓ Closed
CRUC-494	Don't make source type visible in URLs or elsewhere	↑	✓ Closed
CRUC-488	clicking summarise without clicking close makes the review 'disappear' from the workflow	↑	✓ Closed
CRUC-487	Page title says "dead" for a review that has been abandoned	↓	✓ Closed
CRUC-484	you can link a review to itself	↑	✓ Closed
CRUC-479	A comment which was a defect, but is no longer, still shows the defect attributes		✓

		⬇	Closed
CRUC-478	Ability to specify that Branches are not supported	⬆	✓ Closed
CRUC-472	Changing tab while editing review deletes details	⬆	✓ Closed
CRUC-469	Sort list of repositories alphabetically on "Add project" page	▼	✓ Closed
CRUC-461	Line comments are sometimes rendered twice	⬆	✓ Closed
CRUC-457	show authorname next to revision in revision dropdowns	⬇	✓ Closed
CRUC-456	crucible doesn't create default permission scheme when creating a blank db	⬆	✓ Closed
CRUC-455	creating a review from a changeset with a non-existent repo keeps redirecting to the login screen	⬆	✓ Closed
CRUC-453	Allow adding a screenshot as an attachment to a review	⬆	✓ Closed
CRUC-439	Invite to review	⬆	✓ Closed
CRUC-438	Not sending email notifications after Crucible and Fisheye upgrade	⬆	✓ Closed
CRUC-436	create 'remove all revision' link for all tabs under 'manage files'	⬆	✓ Closed
CRUC-433	REST API: Get review list based on predefined and custom filters	⬆	✓ Closed
CRUC-422	Create Schema Upgrade to remove foreign key constraint as reqd. by delete project work	⬆	✓ Closed
CRUC-419	Typo in email "reviewers are now complete"	▼	✓ Closed
CRUC-407	Minimal OSGi Infrastructure	⬆	✓ Closed
CRUC-404	Turn FE Off When only Crucible licence is present	⬆	✓ Closed
CRUC-402	Provide Plugin Authors with somewhere to store properties	⬆	✓ Closed
CRUC-401	Real Admin Users	⬇	✓ Closed
CRUC-400	Add Configuration UI to SVN Plugin, and Polish	⬆	✓ Closed
CRUC-399	Optionally Store Files for all FileRevisions	⬇	✓ Closed
CRUC-397	Stored FRX Create UI		✓

		⬆	Closed
CRUC-396	Stored FRX Data Impl	⬆	✓ Closed
CRUC-394	Add Light SCM Revisions to Reviews	⬆	✓ Closed
CRUC-392	Filesystem Light SCM Plugin	⬆	✓ Closed
CRUC-391	Default Repository for a Project can be a Light SCM repo	⬆	✓ Closed
CRUC-389	Light SCM Plugin instances appear in source/repository dropdowns	⬆	✓ Closed
CRUC-388	Subversion LSCM plugin	⬆	✓ Closed
CRUC-387	Confluence LSCM Plugin	⬆	✓ Closed
CRUC-386	Modify File Browser to Use Light SCM Plugins	⬆	✓ Closed
CRUC-385	Modify Changeset Browser to use Light SCM Plugins	⬆	✓ Closed
CRUC-384	Create Light SCM Module Type	⬆	✓ Closed
CRUC-381	Admin Pages for Plugins	⬆	✓ Closed
CRUC-373	Add General Comment via REST API	⬆	✓ Closed
CRUC-362	Return replies to comments via API	⬆	✓ Closed
CRUC-361	Project RSS feed	⬆	✓ Closed
CRUC-349	REST method /rest-service/reviews-v1/CR-1/comments returns HTTP 500 error	⬆	✓ Closed
CRUC-303	Add REST method to retrieve all reviews which involve a given file	⬆	✓ Closed
CRUC-302	Add REST methods to allow review creation	⬆	✓ Closed
CRUC-297	Upload files for review (not patches)	⬆	✓ Closed
CRUC-292	javax.xml.bind.JAXBException generated when invoking the getGeneralComments web service method	⬆	✓ Closed
CRUC-244	Add revisions not diffs to a review	⬆	✓ Closed
CRUC-227	Cannot delete projects		✓

		↑	Closed
CRUC-186	Allow an abandoned review to be deleted	↑	✓ Closed
CRUC-150	Allow the email address in the from section of the notification emails to be the user's actual email address	↑	✓ Closed
CRUC-18	Load All Users from Crowd/LDAP/etc	↑	✓ Closed

Crucible 1.6 Changelog

On this page:

- From 1.6.5.a to 1.6.6
- From 1.6.4 to 1.6.5.a
- From 1.6.3 to 1.6.4
- From 1.6.2.1 to 1.6.3
- From 1.6.2 to 1.6.2.1
- From 1.6.1 to 1.6.2
- From 1.6.0 to 1.6.1

From 1.6.5.a to 1.6.6

10 February 2009

This release updates the supporting libraries for Crucible plugins. This change enables the use of the new Git Crucible plugin for performing code reviews against a Git repository.

The Git plugin is not currently bundled with Crucible but may be downloaded from the Atlassian Maven repository here:
https://maven.atlassian.com/browse/com.atlassian.crucible.plugins/crucible-git-scm-plugin/1.0

The Git plugin should be considered an early access release. It allows reviews to be performed against a local Git repository clone. Note that the plugin does not update the cloned repository automatically. For more information on the Git plugin, please see the documentation.

We are very interested in any feedback users have on the Git Crucible plugin. Please post feedback in the Crucible forums.

Full list of issues fixed in this release:

JIRA Issues (2 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-1406	test	↑	✓ Closed
CRUC-900	GIT SCM Module	↑	✓ Closed

From 1.6.4 to 1.6.5.a

22 December 2008

This release contains a number of improvements and bug fixes.

Full list of issues fixed in this release:

JIRA Issues (6 issues)			

Key	Summary	Priority	Status
CRUC-947	Patch context displays Index Out of bounds exception	↑	✓ Closed
CRUC-938	doco that we require jdk 1.5.x and 1.6.0u4+	↑	✓ Closed
CRUC-928	Crucible 1.6.4 REST API fails under JavaSE 6	↑	✓ Closed
CRUC-886	Context lines can be duplicated with Patch diff options	↓	✓ Closed
CRUC-867	file names with "--" in their name will not be rendered properly in crucible reviews	↑	✓ Closed
CRUC-683	Add a "expand all unchecked" option to top of review screen	↓	✓ Closed

From 1.6.3 to 1.6.4

20 November 2008

This release contains bug fixes and minor improvements, and includes the new plugin points developed for AtlasCamp 2008.

Full list of issues fixed in this release:

JIRA Issues (27 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-926	GET /reviews-v1/<review id>/reviewitems/X fail to return reviewitems of certain types	↑	✓ Closed
CRUC-862	Crucible 1.6.4 documentation updates	↑	✓ Closed
CRUC-846	Publish new CAC pages on 1.6.4 release	↑	✓ Closed
CRUC-844	Cant update versions of new files with comments	↑	✓ Closed
CRUC-837	P4 Lite SCM plugin does not repect paths when listing changesets	↑	✓ Closed
CRUC-835	Reviews created in 1.6.2 with revisions of unknown content type (text) show as binary in 1.6.3 and never updated	↑	✓ Closed
CRUC-834	Different background color for user comments	↑	✓ Closed
CRUC-832	Can't add new revision of a file already under review if commented on - even if comment is deleted	↑	✓ Closed
CRUC-827	upgrade to hsql 1.8.0.10	↑	✓ Closed
CRUC-826	error when sending review to moderator	↑	✓ Closed
CRUC-821	change all references of 'summary' to 'all comments'	↑	✓ Closed
CRUC-820	Create JSR-170 (JackRabbit) SCM Plugin as Code Example	↓	✓ Closed
			✓

CRUC-809	File upload of a single text file gives ava.lang.Exception: Not enough revisions to diff	⬆	Closed
CRUC-802	File Upload from the Manage Files area drops the path info on submit	⬇	✓ Closed
CRUC-793	File in Review shows up as "Updating" and does not show source due to a Runtime Exception thrown by the SVNlite plugin	⬆	✓ Closed
CRUC-788	add REVISION_LINK and CHANGESET attributes to light scm plugins	⬆	✓ Closed
CRUC-787	No authentication required to retrieve user list in API	⬆	✓ Closed
CRUC-778	rework for CR-FE-696 FE-703 - making quicksearch respect quoted queries	⬆	✓ Closed
CRUC-765	When in Crucible, a AJAX timeout destroys the Crucible page	⬆	✓ Closed
CRUC-753	Deadlock in HSQL?	⬆	✓ Closed
CRUC-748	Add inline diff option bar to patch reviews	⬆	✓ Closed
CRUC-727	Configuring a plugin causes IE to hit 100% CPU and crashes the browser	⬆	✓ Closed
CRUC-582	Provide detailed information for cvs and light SCM repos via REST	⬆	✓ Closed
CRUC-568	Auto Suggest Reviewer	⬇	✓ Closed
CRUC-398	Stored FRX REST API	⬆	✓ Closed
CRUC-328	Reviewer Auto Suggest	⬇	✓ Closed
CRUC-255	Users lists unsorted when Allowed Participants restricted by Group	⬇	✓ Closed

From 1.6.2.1 to 1.6.3

5 November 2008

This release rolls together several improvements and bug fixes.

- Auto-save draft comments.
- Performance improvements when using Light SCM repositories.
- Bundle a Perforce Light SCM implementation.
- Various REST API improvements, including Conditional-Get support, improved error handling and revised review searching, which now allows any criteria to be omitted.
- JSON serialization has been added to the REST API, allowing the use of JSON in REST API calls. This feature is in an experimental state at present. Please report any issues discovered.

Please be aware of the upgrade notes regarding Light SCM repositories (this does not impact FishEye repositories):

- The configuration storage for the bundled File-system, Confluence and Subversion Light SCM plugins changed. Once you have upgraded to 1.6.3 you will need to re-add those repositories. Please read the Crucible 1.6.3 Upgrade Guide.
- The Light SCM plugin API was changed in this release. Light SCM plugins compiled against the old API will not work in this release of Crucible.

