1.2.5.1.3 Creating a Review within Crucible .................................................. 60
1.2.5.1.4 Creating a Review from a URL .................................................. 61
1.2.5.1.5 Creating a Patch Review .............................................................. 62
1.2.5.1.6 Selecting the Files for the Review .............................................. 68
1.2.5.2 Adding Reviewers ................................................................. 75
1.2.5.2.1 Removing Reviewers From An Active Review .......................... 78
1.2.5.3 Issuing a Review ................................................................. 79
1.2.5.4 Performing the Review .......................................................... 79
1.2.5.4.1 Adding Comments ............................................................ 81
1.2.5.4.2 Flagging Defects ............................................................... 83
1.2.5.4.3 Completing your Review ....................................................... 84
1.2.5.4.4 Sending all of a Review's Comments via Email ...................... 85
1.2.5.4.5 Using JIRA Integration in Crucible Reviews ......................... 87
1.2.5.5 Summarising and Closing the Review ........................................ 93
1.2.5.6 Deleting an Abandoned Review ............................................... 96
1.2.5.7 Moving a Review to Another Project ....................................... 96
1.2.6 Using the Dashboard ............................................................. 97
1.2.6.1 Using the Project Dashboard ................................................... 99
1.2.7 Defining your Workflow ......................................................... 100
1.2.8 Using Favourites ................................................................. 105
1.2.9 Using Keyboard Shortcuts in Crucible ........................................ 110
1.2.10 Using RSS Feeds in Crucible .................................................. 111
1.2.11 Using Wiki Markup in Crucible .............................................. 111
1.3 Crucible Administrator's Guide .................................................. 117
1.3.1 Backing Up and Restoring Crucible Data ................................... 117
1.3.2 Creating a Permission Scheme ................................................. 126
1.3.2.1 Associating a Permission Scheme with a Project .................... 130
1.3.3 Creating a Project ................................................................. 132
1.3.3.1 Setting Crucible to Store all Revisions ................................. 134
1.3.4 Crucible and FishEye .............................................................. 135
1.3.5 Customising Email Notifications ............................................. 136
1.3.5.1 Freemarker Data Model for Email Templates ...................... 137
1.3.6 Customising the Defect Classifications .................................... 138
1.3.7 Customising the Welcome Message ......................................... 140
1.3.8 Deleting a Project ................................................................. 142
1.3.9 Editing a Project ................................................................. 142
1.3.9.1 Setting Default Review Objectives ................................... 145
1.3.9.2 Setting the Default Review Duration for a Project ............. 145
1.3.10 JIRA Integration in Crucible ................................................. 145
1.3.11 Migrating to an External Database ....................................... 151
1.3.12 Trusted Applications .......................................................... 156
1.4 Crucible Release Notes .......................................................... 159
1.4.1 Crucible 2.1 Release Notes ..................................................... 160
1.4.1.1 Crucible 2.1 Changelog ......................................................... 168
1.4.2 Crucible 2.0 Release Notes ..................................................... 169
1.4.2.1 Crucible 2.0 Changelog .......................................................... 176
1.4.2.2 Crucible 2.0 Upgrade Notes ................................................... 194