Full list of issues fixed in this release:

JIRA Issues (78 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-785	any screenshots of scroll to changeset for doco	↑	Closed
CRUC-784	filesystem lightscm plugin should not use "current" as the revision name	↑	Closed
CRUC-777	rework for CR-FE-697 CRUC-728	↑	Closed
CRUC-776	rework for CR-FE-702 CRUC-624	↑	Closed
CRUC-775	Create unit tests that verify JSON serialization	↑	Closed
CRUC-773	Add JSON support to Release Notes	↑	Closed
CRUC-772	Add documentation on how to use JSON to Confluence	↑	Closed
CRUC-771	Improve the l&f of manage files	↑	Closed
CRUC-769	Add JSON support to REST	↑	Closed
CRUC-768	Upgrade to Jersey 1.0, and include jersey-spring for possible future springification	↑	Closed
CRUC-767	Autosave race condition	↑	Closed
CRUC-764	Add this change to the release notes	↑	Closed
CRUC-763	DELETE operations in Crucible REST should return status 204 "No Content" to be more RESTful	↓	Closed
CRUC-759	Improve Confluence Light SCM Performance	↑	Closed
CRUC-758	make scroll to changeset look better	↑	Closed
CRUC-757	Custom filter object in Crucible REST should not use primitive values	↑	Closed
CRUC-752	[crucible] in the closed review email subject	↑	Closed

CRUC-750	Plugin config change will affect user configs	⬆	Closed
CRUC-749	Get Selenium Tests working	⬆	Closed
CRUC-747	Auto save draft comment needs to delete the draft on cancel	⬇	Closed
CRUC-744	summary email includes deleted comments	⬆	Closed
CRUC-742	Images in patches don't work	⬇	Closed
CRUC-739	Getting error popup on review due to confusion about a directory.	⬆	Closed
CRUC-736	Automatically save draft comments (autosave)	⬆	Closed
CRUC-735	Error message in Crucible file management is misplaced	⬇	Closed
CRUC-733	Crucible without FishEye still says FishEye in titles	⬆	Closed
CRUC-729	Can't review binary file becoming textual	⤒	Closed
CRUC-728	add CSID and SOURCE_URL to lightSCM details	⬇	Closed
CRUC-725	Light SCM allows creation of repos with the same name as Fisheye repos	⤒	Closed
CRUC-724	Implement Light SCM plugin for Perforce	⬆	Closed
CRUC-723	Converting Crucible reviews from FishEye repo to LightSCM SVN fails to load revisions	⬆	Closed
CRUC-722	Remove the use of deprecated CrucibleRevision.getSource()	⬆	Closed
CRUC-713	Change REST return code from 200 to 201 "Created" for several POST actions	⬇	Closed
CRUC-712	Create unit tests for REST	⬆	Closed
CRUC-711	document Alt+Click for selecting review text	⬆	Closed
CRUC-710	You do not have permission to see all the search results, seen in (my) to summarise and out for review	⬇	Closed
CRUC-709	Refactor exception handling issues	⬇	Closed
CRUC-708	Change REST filter retrieval from POST to GET		

		⬆	Closed
CRUC-706	Retrieving non-existing metrics version in REST gives 500 "Internal Server Error", should be 404	⬆	✓ Closed
CRUC-705	Create RestXxxServices via Spring	⬇	✓ Closed
CRUC-702	summary email documentation	⬆	✓ Closed
CRUC-701	Support Perforce repositories in RepositoryService	⤒	✓ Closed
CRUC-700	add maybe-details and maybe-filehistory to FileSummary	⬆	✓ Closed
CRUC-699	Revision Details should be a map	⬆	✓ Closed
CRUC-698	maybe-details provided by SCMs are not used	⬆	✓ Closed
CRUC-697	Change ManageFiles tab so that it does not require so much information from the SCM	⬆	✓ Closed
CRUC-696	Address performance problems in LightSCM plugin API	⬆	✓ Closed
CRUC-695	Crucible REST should throw an exception when adding changesets to reviews that already have comments	⬆	✓ Closed
CRUC-694	Adding changesets on open reviews messes up the in-line comments	⬆	✓ Closed
CRUC-693	Still cannot delete projects	⬆	✓ Closed
CRUC-691	Implement Conditional Get in REST API	⬆	✓ Closed
CRUC-689	Revision details missing from choose diff dropdown	⬇	✓ Closed
CRUC-688	FR_EXTRA.FRX_ORDER needs a unique constraint	⬆	✓ Closed
CRUC-687	<i>no comment</i> commit message tooltip	⬇	✓ Closed
CRUC-680	Changeset dates are wrong	⬆	✓ Closed
CRUC-679	Add xwork action that returns the text summary (for copying and pasting).	⬆	✓ Closed
CRUC-678	Add send summary button and form	⬆	✓ Closed
CRUC-677	Update summary template to include comments	⬆	✓ Closed

CRUC-671	Create review from changeset gives HTTP 500	↑	✓ Closed
CRUC-670	Improve REST error reporting (HTTP return codes)	↑	✓ Closed
CRUC-667	re address update of revision details	↓	✓ Closed
CRUC-665	Correct documentation for REST API for getting file information	↓	✓ Closed
CRUC-658	Scroll To: box for changelogs	↑	✓ Closed
CRUC-656	Previously deleted files show as having an old version in a review when they are added again	↓	✓ Closed
CRUC-655	Downloaded files do not have the correct file name	↓	✓ Closed
CRUC-653	[admin] adding a "default reviewer" incorrectly adds an "allowed reviewer" in some cases	↑	✓ Closed
CRUC-627	make stored reviews viewable when the source isn't available	↑	✓ Closed
CRUC-624	XML Parsing Error from Crucible Review Service using allReviews filter	↑	✓ Closed
CRUC-536	Improper code colorization for C++	↓	✓ Closed
CRUC-529	Have a way to select text in source windows	↓	✓ Closed
CRUC-509	Add info about IDE integration in Web UI	↑	✓ Closed
CRUC-486	Invalid rendering of Search Comments report (table part)	⇑	✓ Closed
CRUC-475	Create documentation on creating reviews using remote API	↓	✓ Closed
CRUC-470	Summary Email Improvements	↑	✓ Closed
CRUC-395	Stored FRX	↑	✓ Closed
CRUC-383	Maven Archetype for plugin developers	↑	✓ Closed
CRUC-380	Light SCM	↑	✓ Closed
CRUC-327	Copy and paste is not working	↑	✓ Closed

From 1.6.2 to 1.6.2.1

24 October 2008

This release fixes a problem in 1.6.2 when running Crucible on Windows. Due to a file-lock issue, the upgrade script in 1.6.2 could not start.

- CRUC-781 Upgrade from 1.6 to 1.6.2 fails for windows machines.

From 1.6.1 to 1.6.2

21 October 2008

This release fixes a bug in the way Crucible stores review data. This bug was introduced in Crucible 1.6.0. **We strongly recommend all 1.6.0, 1.6.0-beta and 1.6.1 users immediately upgrade to this release.**

If this bug occurs in your Crucible instance, you will find that review data created after that point will be corrupt. If you find that is the case please contact Crucible support for assistance.

- CRUC-743 Switch from CACHED tables mode back to memory table.

From 1.6.0 to 1.6.1

24 September 2008

This is a bug fix release.

- Crowd 1.3 users will need to upgrade to Crowd 1.4.4 or later due to an incompatibility with this version of Crucible.

- CRUC-673 NPE when viewing a review.

- CRUC-674 NPE when closing a review.

Crucible 1.6.3 Upgrade Guide

Upgrade Notes

- The configuration storage for the bundled File-system, Confluence and Subversion Light SCM plugins changed. Please follow the procedure below.
- The Light SCM plugin API was changed in this release. Light SCM plugins compiled against the old API will not work in this release of Crucible.

Upgrade Procedure

Due to a configuration change, you will need to delete and re-add your LightSCM repositories.

1. Before you shut down Crucible, take a note of your Light SCM configuration. You can view this configuration in the **Repository List** Admin page, in the **LightSVN**, **Confluence** and **File System** sections.
2. Follow the general instructions on upgrading Crucible.
3. While Crucible is shut down, delete the `confluence`, `svn` and `filesystem` config files in `FISHEYE_INST/var/plugins/user`.
4. Once Crucible has been restarted, re-add the Light SCM repositories from step 1.

Crucible 1.5 Release Notes

14 April 2008

Atlassian presents Crucible 1.5

Crucible release 1.5 brings new enhancements that make your code review activities quicker and easier. The all-new per-project page consolidates the display of work done on a particular goal or product, while filtered search for defects and comments provides rapid access to Crucible content that you need to see, now.

Highlights of this release:

- Project Dashboard
- Filtered comments & defects search, with statistical summary
- Customisable email templates
- Improvements to Crucible Plugin API beta
- Plus numerous improvements and bug-fixes

DOWNLOAD latest version

Upgrading to Crucible 1.5

You can now download Crucible from here. If upgrading from a previous version, please follow the Upgrade Guide.

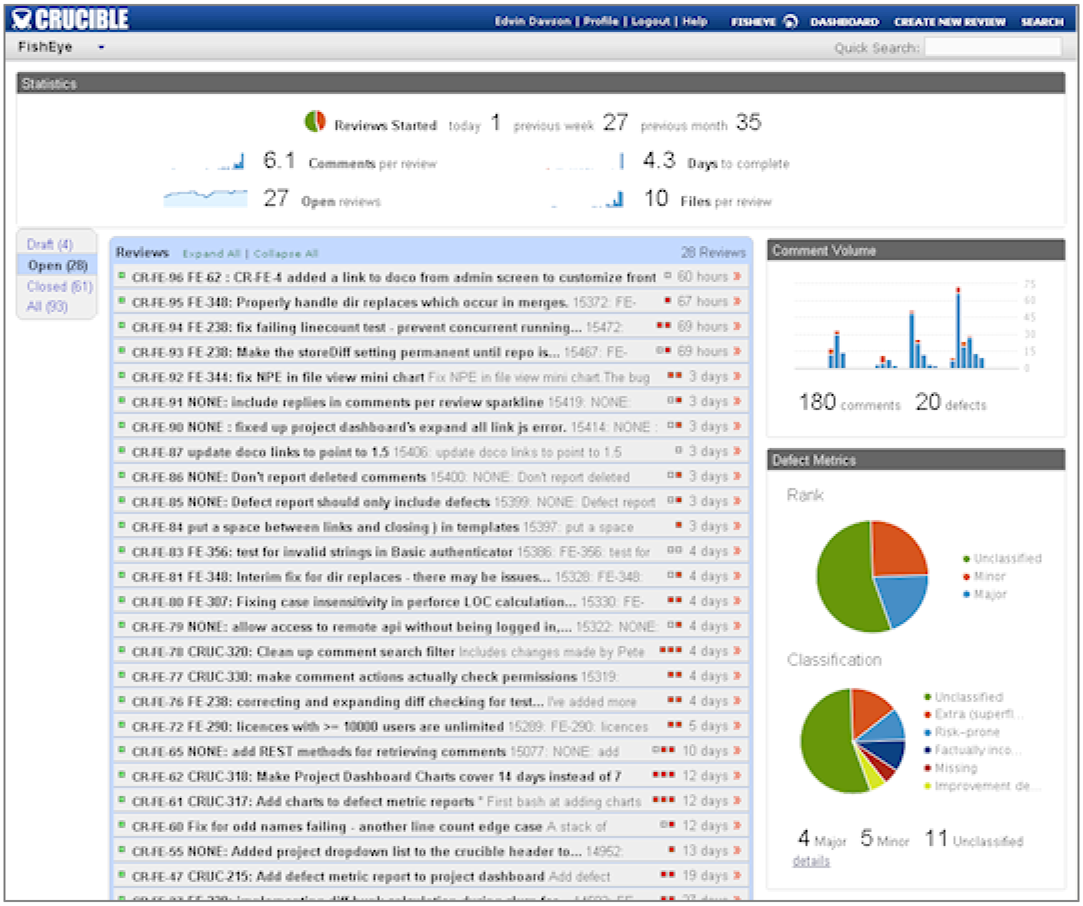
Highlights of Crucible 1.5

1

Project Dashboard

Crucible 1.5 introduces the Project Dashboard, which allows you to see open reviews that belong to a given project, presented with additional project-related data and graphs.