1.4.3 Crucible 2.0 Beta Release Notes ............................................. 194
1.4.3.1 Crucible 2.0 Beta Upgrade Notes ......................................... 200
1.4.3.2 Crucible 2.0 Beta Reviewer's Guide ................................... 202
1.4.3.3 JIRA Integration in Crucible 2.0 Beta ................................ 203
1.4.4 Crucible 1.6 Release Notes ..................................................... 208
1.4.4.1 Crucible 1.6 Changelog .......................................................... 217
1.4.4.1.1 Crucible 1.6.3 Upgrade Guide ......................................... 224
1.4.5 Crucible 1.5 Release Notes ..................................................... 224
1.4.5.1 Crucible 1.5 Changelog .......................................................... 229
1.4.6 Crucible 1.2 Release Notes ..................................................... 231
1.4.6.1 Crucible 1.2 Changelog .......................................................... 236
1.4.6.2 Crucible 1.2 Upgrade Guide ................................................... 239
1.4.7 Crucible 1.1 Release Notes ..................................................... 239
1.4.7.1 Crucible 1.1 Changelog .......................................................... 239
1.4.7.2 Crucible 1.1 Upgrade Guide ................................................... 240
1.4.8 Crucible Release Summary ..................................................... 240
1.5 Crucible FAQ ................................................................. 241
1.5.1 Troubleshooting ................................................................. 242
1.5.1.1 Crucible freezes unexpectedly .......................................... 243
1.5.1.2 JIRA Integration Issues ......................................................... 245
1.5.1.3 Problems with very long comments and MySQL migration ...... 245
1.5.2 General FAQs ................................................................. 246
1.5.2.1 Can Crucible be run as a Windows Service? ...................... 246
1.5.2.2 Can I deploy Crucible or FishEye as a WAR? ..................... 247
1.5.2.3 How to Automate Daily Crucible Backups ...................... 247
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5.3 Licensing FAQ</td>
<td>247</td>
</tr>
<tr>
<td>1.5.3.1 What happens if I decide to stop using FishEye with Crucible?</td>
<td>248</td>
</tr>
<tr>
<td>1.5.3.2 Do I need a FishEye licence to run Crucible?</td>
<td>248</td>
</tr>
<tr>
<td>1.5.4 Support Policies</td>
<td>248</td>
</tr>
<tr>
<td>1.5.4.1 Bug Fixing Policy</td>
<td>249</td>
</tr>
<tr>
<td>1.5.4.2 How to Report a Security Issue</td>
<td>250</td>
</tr>
<tr>
<td>1.5.4.3 New Features Policy</td>
<td>250</td>
</tr>
<tr>
<td>1.5.4.4 Patch Policy</td>
<td>251</td>
</tr>
<tr>
<td>1.5.4.5 Security Advisory Publishing Policy</td>
<td>251</td>
</tr>
<tr>
<td>1.5.4.6 Security Patch Policy</td>
<td>252</td>
</tr>
<tr>
<td>1.5.4.7 Severity Levels for Security Issues</td>
<td>252</td>
</tr>
<tr>
<td>1.5.5 Tips of the Trade</td>
<td>253</td>
</tr>
<tr>
<td>1.6 Glossary</td>
<td>255</td>
</tr>
<tr>
<td>1.6.1 approve</td>
<td>255</td>
</tr>
<tr>
<td>1.6.2 author</td>
<td>255</td>
</tr>
<tr>
<td>1.6.3 code review</td>
<td>256</td>
</tr>
<tr>
<td>1.6.4 comment</td>
<td>256</td>
</tr>
<tr>
<td>1.6.5 creator</td>
<td>256</td>
</tr>
<tr>
<td>1.6.6 defect</td>
<td>256</td>
</tr>
<tr>
<td>1.6.7 moderator</td>
<td>256</td>
</tr>
<tr>
<td>1.6.8 participant</td>
<td>256</td>
</tr>
<tr>
<td>1.6.9 permission</td>
<td>256</td>
</tr>
<tr>
<td>1.6.10 permission scheme</td>
<td>257</td>
</tr>
<tr>
<td>1.6.11 project</td>
<td>258</td>
</tr>
<tr>
<td>1.6.12 review duration</td>
<td>258</td>
</tr>
<tr>
<td>1.6.13 reviewer</td>
<td>258</td>
</tr>
<tr>
<td>1.6.14 role</td>
<td>258</td>
</tr>
<tr>
<td>1.6.15 state</td>
<td>259</td>
</tr>
<tr>
<td>1.6.16 statement of objective</td>
<td>259</td>
</tr>
<tr>
<td>1.6.17 user</td>
<td>259</td>
</tr>
<tr>
<td>1.7 About Crucible</td>
<td>259</td>
</tr>
<tr>
<td>1.7.1 Background Reading</td>
<td>260</td>
</tr>
</tbody>
</table>
# 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.
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 the following browsers:

- Firefox 3
- Internet Explorer 7
- Safari 4

ℹ️ 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
- Setting up a Repository for use with Stand-alone Crucible
- Setting up a Repository for use with FishEye and Crucible
- Next: Configuration

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
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Repository Server</td>
<td>Enter the base URL and port for the repository, for example: example.com:666.</td>
</tr>
<tr>
<td>Repository Path</td>
<td>Add the path to your Perforce repository. For example: //depot/code/example/main.</td>
</tr>
<tr>
<td>Perforce Username</td>
<td>Enter the username of the Perforce account that Crucible will use. (optional)</td>
</tr>
<tr>
<td>Perforce Password</td>
<td>Enter the password of the Perforce account that Crucible will use. (optional)</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Repository Root</td>
<td>Enter the repository root URL for the repository. If you are not sure what the repository root is, please see the instructions below under &quot;Finding your Repository Root&quot;.</td>
</tr>
<tr>
<td>Repository Path</td>
<td>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 '<a href="http://svn.example.com/svn5/">http://svn.example.com/svn5/</a>', you would enter 'svn5' into this field.</td>
</tr>
<tr>
<td>SVN Username</td>
<td>Enter the username of the Subversion account that Crucible will use. Note that this account should only have read-only access to the repository.</td>
</tr>
<tr>
<td>SVN Password</td>
<td>Enter the password of the Subversion account that Crucible will use.</td>
</tr>
</tbody>
</table>

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.

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>URL</td>
<td>Enter the URL of your Confluence instance.</td>
</tr>
<tr>
<td>Space Key</td>
<td>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.</td>
</tr>
</tbody>
</table>

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](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Base Path</td>
<td>Choose the lowest level of directory that Crucible will access.</td>
</tr>
</tbody>
</table>

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/](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.

ℹ️ 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:

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/](http://HOSTNAME:8060/) in a browser.

Or you can go directly into the Crucible homepage at [http://HOSTNAME:8060/crucible](http://HOSTNAME:8060/crucible)

**Stopping Crucible**

To stop the Crucible server:

- **On Windows:**
  
  ```sh
  C:\> cd FISHEYE_HOME\bin
  "C:\FISHEYE_HOME\bin> stop.bat"
  ```

- **On Unix-based systems:**
  
  ```sh
  $ 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Base Path</td>
<td>Choose the lowest level of directory that Crucible will access.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name for the repository eg. <em>Project</em></td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Repository Path</td>
<td>The location of the local Git repository clone</td>
</tr>
</tbody>
</table>

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**

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Repository Server</td>
<td>Enter the base URL and port for the repository, for example: example.com:666.</td>
</tr>
<tr>
<td>Repository Path</td>
<td>Add the path to your Perforce repository. For example: //depot/code/example/main.</td>
</tr>
<tr>
<td>Perforce Username</td>
<td>Enter the username of the Perforce account that Crucible will use. (optional)</td>
</tr>
<tr>
<td>Perforce Password</td>
<td>Enter the password of the Perforce account that Crucible will use. (optional)</td>
</tr>
</tbody>
</table>

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 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.

⚠️ 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>Repository Root</td>
<td>Enter the repository root URL for the repository. If you are not sure what the repository root is, please see the instructions below under &quot;Finding your Repository Root&quot;.</td>
</tr>
<tr>
<td>Repository Path</td>
<td>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 '<a href="http://svn.example.com/svn5/">http://svn.example.com/svn5/</a>', you would enter 'svn5' into this field.</td>
</tr>
<tr>
<td>SVN Username</td>
<td>Enter the username of the Subversion account that Crucible will use. Note that this account should only have read-only access to the repository.</td>
</tr>
<tr>
<td>SVN Password</td>
<td>Enter the password of the Subversion account that Crucible will use.</td>
</tr>
</tbody>
</table>

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

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’.

<table>
<thead>
<tr>
<th>Field</th>
<th>What to enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Choose a unique name for the repository.</td>
</tr>
<tr>
<td>URL</td>
<td>Enter the URL of your Confluence instance.</td>
</tr>
<tr>
<td>Space Key</td>
<td>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.</td>
</tr>
</tbody>
</table>

7. Click ‘Save’. The view will return to the list of repositories.
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

1. 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.

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
- Upgrade Procedure
  - Method 1 - Without a FISHEYE_INST Directory (default)
  - Method 2 - Using a FISHEYE_INST Directory
  - Method 3 - Without a FISHEYE_INST Directory, but intending to set one up
- Initial Crucible Configuration

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.)
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.

- Getting Started with Crucible
- Using the Crucible Screens
  - Browsing Your Reviews
  - Browsing Source Files
  - Browsing Projects
    - Viewing Project Statistics
  - Viewing People's Statistics
  - Viewing Reports
  - Searching in Crucible
  - Crucible Icons
- Changing your User Profile
- Roles and Status Classifications
- Conducting a Review
  - Creating a Review
    - 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
  - Adding Reviewers
  - Removing Reviewers From An Active Review
  - Issuing a Review
  - Performing the Review
    - Adding Comments
    - Flagging Defects
    - Completing your Review
    - Sending all of a Review's Comments via Email
    - Using JIRA Integration in Crucible Reviews
  - (hidden draft) Using JIRA Integration in Crucible Reviews
- Summarising and Closing the Review
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*
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

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
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
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

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

*Screenshot: The Crucible Review screen*
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 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
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**

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

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

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.

Tip: 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

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.

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

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

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.

<table>
<thead>
<tr>
<th>Sub-Tab name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revisions</td>
<td>When viewing a file, shows the latest revisions of the file.</td>
</tr>
<tr>
<td>Files</td>
<td>When viewing a folder, shows the contents of the directory.</td>
</tr>
<tr>
<td>Activity</td>
<td>Shows recent activity on the item. There are a number of sub-options here:</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>- All Activity — The default view, showing commits, reviews and JIRA issues.</td>
</tr>
<tr>
<td></td>
<td>- Commits — Shows commits in the activity stream.</td>
</tr>
<tr>
<td></td>
<td>- Reviews — Shows review activity in the activity stream.</td>
</tr>
<tr>
<td></td>
<td>- Scroll to Changeset — Opens the changeset ID specified in the text field (press Enter to carry out the action).</td>
</tr>
<tr>
<td></td>
<td>- Filter — Applies constraints to the current activity stream.</td>
</tr>
<tr>
<td></td>
<td>- Show Revisions — If this is selected, then changeset items are automatically expanded to show modified files.</td>
</tr>
<tr>
<td></td>
<td>- Earlier Activity (Left Arrow icon) — Loads a page of earlier activity.</td>
</tr>
<tr>
<td></td>
<td>- Later Activity (Right Arrow icon) — Loads a page of later activity.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Users</th>
<th>Shows the commit history of the different users that have committed changes on the item.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports</td>
<td>Shows activity charts for the item. Various chart options can be selected in the left navigation bar.</td>
</tr>
<tr>
<td>Source</td>
<td>Shows the contents of the file.</td>
</tr>
<tr>
<td>Query</td>
<td>Allows you to run an advanced search.</td>
</tr>
</tbody>
</table>

**Screenshot: The Repositories View**

**Screenshot: The Source View Item Activity**
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.