Screenshot: Crucible Project Dashboard



Reviews Started today 1 previous week 27 previous month 35

6.1 Comments per review

4.3 Days to complete

27 Open reviews

10 Files per review

Draft (4)
Open (28)
Closed (61)
All (93)

Reviews Expand All | Collapse All 28 Reviews

- CR-FE-96 FE-62 : CR-FE-4 added a link to doco from admin screen to customize front 60 hours
- CR-FE-95 FE-348: Properly handle dir replaces which occur in merges. 15372: FE- 67 hours
- CR-FE-94 FE-238: fix failing linecount test - prevent concurrent running... 15472: 69 hours
- CR-FE-93 FE-238: Make the storeDiff setting permanent until repo is... 15467: FE- 69 hours
- CR-FE-92 FE-344: fix NPE in file view mini chart Fix NPE in file view mini chart The bug 3 days
- CR-FE-91 NONE: include replies in comments per review sparkline 15419: NONE: 3 days
- CR-FE-90 NONE : fixed up project dashboard's expand all link js error. 15414: NONE : 3 days
- CR-FE-87 update doco links to point to 1.5 15406: update doco links to point to 1.5 3 days
- CR-FE-86 NONE: Don't report deleted comments 15400: NONE: Don't report deleted 3 days
- CR-FE-85 NONE: Defect report should only include defects 15399: NONE: Defect report 3 days
- CR-FE-84 put a space between links and closing) in templates 15397: put a space 3 days
- CR-FE-83 FE-356: test for invalid strings in Basic authenticator 15386: FE-356: test for 4 days
- CR-FE-81 FE-348: Interim fix for dir replaces - there may be issues... 15328: FE-348: 4 days
- CR-FE-80 FE-307: Fixing case insensitivity in perforce LOC calculation... 15330: FE- 4 days
- CR-FE-79 NONE: allow access to remote api without being logged in,... 15322: NONE: 4 days
- CR-FE-78 CRUC-320: Clean up comment search filter Includes changes made by Pete 4 days
- CR-FE-77 CRUC-330: make comment actions actually check permissions 15319: 4 days
- CR-FE-76 FE-238: correcting and expanding diff checking for test... I've added more 4 days
- CR-FE-72 FE-290: licences with >= 10000 users are unlimited 15289: FE-290: licences 5 days
- CR-FE-65 NONE: add REST methods for retrieving comments 15077: NONE: add 10 days
- CR-FE-62 CRUC-318: Make Project Dashboard Charts cover 14 days instead of 7 12 days
- CR-FE-61 CRUC-317: Add charts to defect metric reports * First bash at adding charts 12 days
- CR-FE-60 Fix for odd names failing - another line count edge case A stack of 12 days
- CR-FE-55 NONE: Added project dropdown list to the crucible header to... 14952: 13 days
- CR-FE-47 CRUC-215: Add defect metric report to project dashboard Add defect 19 days

180 comments 20 defects

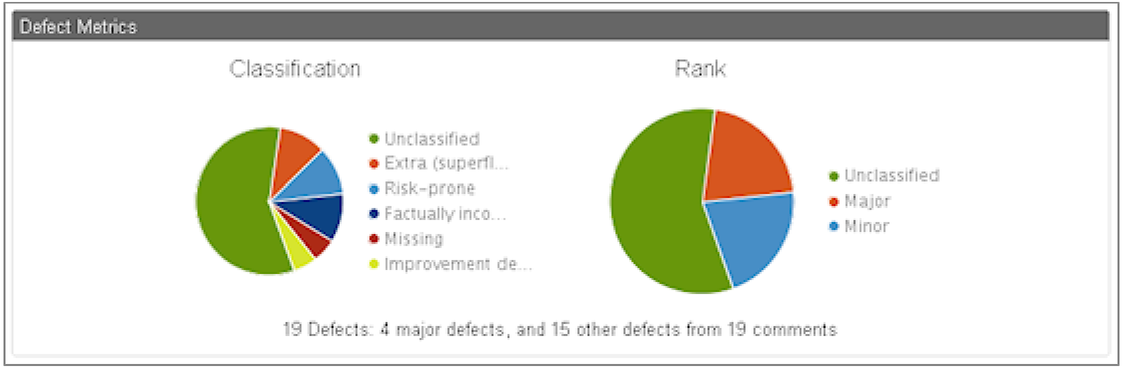
4 Major 5 Minor 11 Unclassified
details

2

Filtered comments & defects search, with statistical summary

Defects and comments are now searchable, easing the difficulty of finding a particular piece of work or revision (and its relevant comments). These search results now also show a very useful statistical summary. Also, a new defect metrics report is available.

Screenshot: Crucible Defect Metrics Report



19 Defects: 4 major defects, and 15 other defects from 19 comments

3

Customisable email templates

You can now customise the content and appearance of email notifications that get sent to Crucible users. For example you can append a legal disclaimer, alter the subject line or provide custom header text for all messages.

4

Improvements to Crucible Plugin API beta

Now with REST support and the ability to upload patches, the Crucible Plugin API beta is for Crucible integrators who want to extend Crucible to interoperate with their enterprise infrastructure or processes.

Plus numerous improvements and bug-fixes

JIRA Issues (38 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-344	Defect pie chart on comment search page includes non-defect comments.	↑	Closed
CRUC-332	create a better looking indication icon	↑	Closed
CRUC-324	Add Project Dropdown to Search Pages	↑	Closed
CRUC-323	Create 'Reviews Started' Pie Chart for Project Dashboard	↑	Closed
CRUC-318	Make Project Dashboard Charts cover 14 days instead of 7	↑	Closed
CRUC-314	sidebyside diff line numbers must be aligned to the top	↑	Closed
CRUC-307	Optimize chart queries	↑	Closed
CRUC-299	link to project page	↑	Closed
CRUC-290	Warnings during backup: File does not exist /data/crucible/inst/var/data/crudb/crucible.lck & File does not exist /data/crucible/inst/var/data/crudb/crucible.log	↓	Closed
CRUC-289	permissions are not checked in rss feeds (but are in list html display)	↑	Closed
CRUC-285	"project project" in Permission Screen in admin	↑	Closed
CRUC-283	Escape HTML in Javascript properly	↑	Closed
CRUC-282	Crucible fails to upload patch file with file information that contain blank spaces between file path and timestamp	↓	Closed
CRUC-280	remove debug log from Process Notification	↑	Closed

CRUC-266	let the subject of the email be modifiable via the notification template	⬇	Closed
CRUC-260	How to customize the email notification content	⬇	Closed
CRUC-248	Configurable email subject line	⬇	Closed
CRUC-247	Flag files which have been modified since a review was created	⬆	Closed
CRUC-246	Parentheses should be valid URL characters	⬆	Closed
CRUC-242	Add RPC call to create review from patch	⬆	Closed
CRUC-241	Incorrect dates in RSS feeds	⬇	Closed
CRUC-240	Checking box in "manage files" -> "changesets" spins forever	⬆	Closed
CRUC-238	Perforce unit tests do not work on Windows platform	⬇	Closed
CRUC-232	Address all issues outline here	⬆	Closed
CRUC-230	Crucible allows circular review linking and can't delete a review link	⬇	Closed
CRUC-226	Title wrong for post-approval in manage files tab	⬇	Closed
CRUC-220	permalink for a comment does not work when summarize mode is active	⬆	Closed
CRUC-212	Create Project Dashboard page	⬆	Closed
CRUC-206	URL recognition in Crucible comment does not always match whole string	⬆	Closed
CRUC-201	Allow project to be specified on create review URL	⬆	Closed
CRUC-194	Documentation: restoring Crucible backup data	⬆	Closed
CRUC-184	When Creating Review From FishEye, Default to Right Project	⬆	Closed
CRUC-172	Create 'Getting started with Crucible' document as part of User Guide revisions	⬆	Closed
CRUC-162	From field on emails is incorrect or a least deceiving	⬇	Closed
CRUC-147	Add Permissions Checks to All Operations on Reviews	⬆	Closed

CRUC-128	email notification of review closure prints literal "null" for absent field value	↑	✓ Closed
CRUC-85	Sometimes we need to see differences in UNICODE files.	↑	✓ Closed
CRUC-21	change icon for "View History In Fisheye"	↑	✓ Closed

Crucible 1.5 Changelog

On this page:

- From 1.5.3 to 1.5.4
- From 1.5.2 to 1.5.3
- From 1.5.1 to 1.5.2
- From 1.5.0 to 1.5.1

From 1.5.3 to 1.5.4

1 August 2008

This release contains minor improvements and bug fixes.

JIRA Issues (10 issues)

Key	Summary	Priority	Status
CRUC-638	How to (easily) add specific changeset to a review	↑	✓ Closed
CRUC-633	How to remove a reviewer from a review	↓	✓ Closed
CRUC-538	Comments weirdly duplicated on occasion	↓	✓ Closed
CRUC-513	If we display no change on a revision, comments made are hidden by default.	↑	✓ Closed
CRUC-503	Upgrade to crowd-integration-client 1.4.4	↑	✓ Closed
CRUC-498	Draft comment replies are deleted when completing review	↑	✓ Closed
CRUC-454	two notifcation emails sent	↑	✓ Closed
CRUC-429	Prevent two projects with same key different case	↑	✓ Closed
CRUC-417	large var/data/crudb/crucible.log	↓	✓ Closed
CRUC-284	AnnotatorTag error: Could not get blame for file	↑	✓ Closed

From 1.5.2 to 1.5.3

23 June 2008

This release contains bug fixes.

JIRA Issues (1 issues)

Key	Summary	Priority	Status
CRUC-462	"File closed: var\data\data0.bin" after backups	↑	✓ Closed

From 1.5.1 to 1.5.2

27 May 2008

This release contains bug fixes.