<table>
<thead>
<tr>
<th>Sub-Tab Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **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*
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*
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**

<table>
<thead>
<tr>
<th>Stats</th>
<th>Last Week</th>
<th>All Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commits:</td>
<td>137</td>
<td>16,033</td>
</tr>
<tr>
<td>Files changed:</td>
<td>543</td>
<td>69,896</td>
</tr>
<tr>
<td>Lines changed:</td>
<td>33,543</td>
<td>504,583</td>
</tr>
</tbody>
</table>

**Project Commit Activity Chart**

This panel contains a number of charts:

<table>
<thead>
<tr>
<th>Chart</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>52 week commits volume</td>
<td>This chart shows the amount of commits, shown by week over a one year period.</td>
</tr>
<tr>
<td>Commits by day</td>
<td>This chart shows the amount of commits, graphed by day over the past week.</td>
</tr>
<tr>
<td>Commits by hour</td>
<td>This chart shows the amount of commits, graphed by hours over the past day.</td>
</tr>
<tr>
<td>Commit calendar</td>
<td>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.</td>
</tr>
</tbody>
</table>
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)

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
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*
- **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

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.
      ⚠️ 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':
      ⚠️ 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
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:

- Using the Filters Navigation Bar
  - ‘Everyone’s Reviews’ Filters
  - Custom Filters
  - Review State Filter
  - Comment Search
- Viewing Search Results
  - Review Search
  - Comment Search

Screenshot: Search Filter Options
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':

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Find reviews by searching for words within the title.</td>
</tr>
<tr>
<td>Project</td>
<td>Find reviews under a particular project.</td>
</tr>
<tr>
<td>Author</td>
<td>Find reviews moderated by a particular authors.</td>
</tr>
<tr>
<td>Moderator</td>
<td>Find reviews moderated by a particular moderators.</td>
</tr>
<tr>
<td>Creator</td>
<td>Find reviews created by a particular creator.</td>
</tr>
<tr>
<td>Reviewer</td>
<td>Find reviews that are reviewed by a particular reviewer. This will default to the user logged in.</td>
</tr>
<tr>
<td>Reviewer</td>
<td>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.</td>
</tr>
<tr>
<td>Status</td>
<td>To use all the above filters, choose 'all'. To use any of the filters, choose 'any'.</td>
</tr>
</tbody>
</table>

Review State Filter

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

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>Reviews that are still in 'Draft' state.</td>
</tr>
<tr>
<td>Pending Approval</td>
<td>Reviews that have been moved out of 'Draft' state and are now waiting for the moderator to approve.</td>
</tr>
<tr>
<td>Under Review</td>
<td>The review is now 'Under Review'.</td>
</tr>
<tr>
<td>Summarize</td>
<td>The review is now in 'Summarize' state.</td>
</tr>
<tr>
<td>Closed</td>
<td>Reviews that are now 'Closed'.</td>
</tr>
<tr>
<td>Abandoned</td>
<td>Any reviews that are no longer relevant and can be deleted.</td>
</tr>
<tr>
<td>Rejected</td>
<td>Any reviews that a moderator has rejected.</td>
</tr>
<tr>
<td>Review needs fixing</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

Screenshot: Search Comment Filter Options
**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*
Comment Search

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

Screenshot: Comment Search Results
Crucible Icons

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔊</td>
<td>View review-level comments</td>
</tr>
<tr>
<td>🔷</td>
<td>Go to the previous comment</td>
</tr>
<tr>
<td>🔷</td>
<td>Go to the next comment</td>
</tr>
<tr>
<td>⌐</td>
<td>Add a comment</td>
</tr>
<tr>
<td>🔷</td>
<td>Go to the previous file in this review</td>
</tr>
<tr>
<td>🔷</td>
<td>Go to the next file in this review</td>
</tr>
<tr>
<td>🔷</td>
<td>Expand all files</td>
</tr>
<tr>
<td>🔷</td>
<td>Collapse all files</td>
</tr>
<tr>
<td>🔷🔗</td>
<td>A file included in this review</td>
</tr>
<tr>
<td>🗂🔗</td>
<td>A directory included in this review</td>
</tr>
</tbody>
</table>

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'.

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 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

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
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

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.
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:

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

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:
The parameters are as follows:

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

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:**

- Loading a Patch Into Crucible
- Creating a Patch File From Your IDE
  - Creating a Patch File in IntelliJ IDEA 7.0
  - Creating a Patch File in Eclipse 3.3.1.1
- Creating a Basic Patch File From The Repository Command Line
  - CVS Patch Creation On The Command Line
  - Subversion Patch Creation Via The Command Line
  - Perforce Patch Creation Via The Command Line
- Creating Patches That Include All Lines of Code
  - Creating a Patch in CVS 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**

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

*Screenshot: The IDEA Create Patch dialog*

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

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:
Crucible 2.1 Documentation

\texttt{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**

\textbf{Set \texttt{P4DIFF}}

To create a patch in Perforce, you must ensure you have set \texttt{P4DIFF} to point to a GNU-compatible diff program.

Then use \texttt{p4 diff -du} to generate a patch for changed files. For example:

\begin{verbatim}
\texttt{p4 diff -du > patch.txt}
\end{verbatim}

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

\begin{verbatim}
\texttt{diff -u path_to_added_file /dev/null >> patch.txt}
\end{verbatim}

In the example above, replace \texttt{path_to_added_file} with the actual path of your added file. You can follow a similar procedure with deleted files using \texttt{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:
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:

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

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.

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
```