JIRA Issues (16 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-434	'Expand Commented' appears when there are general comments but no revision comments	↓	Closed
CRUC-413	Crucible always creates a new default project after it is deleted	↓	Closed
CRUC-406	Remove 'All Projects' from projects dropdown	↓	Closed
CRUC-405	[Performance] AnnotatorTag can hang in sort loop for (perhaps) large diffs	↑	Closed
CRUC-376	Prevent deletion of default project	↓	Closed
CRUC-371	Comment volume chart reports confusion	↓	Closed
CRUC-370	create action to remove all revisions from a review	↑	Closed
CRUC-363	Remove all Revisions in manager files-> Changesets tab doesn't work	↑	Closed
CRUC-358	Height not respected if less than 144 in defect chart	↓	Closed
CRUC-355	Review Stats Wrong	↑	Closed
CRUC-354	Personal dashboard stats don't match	↑	Closed
CRUC-336	can't search for review keys in quicksearch	↑	Closed
CRUC-335	[CSS] project list a little wonky in IE	↓	Closed
CRUC-262	Urgent : SVNRepository methods are not reenterable	❗	Closed
CRUC-233	Changing or expanding the term - 'Approve'	↓	Closed
CRUC-185	Abandoned Reviews Still Show in FishEye	↑	Closed

From 1.5.0 to 1.5.1

24 April 2008

This release contains bug fixes.

JIRA Issues (9 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-367	LOC exceptions filling Debug log to the tune of 3-6GB per day	↑	✓ Closed
CRUC-364	Starting Crucible 1.5 Spits Lots of Errors Out	↥	✓ Closed
CRUC-357	Sparklines have wrong content type	↓	✓ Closed
CRUC-353	Crucible seems not to be able to handle uploaded git patches	↑	✓ Closed
CRUC-350	javax.servlet.ServletException	↑	✓ Closed
CRUC-333	In IE, "view diff to latest" partially obscured	↓	✓ Closed
CRUC-304	Add REST method to retrieve review items for a given review	↑	✓ Closed
CRUC-291	NullPointerException thrown when invoking the getAllRevisionComments web service method	↑	✓ Closed
CRUC-161	Create document stating when a customer purchases Crucible that they do not need to have the latest version of FishEye	↑	✓ Closed

Crucible 1.2 Release Notes

December 5, 2007.

The Atlassian Crucible team is delighted to present Crucible 1.2.
Crucible release 1.2 brings you a host of popular new features. You can now group your reviews into projects (similar to JIRA projects) and authorise your users via project permission schemes.

New user management screens make the administrator's job a lot easier. The new built-in integration with Atlassian Crowd extends your authentication and authorisation capabilities. You can now include users and groups from one or more Crowd directories, and provide single sign-on (SSO) across Atlassian products plus any other applications that support SSO.

Crucible's integration with JIRA and FishEye is now closer than ever before. Read the details below.

Highlights of this release:

- Reviews grouped into projects
- Customisable permission schemes

- Plugin API
- Enhancements to user management
- JIRA integration
- Crucible 1.2 includes FishEye 1.4
- Plus over 20 improvements and bug-fixes

Responding to your feedback:

8 new feature requests/improvements implemented

9 votes satisfied

Your [votes and issues|http://jira.atlassian.com/browse/CRUC] help us keep improving our products, and are much appreciated.

Upgrading to Crucible 1.2

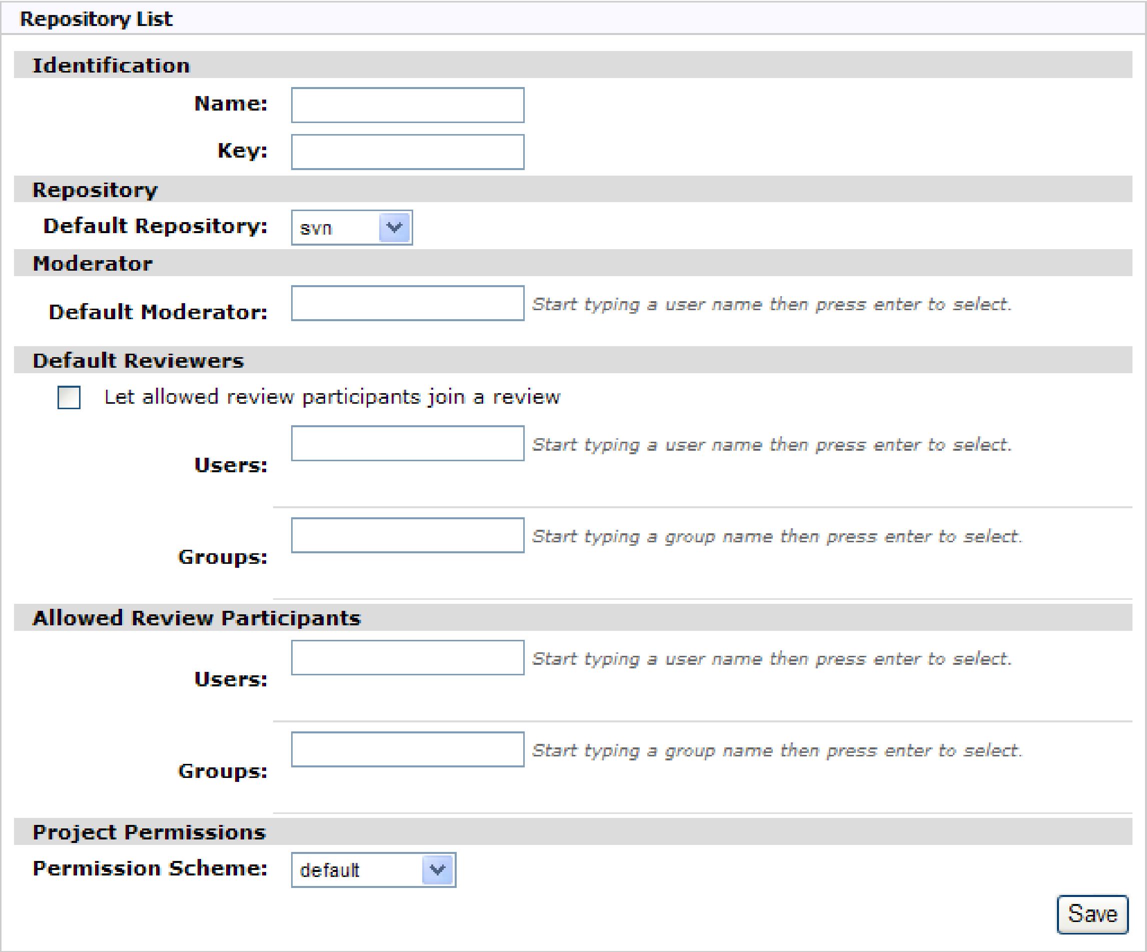
You can now download Crucible from here. If upgrading from a previous version, please follow the Upgrade Guide.

Highlights of Crucible 1.2

1

Reviews grouped into projects

- Crucible now supports projects - every review will belong to a project.
- Each project has a unique key (prefix), modelled on JIRA keys.
- You can add your own projects via the new administration screens.
- You can specify review defaults per project, such as the default users for each role and a default repository.
- And you can restrict the users/groups who can perform a particular role, e.g. only team leaders can be moderators.
- Each project has its own permission scheme (see below).



Repository List

Identification

Name:

Key:

Repository

Default Repository: svn

Moderator

Default Moderator: *Start typing a user name then press enter to select.*

Default Reviewers

□ Let allowed review participants join a review

Users: *Start typing a user name then press enter to select.*

Groups: *Start typing a group name then press enter to select.*

Allowed Review Participants

Users: *Start typing a user name then press enter to select.*

Groups: *Start typing a group name then press enter to select.*

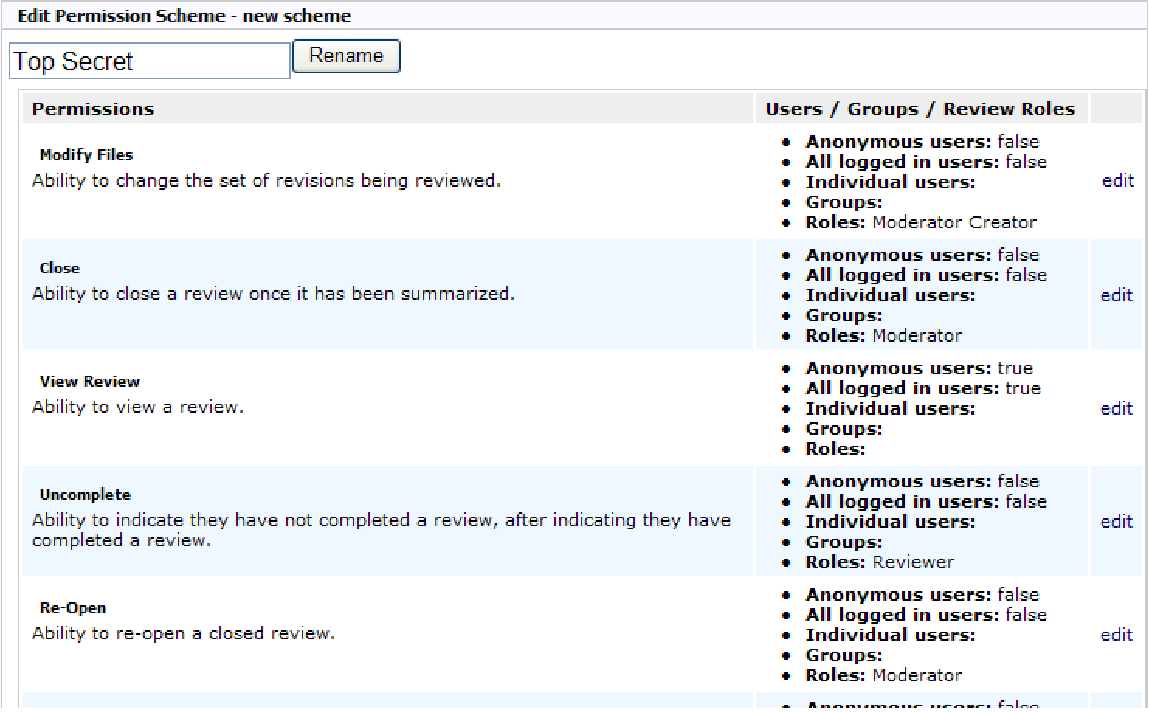
Project Permissions

Permission Scheme: default

2

Customisable permission schemes

- A permission scheme is a set of actions which a user can perform (e.g. create a review, approve a review, etc).
- Each project can have its own custom permission scheme — or you can use the same scheme for multiple projects.
- The permission scheme for a review is determined by the review's project.



Edit Permission Scheme - new scheme

Top Secret

Permissions	Users / Groups / Review Roles	
Modify Files Ability to change the set of revisions being reviewed.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Roles:** Moderator Creator	edit
Close Ability to close a review once it has been summarized.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Roles:** Moderator	edit
View Review Ability to view a review.	**Anonymous users:** true **All logged in users:** true **Individual users:** **Groups:** **Roles:**	edit
Uncomplete Ability to indicate they have not completed a review, after indicating they have completed a review.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Roles:** Reviewer	edit
Re-Open Ability to re-open a closed review.	**Anonymous users:** false **All logged in users:** false **Individual users:** **Groups:** **Roles:** Moderator	edit
	Anonymous users: false	

3

Plugin API

- A new plugin Crucible programming interface (API), in **beta** for this release, supports the following functionality:
  - Create or modify reviews and comments.
  - Add files, patches, etc to reviews.
  - Invoke state transitions.
  - Add custom servlet handlers.
- More information.

4

Enhancements to user management

In Crucible 1.1.2, we introduced support for public signup (self-registration). Now in Crucible 1.2:

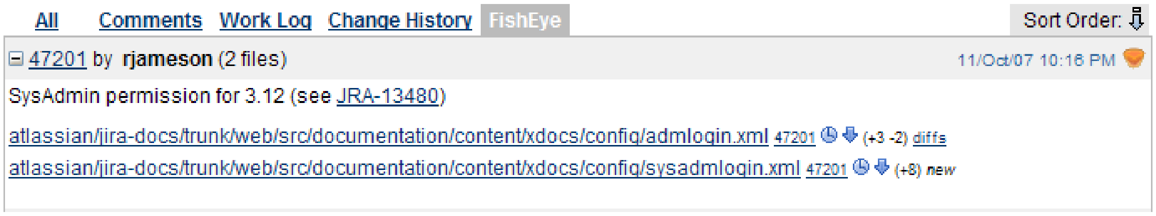
- Administrator can make the email address for self-signups optional.
- Improved user interface makes user administration easier.
- Groups are supported.
- Read the FishEye documentation.

5

JIRA integration

The new version 1.2 of the FishEye-for-JIRA plugin includes some useful improvments:

- new 'FishEye' tab for JIRA issues and projects
- improved ability to create a Crucible review from the 'FishEye' tab within a JIRA issue — just click the Crucible icon:



⊟ 47201 by rjameson (2 files) 11/Oct/07 10:16 PM

SysAdmin permission for 3.12 (see JRA-13480)

atlassian/jira-docs/trunk/web/src/documentation/content/xdocs/config/admlogin.xml 47201 (+3 -2) diffs

atlassian/jira-docs/trunk/web/src/documentation/content/xdocs/config/sysadmlogin.xml 47201 (+8) new

6

Crucible 1.2 includes FishEye 1.4

... and provides closer integration than ever before.

- FishEye screens include links to existing Crucible reviews. So you can see which files/changesets have been reviewed.
- EyeQL allows you to search for Crucible data. For example, you can search for files that have not yet been reviewed.
- Crucible now has built-in Crowd/SSO support.
- See the FishEye 1.4 Release Notes.

7

Plus over 20 improvements and bug-fixes

JIRA Issues (32 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-181	Old screenshot in 1.2 release notes	⇧	✓ Closed
CRUC-175	LHS abandon button on Crucible review screen broken	⇧	✓ Closed
CRUC-166	Some files not work	⇧	✓ Closed
CRUC-152	sysinfo admin screen should show both CRU and FE license string	⇧	✓ Closed
CRUC-146	Add review information to ALL search result pages	⇧	✓ Closed
CRUC-143	add review-constraints in the EyeQL where clause	⇧	✓ Closed
CRUC-140	ensure "linked" reviews are exported thru data in remote api	⇧	✓ Closed
CRUC-122	Move review to another project	⇧	✓ Closed
CRUC-121	EyeQL return clause: reviews	⇧	✓ Closed

CRUC-115	Accept Patch Review From Clipboard	⬇	✓ Closed
CRUC-113	Allow creation of reviews for multiple changesets through /cru/create URL	⬆	✓ Closed
CRUC-109	Presented with Post/delete drafts buttons when no comments drafted made	⬆	✓ Closed
CRUC-107	Ability to change username	⬆	✓ Closed
CRUC-104	AnnotatorTag error reviewing patch	⬆	✓ Closed
CRUC-102	Create Project Model object and Hibernate DB upgrade	⬆	✓ Closed
CRUC-101	Add 'Projects'	⬆	✓ Closed
CRUC-95	Change Diff Buttons Losing State	⬇	✓ Closed
CRUC-94	Show Existing Reviews In Fisheye	⬇	✓ Closed
CRUC-93	Crucible should preserve request params/URLs through login redirects	⬆	✓ Closed
CRUC-90	More Administrator Options	⬆	✓ Closed
CRUC-89	Add 'Default Reviewers'	⬆	✓ Closed
CRUC-88	Review Groups	⬆	✓ Closed
CRUC-86	add "allow anyone" as a per-project default	⬆	✓ Closed
CRUC-80	Should be able to create a review when there are no configured repositories	⬆	✓ Closed
CRUC-78	merge cru/fe src and content trees	⬆	✓ Closed
CRUC-73	Generate HEAD review from directory	⬆	✓ Closed
CRUC-68	Dragging to deselect source lines no longer works	⬆	✓ Closed
CRUC-65	beta plugin api	⬆	✓ Closed
CRUC-61	Author should be able to "complete"	⬇	✓ Closed
CRUC-56	Self-registration	⬆	✓ Closed
CRUC-43	Webservice API for Reviews	⬆	✓ Closed
CRUC-36	Should my review status go back to incomplete if I start adding comments?	⬇	✓ Closed

Crucible 1.2 Changelog

On this page:

- From 1.2.2 to 1.2.3
- From 1.2.1 to 1.2.2
- From 1.2 to 1.2.1

From 1.2.2 to 1.2.3

7 February 2008

This release contains bug fixes (including those in from FishEye 1.4.3).

JIRA Issues (26 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-274	Create IDE integration with IntelliJ IDEA	⬆	✓ Closed

CRUC-273	Create IDE integration with Eclipse	↑	✓ Closed
CRUC-258	Allow all users to be able to a "Moderator" or "Author" in a review project	↓	✓ Closed
CRUC-256	Check boxes for checking off files reviewed in a review are lost if navigate to raw text by "download raw text" icon	↑	✓ Closed
CRUC-249	Accept review comments in Japanese (unicode characters)	↑	✓ Closed
CRUC-243	race condition between crucible backup and repository scan	↑	✓ Closed
CRUC-235	File tabs should be kept open	↑	✓ Closed
CRUC-210	Create JIRA issues from code reviews	↑	✓ Closed
CRUC-208	Nullpointer exception when creating new review	↑	✓ Closed
CRUC-202	update 'Repositories' screenshot	↑	✓ Closed
CRUC-196	Error on attempting to create a Review.	↑	✓ Closed
CRUC-180	'Show Full Source' should continue to highlight the diff	↑	✓ Closed
CRUC-177	Comments on reviews are being sent but don't appear in the review when viewed online	↑	✓ Closed
CRUC-165	Folder Issue	↑	✓ Closed
CRUC-139	Can't add new versions of the same file to a review	↑	✓ Closed
CRUC-127	Improperly stopping crucible causes data loss!	❗	✓ Closed
CRUC-112	Point online help links to new Crucible doc space	↑	✓ Closed
CRUC-106	Add content to new documentation page ' Crucible and FishEye'	↑	✓ Closed
CRUC-105	Upload PDF, XML and HTML versions of Crucible docs	↑	✓ Closed
CRUC-103	Create admin page for Projects	↑	✓ Closed
CRUC-100	Move Crucible docs to Confluence	↑	✓ Closed
CRUC-99	Feature request: I would like to see statistics about number of major and minor issues found per developer.	↑	✓ Closed

CRUC-92	Allow Anonymous Access Per Repository	↑	Closed
CRUC-84	Ability to create a review directly from JIRA	↑	Closed
CRUC-50	Allow review of entire file, not a particular diff	↑	Closed
CRUC-28	After Crucible forums are moved, update all links in docs	↑	Closed

From 1.2.1 to 1.2.2

This release contains some minor improvements and bug fixes.

- Trusted Application Support
  FishEye/Crucible now allows you to set up trusted communications with other Atlassian applications. At this point, the JIRA FishEye plugin supports Trusted Applications. The JIRA FishEye plugin can request information from FishEye on behalf of the currently logged-in user, and FishEye will not ask the user to log in again or to supply a password. Previously FishEye/Crucible would have used a single 'system' account to determine permissions. Now, FishEye/Crucible can apply the correct permission settings for the logged-in user.

- Hyphens are now allowed in project key names.

JIRA Issues (3 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-207	allow hyphens in Crucible project keys	↑	Closed
CRUC-204	Add project to dashboard filter	↑	Closed
CRUC-195	groups access control to repositories failed after configured External authentication (LDAP)	↑	Closed

From 1.2 to 1.2.1

This is a small bug-fix release.

JIRA Issues (6 issues)			
Key	**Summary**	**Priority**	**Status**
CRUC-199	Top of project edit form says Repositories List	↑	Closed
CRUC-198	multiple copies of revisions added to review if you add changesets in reverse chronological order	↑	Closed
CRUC-182	Projects Don't Persist Through Pages	↑	Closed
CRUC-179	Support addition of multiple revisions of the same file that appears in multiple changesets	↑	Closed
CRUC-178	Hit NPE when perform backup		

		⬆	Closed
CRUC-40	Getting wrong changeset number causes diff to be missing	⬆	✓ Closed

Crucible 1.2 Upgrade Guide

Upgrade Notes

- During the upgrade, a default project and default permission scheme will be created. All existing reviews will be assigned to the default project.

Upgrade Procedure

- Please read the Release Notes for the version you are upgrading to, as well as any versions you are skipping.
- Follow the instructions on upgrading Crucible.

Crucible 1.1 Release Notes

> ⚠ Crucible 2.1 has now been released. Read the Release Notes.

Crucible 1.1 allows pre-commit (patch) reviews, side-by-side diff mode, syntax highlighting in diffs, and many other bug fixes and improvements.

Upgrading Crucible

You can now download Crucible from here. Information on installing Crucible can be found here. If upgrading from a previous version, please follow the Upgrade Guide.

Highlights of Crucible 1.1

- Pre-commit review (patch review).
- Progress tracking through a review by marking each file as 'done'.
- Side-by-side diff mode within the review display.
- Syntax highlighting when displaying a diff.
- Many small UI fixes and improvements. Refer to the changelog for more details.

Crucible 1.1 Changelog

On this page:

- From 1.1.3 to 1.1.4
- From 1.1.2 to 1.1.3
- From 1.1.1 to 1.1.2
- From 1.1 to 1.1.1

From 1.1.3 to 1.1.4

This release updates the included FishEye component and includes a number of performance improvements and bug fixes for Subversion and Perforce repository indexing.

From 1.1.2 to 1.1.3

This release fixes a bug CRUC-104 that prevented Crucible from correctly displaying large patches.

From 1.1.1 to 1.1.2

This release adds some new user-related functions and includes bug fixes.

New Features

- You can now allow users to create their own user accounts (sign-up).
- You can now allow anonymous browsing of reviews.
- Users can now add themselves as a reviewer ('Join a review'). This is an option that is configured per review.

Bug fixes

- Fix problem where Crucible would only display the top part of each diff in a patch.
- Fix various JavaScript and UI errors.
- Fix various IE6 and IE7 problems.
- Fix problem where some users where redirected to /uar/browser.css after login.

From 1.1 to 1.1.1

This is a small bug-fix release. It addresses a stack-overflow problem for some configurations.

Crucible 1.1 Upgrade Guide

Upgrade Notes

- As of version 1.0, Crucible now requires a JVM version 1.5 or later. Previously, 1.4+ was required.
- Crucible 1.1.4 includes FishEye 1.3.8.
- Upgrading from 1.0.4 (or earlier) will force a complete re-index of P4 repositories.

Upgrade Procedure

- Please read the Release Notes and Upgrade Guides for the version you are upgrading to, as well as any versions you are skipping.
- Follow the instructions on upgrading Crucible.

Crucible Release Summary

Crucible 2.1 (12-Nov-09)

- Wiki Markup Rendering
- Progress Tracking
- Usability and Productivity Updates
- Streamlined JIRA Integration
- Review Time Tracking
- Review History Dialog
- "Blockers" Reports
- Threaded Comments
- Plugin Developer Tools
- More in release notes.

Crucible 2.0 (30-Jun-09)

- Support for iterative reviews
- New User Interface
- Indicators for read/unread comments
- Enhanced JIRA integration
- More in release notes.

Crucible 1.6 (23-Sep-08)

- Support for non-FishEye repositories
- Confluence page reviews
- Shared file system repositories
- Enhanced pre-commit reviews & image support
- Multiple admin users
- Expanded API
- More in release notes.

Crucible 1.5 (14-Apr-08)

- Project Dashboard
- Filtered comments & defects search, with statistical summary
- Customisable email templates
- Improvements to Crucible Plugin API beta
- More in release notes.

Crucible 1.2 (5-Dec-07)

- Reviews grouped into projects
- Customisable permission schemes
- Plugin API
- Enhancements to user management
- JIRA integration
- Crucible 1.2 includes FishEye 1.4
- More in release notes.

Crucible 1.1 (18-Sep-07)

- Pre-commit review (patch review)
- Review participants can keep track of their progress through a review by marking each file as "done"
- Side-by-side diff mode within the Review display
- Syntax highlighting when displaying a diff
- More in release notes.

Crucible FAQ

Crucible FAQ

Answers to frequently asked questions about configuring and using Crucible.

- Troubleshooting
  - Crucible freezes unexpectedly
  - JIRA Integration Issues
  - Problems with very long comments and MySQL migration
- General FAQs
  - Can Crucible be run as a Windows Service?
  - Can I deploy Crucible or FishEye as a WAR?
  - How to Automate Daily Crucible Backups
- Licensing FAQ
  - What happens if I decide to stop using FishEye with Crucible?
  - Do I need a FishEye licence to run Crucible?
- Support Policies
  - Bug Fixing Policy
  - How to Report a Security Issue
  - New Features Policy
  - Patch Policy
  - Security Advisory Publishing Policy
  - Security Patch Policy
  - Severity Levels for Security Issues
- Tips of the Trade

- Fix 'Out of Memory' errors by increasing available memory

Most setup issues are likely to be related to the FishEye component of Crucible. Refer to the FishEye documentation:

- FishEye documentation
- FishEye FAQs

- Top Evaluator Questions
  - Can Crucible add support for new repositories?
  - Can I purchase Crucible on it's own?
  - Can I trial Crucible without FishEye?
  - How can I do reviews from the file system?
  - How does Crucible help enforce compliance and auditability?
  - How do I convince my team of the benefits of code review?
  - How do I do pre-commit reviews?
  - How do I raise defects in JIRA?
  - How do I review patch diffs?
  - What user permissions and review security is available?

Do you still have a question, or need help with Crucible? Please create a support request.

Troubleshooting

Crucible Troubleshooting

Crucible Troubleshooting

The most common cause of FishEye/Crucible issues is an incorrect symbolic setup (trunk/branch/tag) for Subversion repositories. If you are using Subversion and your initial index is taking forever, double-check that your symbolic setup matches your repository.

FishEye runs with the default Java heap of 64 megabytes. This is sometimes problematic for FishEye, especially for Subversion repositories during the initial scan. You can give FishEye's JVM more memory by setting the FISHEYE_OPTS environment variable.

Starting Crucible with the command line options `--debug --debug-perf` will print a lot of information to Crucible's logs. This can give you an insight into what is happening and possibly where you are stuck. Attach these logs along with your `config.xml` to an Atlassian support ticket, to speed up your support request.

Crucible freezes unexpectedly

Issue Symptoms

If your Crucible 2.0 or 2.0.1 instance freezes unexpectedly, this could be caused by a known issue with Crucible and MySQL database technology.

This issue manifests itself in some Crucible pages returning a server timeout error. To identify the issue, check the Crucible error log. For this issue, the following output will appear in the error log:

```
2009-07-15 15:34:45,555 ERROR [btpool0-519]
fisheye.app HibernateUtil-commitTransaction -
Commit fail msg-0:Could not execute JDBC
batch update
2009-07-15 15:34:45,556 ERROR [btpool0-519]
fisheye.app HibernateUtil-commitTransaction -
Commit fail msg-1:Lock wait timeout exceeded;
try restarting transaction
2009-07-15 15:34:45,557 ERROR [btpool0-519]
fisheye.app HibernateUtil-commitTransaction -
Commit failed rolling back.
...
...
Caused by: java.sql.BatchUpdateException:
Lock wait timeout exceeded; try restarting
transaction
 at
com.mysql.jdbc.ServerPreparedStatement.executeE
at
com.mchange.v2.c3p0.impl.NewProxyPreparedStatem
at
org.hibernate.jdbc.BatchingBatcher.doExecuteBat
at
org.hibernate.jdbc.AbstractBatcher.executeBatch
... 163 more
```

The Crucible error log can be found under `FISHEYE_INST/var/log/fisheye-error.log.YYYY-MM-DD` .

See the JIRA issue for more information.

Workaround

Until the issue is solved, the suggested course of action is to restart your Crucible instance. This will return Crucible to normal operation.

The Crucible development team is actively working on a solution and this be part of an upcoming point release of Crucible.

Requesting Support

If you require assistance in resolving the problem, please raise a support request under the Crucible project.

JIRA Integration Issues

Users are mapped to their own accounts when using Trusted Applications.

If you (or the general account used for JIRA access, if not using Trusted Applications) do not have the permissions to carry out the JIRA actions linked from Crucible, an error will occur. Depending on the error returned from JIRA, Crucible may not display the error correctly or display it at all, simply reporting that "An error has occurred". To investigate what the error was, you can access the Crucible debug log, named `fisheye-debug.log.YYYY-MM-DD` under the `dist.inst/var/log` folder of your Crucible installation. In the debug log, look for the date and time when your error took place. Here, you will be able to follow the links and see what error the JIRA instance was producing by clicking through to JIRA.

Problems with very long comments and MySQL migration

Issue Symptoms

There is a known issue with Crucible 2.0.x and very long comments when migrating your database to MySQL. In some circumstances, this might result in truncation of very long comments, causing data loss.

Depending on your MySQL configuration, you may see an error message like this while migrating to MySQL, causing the migration to fail:

```
2009-07-16 16:56:12,390 ERROR [ThreadPool1]
fisheye.app
com.cenqua.crucible.actions.admin.database.DBEc
-
Database migration failed:
java.sql.BatchUpdateException: Data
truncation: Data too long for column
'cru_message' at row 1
java.sql.BatchUpdateException: Data
truncation: Data too long for column
'cru_message' at row 1
```

ⓘ You may not see the message if you are running MySQL with default settings.

For more information, see the JIRA issue.

Workaround

If your data contains very long comments or review descriptions (longer than 21,845 multibyte unicode characters), consider avoiding use of MySQL until this issue is resolved fully. Alternatively, use PostgreSQL or the default (built-in) HSQLDB database.

The Crucible developers are actively working on a solution to this problem and it will be addressed in an upcoming Crucible point release.

Requesting Support

If you require assistance in resolving the problem, please raise a support request under the Crucible project.

General FAQs

Crucible General FAQs

- Can Crucible be run as a Windows Service? — To run Crucible as a service you can either use SRVANY and INSTSRV to run `java.exe` or create a Java Service Wrapper. A mechanism to run Crucible as a service will be incorporated at a later stage. In the meantime, example wrapper files written by users can be found here.
- Can I deploy Crucible or FishEye as a WAR? — Unfortunately FishEye and Crucible cannot be deployed as a WAR.
- How to Automate Daily Crucible Backups — Configuring Crucible backups is easy.

Can Crucible be run as a Windows Service?

To run Crucible as a service you can either use SRVANY and INSTSRV to run `java.exe` or create a Java Service Wrapper. A mechanism to run Crucible as a service will be incorporated at a later stage. In the meantime, example wrapper files written by users can be found here.

To install on Windows:

1. Unzip the wrapper zip file into your `FISHEYE_HOME` directory (Note, the end structure should be `FISHEYE_HOME/wrapper`, `FISHEYE_HOME/wrapper/bin`, etc and NOT `FISHEYE_HOME/wrapper/wrapper`, `FISHEYE_HOME/wrapper/wrapper/bin`. The location of the wrapper directory is important).
2. Run `Fisheye-Install-NTService.bat`, found in `FISHEYE_HOME/wrapper/bin`.
3. Start the Fisheye service under the Windows Control Panel.
4. Set your FISHEYE_INST within your `FISHEYE_HOME/wrapper/conf/wrapper.conf` as per the instructions below:

Please note, that if you do run as a service, then any Environment Variables that you want to set, need to be set in your `FISHEYE_HOME/wrapper/conf/wrapper.conf` file.

If there are other java parameters you wish to add, then you will need to add them under the additional parameters, e.g.

```
# JDK 1.5 Additional Parameters for jmx
wrapper.java.additional.4=-Dcom.sun.management.
Service Wrapper Control"
```

For example if you wish to add a FISHEYE_INST environment variable or add the java parameter "MaxPermSize", or the -Xrs options (should be used if running FishEye as a service under Windows, to prevent the JVM closing when an interactive user logs out) then it would be something like:

```
wrapper.java.additional.11=-Dfisheye.inst="c:/p
```

Your memory settings can also be found in this file:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=32

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=256
```

Increase these values if you have a large repository or expect to use more memory (init of 256, and a max of 1024 would be reasonable).

Can I deploy Crucible or FishEye as a WAR?

Unfortunately FishEye and Crucible cannot be deployed as a WAR. FishEye has some special needs for performance reasons that are not easily supported on third-party containers. Whilst this is an often requested feature, there are no immediate plans to provide a WAR version of FishEye or FishEye+Crucible. However the upcoming separate edition of Crucible (i.e. without FishEye) may at some stage be available as a WAR.

How to Automate Daily Crucible Backups

Configuring Crucible backups is easy. To set daily Crucible backups, open the administration page, click the '**Backup**' link under '**System**' on the left navigation bar, and simply follow the instructions set out on the Backing Up and Restoring Crucible Data page.

Licensing FAQ

Crucible Licensing FAQ

- What happens if I decide to stop using FishEye with Crucible? — If you are using FishEye and Crucible together and decide to start using Crucible only, you will lose certain functionality that is unique to FishEye.
- Do I need a FishEye licence to run Crucible? — FishEye and Crucible are separate products. They can be run separately, and they can also be run together.

What happens if I decide to stop using FishEye with Crucible?

If you are using FishEye and Crucible together and decide to start using Crucible only, you will lose certain functionality that is unique to FishEye. You will also need to make some configuration changes, as detailed below:

Considerations Before Removing FishEye:

1. CVS repositories are not supported in Crucible, only in FishEye.
2. Any repositories currently defined using FishEye's powerful scanning mechanism will will not be visible in Crucible after removing FishEye. You will need to reconnect these repositories using the Crucible Light SCM.
3. The Crucible Light SCM does not offer the same high performance experience that FishEye allows when accessing your repository. The Light SCM does not require time to scan the repository, but is slower to retrieve information from your repositories (as it always does it on demand rather than using caches).
4. When using Iterative Reviews in Crucible, you will not be prompted when a new version of a file is available (this is a feature that requires FishEye).
5. You will lose the ability to browse and search the repository, see the detailed history and charts showing commits and LOC information from your repository data that FishEye provides with its in-depth scanning functionality.
6. You will no longer be able to see commit activity from your repository in Activity Streams.
7. You will lose the ability to see your content roots and repositories associated with projects.
8. You will no longer be able to see repository lists and browse repositories using the '**Source**' tab.

Do I need a FishEye licence to run Crucible?

FishEye and Crucible are separate products. They can be run separately, and they can also be run together.

You can run Crucible standalone, but you will not have access to some of FishEye's advanced features, such as pre-caching repository content (for maximum performance), advanced searching and FishEye's activity graphs.

Crucible stand-alone can be used with Subversion, CVS, Perforce, Git and Clearcase via SCM plugins. In this situation, access to the repositories is strictly on-demand.

For the best combination of features and performance, run both Crucible and FishEye.

Support Policies

This page is an index for the Atlassian Support Policies that apply to this product:

- Bug Fixing Policy
- How to Report a Security Issue
- New Features Policy
- Patch Policy

- Security Advisory Publishing Policy
- Security Patch Policy
- Severity Levels for Security Issues

Bug Fixing Policy

Summary

- Atlassian Support will help with workarounds and bug reporting.
- Critical bugs will generally be fixed in the next maintenance release.
- Non critical bugs will be scheduled according to a variety of considerations.

Raising a Bug Report

Atlassian Support is eager and happy to help verify bugs — we take pride in it! Please open a support request in our support system providing as much information as possible about how to replicate the problem you are experiencing. We will replicate the bug to verify, then lodge the report for you. We'll also try to construct workarounds if they're possible.

Customers and plugin developers are also welcome to open bug reports on our issue tracking systems directly. Use http://jira.atlassian.com for the stand-alone products and http://studio.atlassian.com for JIRA Studio.

When raising a new bug, you should rate the priority of a bug according to our JIRA usage guidelines. Customers should watch a filed bug in order to receive e-mail notification when a "Fix Version" is scheduled for release.

How Atlassian Approaches Bug Fixing

Maintenance (bug fix) releases come out more frequently than major releases and attempt to target the most critical bugs affecting our customers. The notation for a maintenance release is the final number in the version (ie the 1 in 3.0.1).

If a bug is critical (production application down or major malfunction causing business revenue loss or high numbers of staff unable to perform their normal functions) then it will be fixed in the next maintenance release provided that:

- The fix is technically feasible (i.e. it doesn't require a major architectural change).
- It does not impact the quality or integrity of a product.

For non-critical bugs, the developer assigned to fixing bugs prioritises the non-critical bug according to these factors:

- How many of our supported configurations are affected by the problem.
- Whether there is an effective workaround or patch.
- How difficult the issue is to fix.
- Whether many bugs in one area can be fixed at one time.

The developers responsible for bug fixing also monitor comments on existing bugs and new bugs submitted in JIRA, so you can provide feedback in this way. We give high priority consideration to security issues.

When considering the priority of a non-critical bug we try to determine a 'value' score for a bug which takes into account the severity of the bug from the customer's perspective, how prevalent the bug is and whether roadmap features may render the bug obsolete. We combine this with a complexity score (i.e. how difficult the bug is). These two dimensions are used when developers self serve from the bug pile.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

How to Report a Security Issue

Finding and Reporting a Security Vulnerability

If you find a security bug in the product, please open an issue on http://jira.atlassian.com in the relevant project.

- Set the priority of the bug to 'Blocker'.
- Provide as much information on reproducing the bug as possible.
- Set the security level of the bug to 'Developer and Reporters only'.

All communication about the vulnerability should be performed through JIRA, so that Atlassian can keep track of the issue and get a patch out as soon as possible.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

New Features Policy

Summary

- We encourage and display customer comments and votes openly in our issue tracking systems, http://jira.atlassian.com and http://studio.atlassian.com.
- We do not publish roadmaps.
- Product Managers review our most popular voted issues on a regular basis.
- We schedule features based on a variety of factors.
- Our Atlassian Bug Fixing Policy is distinct from our Feature Request process.
- Atlassian provides consistent updates on the top 20 feature/improvement requests (in our issue tracker systems).

How to Track what Features are Being Implemented

When a new feature or improvement is scheduled, the 'fix-for' version will be indicated in the JIRA issue. This happens for the upcoming release only. We maintain roadmaps for more distant releases internally, but because these roadmaps are often pre-empted by changing customer demands, we do not publish them.

How Atlassian Chooses What to Implement

In every major release we *aim* to implement highly requested features, but it is not the only determining factor. Other factors include:

- **Direct feedback** from face to face meetings with customers, and through our support and sales channels.
- **Availability of staff** to implement features.
- **Impact** of the proposed changes on the application and its underlying architecture.
- How **well defined** the requested feature is (some issues gain in popularity rapidly, allowing little time to plan their implementation).
- Our long-term **strategic vision** for the product.

How to Contribute to Feature Development

Influencing Atlassian's release cycle

We encourage our customers to vote on feature requests in JIRA. The current tally of votes is available online in our issue tracking systems, http://jira.atlassian.com and http://studio.atlassian.com. Find out if your improvement request already exists. If it does, please vote for it. If you do not find it, create a new feature or improvement request online.

Extending Atlassian Products
Atlassian products have powerful and flexible extension APIs. If you would like to see a particular feature implemented, it may be possible to develop the feature as a plugin. Documentation regarding the plugin APIs is available. Advice on extending either product may be available on the user mailing-lists, or at our community forums.

If you require significant customisations, you may wish to get in touch with our partners. They specialise in extending Atlassian products and can do this work for you. If you are interested, please contact us.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Patch Policy

Patch Policy

Atlassian will only provide software patches in extremely unusual circumstances. If a problem has been fixed in a newer release of the product, Atlassian will request that you upgrade your instance to fix the issue. If it is deemed necessary to provide a patch, a patch will be provided for the current release (e.g. JIRA 4.0) and the last maintenance release of the last major version (e.g. JIRA 3.13.5) only.

Patches are issued under the following conditions:

- The bug is critical (production application down or major malfunction causing business revenue loss or high numbers of staff unable to perform their normal functions).
- A patch is technically feasible (ie it doesn't require a major architectural change)
  OR
- The issue is a security issue, and falls under our Security Policy .

Atlassian does not provide patches for non-critical bugs.

Provided that a patch does not impact the quality or integrity of a product, Atlassian will ensure that patches supplied to customers are added to the next maintenance release. Customers should watch a filed bug in order to receive e-mail notification when a "Fix Version" is scheduled for release.

Patches are generally attached to the relevant http://jira.atlassian.com issue.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Security Advisory Publishing Policy

Publication of Security Advisories

When a security issue in an Atlassian product is discovered and resolved, Atlassian will inform customers through the following mechanisms:

- A security advisory will be posted in the documentation.
- A copy of the advisory will be sent to the product mailing-lists. These lists are mirrored on our forums.
- If the person who reported the issue wants to publish an advisory through some other agency (for example, CERT), Atlassian will assist in the production of that advisory, and link to it from our own.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Security Patch Policy

Our Security Patch Policy

When a security issue is discovered, Atlassian will endeavour to:

- Issue a new, fixed version as soon as possible,
- Issue a patch to the current release (e.g. JIRA 4.0) and the latest maintenance release for the last major version of a product (e.g. JIRA 3.13.5),
- Issue patches for older versions if feasible.

Patches will generally be attached to the relevant JIRA issue.

Visit our general Atlassian Patch Policy as well.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Severity Levels for Security Issues

Severity Levels

Atlassian security advisories include a severity level, rating the vulnerability as one of the following:

- Critical
- High
- Moderate
- Low

Below is a summary of the factors which we use to decide on the severity level, and the implications for your installation.

Severity Level: Critical

We classify a vulnerability as critical if most or all of the following are true:

- Exploitation of the vulnerability results in root-level compromise of servers or infrastructure devices.
- The information required in order to exploit the vulnerability, such as example code, is widely available to attackers.
- Exploitation is usually straightforward, in the sense that the attacker does not need any special authentication credentials or knowledge about individual victims, and does not need to persuade a target user, for example via social engineering, into performing any special functions.

Severity Level: High

We give a high severity level to those vulnerabilities which have the potential to become critical, but have one or more mitigating factors that make exploitation less attractive to attackers.

For example, given a vulnerability which has many characteristics of the critical severity level, we would give it a level of high if any of the following are true:

- The vulnerability is difficult to exploit.
- Exploitation does not result in elevated privileges.
- The pool of potential victims is very small.

Note: If the mitigating factor arises from a lack of technical details, the severity level would be elevated to critical if those details later became available. If your installation is mission-critical, you may want to treat this as a critical vulnerability.

Severity Level: Moderate

We give a moderate severity level to those vulnerabilities where the scales are slightly tipped in favour of the potential victim.

The following vulnerabilities are typically rated moderate:

- Denial of service vulnerabilities, since they do not result in compromise of a target.
- Exploits that require an attacker to reside on the same local network as the victim.
- Vulnerabilities that affect only nonstandard configurations or obscure applications.
- Vulnerabilities that require the attacker to manipulate individual victims via social engineering tactics.
- Vulnerabilities where exploitation provides only very limited access.

Severity Level: Low

We give a low severity level to those vulnerabilities which by themselves have typically very little impact on an organisation's infrastructure.

Exploitation of such vulnerabilities usually requires local or physical system access. Exploitation may result in client-side privacy or denial of service issues and leakage of information about organisational structure, system configuration and versions, or network topology.

> **Original ranking compiled by the SANS Institute**
> Our vulnerability ranking is based on a scale originally published by the SANS Institute.

Further reading

See How to Get Legendary Support from Atlassian for more support-related information.

Tips of the Trade

Below are some links to external blog posts and articles containing technical tips and instructions on setting up and using Crucible. This page presents an opportunity for customers and community authors to share information and experiences.

The references here are specific to Crucible and are technical 'how to' guides written by bloggers who use Crucible. For more general information on code review solutions, best practices and business cases, please refer to the Atlassian website.

⚠ **Please be aware that these are external blogs and articles.**
Most of the links point to external sites, and some of the information is relevant to a specific release of Crucible. Atlassian provides these links because the information is useful and relevant at the time it was written. Please check carefully whether the information is still relevant when you read it, and whether it is relevant to your version of Crucible. Unless explicitly stated, Atlassian does not offer support for third-party extensions or plugins. The information in the linked blog posts has not been tested or reviewed by Atlassian. We recommend that you test all solutions on a **test** server before trying it on your production site.

On this page:

- Running Atlassian Crucible (or Fisheye) on Linux Startup
- Reviewing wiki documentation via Crucible

Administration

Running Atlassian Crucible (or Fisheye) on Linux Startup

- By: Jarrod, on blog 'The stupid people made me do it!'
- About: How to get Crucible to start automatically when the server boots, via an initialisation script that runs on RedHat-like Linux boxes
- Date and Crucible version: 9 March 2009; Crucible 1.6
- Related documentation: Auto Start on Windows via a Windows Service

Reviewing Confluence Pages

Reviewing wiki documentation via Crucible

- By: Sarah Maddox, on blog 'ffeathers'
- About: Setting up Crucible and Confluence so that you can use Crucible to review Confluence wiki pages
- Date and software versions: 17 January 2009; Crucible 1.6 and Confluence 2.10
- Related documentation:
  - Setting Up Reviewing of Confluence Pages in Crucible
  - Confluence SCM Plugin

✓ **Have you written a technical tip for Crucible?**
Add a comment to this page, linking to your blog post or article. We will include it if the content fits the requirements of this page.

✓ **Feedback?**
Your first port of call should be the author of the linked blog post. If you want to let us know how useful (or otherwise) a linked post is, please add a comment to this page.

Other Sources of Information

Crucible documentation
Atlassian website
Atlassian forums
Atlassian Blog
Crucible plugins

Glossary

Code review takes many forms and has many names. For conciseness, Crucible has adopted the following terms and meanings:

approve

author

code review

comment

creator

defect

moderator

participant

permission

permission scheme

project

review duration

reviewer

role

state

statement of objective

user

approve

Issuing a review to the reviewers is known as *approving* the review.

author

The *author* is the person primarily responsible for acting on the outcomes of the review. In the vast majority of cases the author will be the person who made the code change under review.

Note: to map your repository username to your FishEye/Crucible username, see Changing your User Profile.

code review

Without prejudice to 'code inspection', 'peer review' or a myriad of other terms, Crucible uses the phrase *code review* for simplicity.

See About Crucible and Background Reading.

comment

A *comment* is a short textual note that is linked to a review, revision/diff, source line, or to another comment.

See Adding Comments.

creator

The *creator* is the person who creates the review. In most cases this person will also act as moderator.

defect

A *defect* is a comment flagged as something that requires addressing and includes optional defect classifications.

See Flagging Defects and Customising the Defect Classifications.

moderator

The *moderator* is the person responsible for creating the review, approving the review, determining when reviewing is finished, summarising the outcomes and closing the review. By default, the moderator is the creator.

participant

Crucible uses the terms creator, author, moderator, and reviewer to describe the *roles* of review participants.

permission

A *permission* is the ability to perform a particular action in Crucible, e.g. 'Create Review'.
Permissions are assigned to particular users, groups or review roles by means of permission schemes.

The following permissions are available:

Permission	Description	Default Assignees
'**Edit**'	Ability to edit a review's details and change the set of revisions being reviewed.	'Creator' 'Moderator'

'**View**'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users 'Creator' 'Author' 'Reviewer' 'Moderator'
'**Abandon**'	Ability to abandon (i.e. cancel) a review.	'Moderator' 'Creator'
'**Re-Open**'	Ability to re-open a closed or abandoned review.	'Creator' 'Moderator'
'**Uncomplete**'	Ability of a reviewer to change their individual review status from 'Complete' to 'Uncomplete'.	'Reviewer'
'**Reject**'	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	'Moderator'
'**Complete**'	Ability of a reviewer to change their individual review status to 'Complete'.	'Reviewer'
'**Comment**'	Ability to add or remove a comment to or from a review.	'Creator' 'Author' 'Reviewer' 'Moderator'
'**Approve**'	Ability to approve a review (i.e. issue it to the reviewers).	'Moderator'
'**Submit**'	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	'Creator' 'Author'
'**Close**'	Ability to close a review once it has been summarised.	'Moderator'
'**Delete**'	Ability to delete a review.	'Creator' 'Moderator'
'**Summarise**'	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	'Moderator'
'**Create**'	Ability to create a review.	All logged-in users
'**Recover**'	Ability to resurrect an abandoned (i.e. cancelled) review.	'Creator' 'Moderator'

permission scheme

A *permission scheme* assigns particular permissions to any or all of the following:

- Particular Users.

- Particular Groups.
- All logged-in users.
- Anonymous Users
- People in particular Review Roles, such as:
  - 'Author';
  - 'Reviewer';
  - 'Creator';
  - 'Moderator'.

The scheme's permissions will apply to all reviews belonging to the project(s) with which the scheme is associated.

You can create as many permission schemes as you wish. Each permission scheme can be associated with many projects or just one project, allowing you to tailor appropriate permissions for individual projects as required.

See Creating a Permission Scheme.

project

A Crucible *project* is a collection of reviews, typically reviews that all relate to the same application. In addition to providing a logical way of grouping reviews together, a project allows you to

- define default moderators, authors and reviewers for the reviews in that project.
- define which people are eligible to be reviewers for the reviews in that project.
- use permission schemes to restrict who can perform particular actions (e.g. 'Create Review') in that project.

Every Crucible review belongs to a project. Each project has a *name* (e.g. **ACME Development**) and a *key* (e.g. **ACME**). The project key becomes the first part of that project's *review keys*, e.g. **ACME-101**, **ACME-102**, etc:

By default, Crucible contains one project. This default project has the key '**CR**' and the name '**Default Project**'.

See Creating a Project.

review duration

The *review duration* is the period of time for which a review will run.

See Setting the Default Review Duration for a Project.

reviewer

A *reviewer* is a person assigned to review the change. Reviewers can make comments and indicate when they have completed their review. The moderator and author are implicitly considered reviewers.

role

See participant.

state

A Crucible review moves through the following states in the following sequence:

Draft	See Creating a Review.
Require Approval	Relevant only when the moderator is not the creator. See Issuing a Review.
Under Review	See Issuing a Review and Reviewing the Code.
Summarize	See Summarising and Closing the Review.
Closed	See Summarising and Closing the Review.

ⓘ Reviews can be re-opened, i.e. moved from **Summarize** or **Closed** back to **Under Review**.

A review may also be in the following states:

Abandoned	This happens when a review is deleted.
Rejected	Any reviews that a moderator has rejected.

statement of objective

A *statement of objective* is an optional text description of the review and any specific areas the reviewers should focus on.

user

A *user* is a person using Crucible.

About Crucible

Crucible is a powerful addition to FishEye, making it easy to review code changes, make comments, and record outcomes in an efficient, distributed, and process-neutral way.

Introduction

Crucible is a tool that facilitates code review. It can be as valuable to organisations that already have a formal inspection process as it is to teams that don't review at all.

Regular peer review is a proven process with demonstrable return on investment (ROI). The benefits vary from team to team but commonly include:

- Identifying bugs and defects early.
- Sharing expertise and encouraging knowledge transfer.
- Improving system-wide knowledge.
- Encouraging adherence to internal standards and style conventions.
- Identifying individual strengths and weaknesses.

One of the less apparent, but nonetheless important, benefits that comes from a transparent code review process is that quality improves simply from the knowledge that code may be critically reviewed. Developers take more care with style, readability, comments, and commit-messages because their peers are going to see them.

Despite these and many other clear benefits, code review is often seen as 'impractical on time sensitive projects', 'only valuable in large teams working on mission critical applications', or at worst 'a total waste of time foisted on developers by management'. Formal code review can *feel* like an expensive use of time, because the review process can:

- Be burdened by excessive paperwork and other administration.
- Interrupt your current task and make you less productive.
- Include meetings where participants fail to prepare, so that the meeting becomes a walkthrough rather than a critical review.
- Become an ego battle or point-scoring exercise dominated by a vocal minority.

These issues do not affect the immense potential value of code review. They are simply problems with some review processes.

Crucible's mission is to streamline the *process* aspects so development teams can access the benefits. Crucible achieves this by:

- Making reviews asynchronous.
- Bringing reviewing to your desk (wherever that might be).
- Eliminating most of the administration.
- Limiting the ability for individuals to dominate the dialogue.
- Providing an archival record of reviews.

Crucible increases the quality, quantity, and frequency of code reviews thereby reducing bugs, helping knowledge sharing and fundamentally improving system quality.

Starting Points

Visit the Crucible Feature Tour to understand how Crucible can benefit you.

You can run Crucible with a FishEye-compatible source code repository set up, such as CVS, Subversion, or Perforce. For more information, please read the FishEye documentation.

Read the Installation Guide to get started quickly.

For Crucible troubleshooting, see the FAQ.

Background Reading

The following resources are recommended for background reading on peer code reviews:

- White paper on effective code review by Karl Wiegers.
- Book, Peer Reviews in Software: A Practical Guide by Karl Weigers.
- Software Engineering Institute web page: Software Inspections.
- NASA Software Assurance Technology Center web page: Software Formal Inspections.