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
  - Left Navigation Bar Links In The Manage Files Dialog
- Selecting Changesets for Review
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

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

<table>
<thead>
<tr>
<th>Left Nav Title</th>
<th>Explanation</th>
<th>Special Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review Content</td>
<td>Shows the list of files selected for review and allows you to manage them.</td>
<td></td>
</tr>
<tr>
<td>Change Sets</td>
<td>Allows you to choose change sets from a Source Code Management (SCM) repository and add them to a review.</td>
<td>Only appears when FishEye is installed.</td>
</tr>
<tr>
<td>Files</td>
<td>Allows you to choose files from a Source Code Management (SCM) repository and add them to a review.</td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>Allows you to search a Source Code Management (SCM) repository for files or change sets and add them to a review.</td>
<td>Only appears when FishEye is installed.</td>
</tr>
<tr>
<td>Suggestions</td>
<td>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).</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Patches</td>
<td>Allows you to upload patch files to a review.</td>
<td></td>
</tr>
<tr>
<td>Uploads</td>
<td>Allows you to upload any file to a review, including binary files and files outside of a Source Code Management (SCM) repository.</td>
<td></td>
</tr>
</tbody>
</table>

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](image)

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**

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**
<table>
<thead>
<tr>
<th><strong>Repository</strong></th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Branch</strong></td>
<td>This will only show files and recent changes on that branch from the repository set above.</td>
</tr>
<tr>
<td><strong>Tag</strong></td>
<td>This will only show files and recent changes tagged.</td>
</tr>
<tr>
<td><strong>Add to Review As</strong></td>
<td>Choose the form of the review; see full details.</td>
</tr>
<tr>
<td><strong>Go to Changeset</strong></td>
<td>Allows you to jump to a particular change set by entering its title and pressing Enter.</td>
</tr>
</tbody>
</table>

*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:

<table>
<thead>
<tr>
<th><strong>Clock icon or the file URL</strong></th>
<th>A view detailing the history of this particular file.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Changeset ID next to the file URL</strong></td>
<td>A view of the complete file. Amended lines are highlighted on the left in yellow.</td>
</tr>
<tr>
<td><strong>Down arrow icon</strong></td>
<td>Option to download the file.</td>
</tr>
<tr>
<td><strong>Change Indicator (+n -n)</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

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**

Searching in the Manage Files View

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

**Screenshot: The Search View in the Manage Files Dialog**

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**

**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**
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.

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

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
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**

<table>
<thead>
<tr>
<th>Repository:</th>
<th>CLOV</th>
<th>Author:</th>
<th>(any)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branch:</td>
<td>(any)</td>
<td>Tag:</td>
<td></td>
</tr>
<tr>
<td>Add to Review as:</td>
<td>an Iteration</td>
<td>Remove all revisions from review</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Whole File</th>
<th>Adds the entire file with all content, rather than just a diff with context.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Diff to Previous Version</strong></td>
<td>This is the default behaviour, and how previous versions of Crucible worked by default.</td>
</tr>
<tr>
<td><strong>An Iteration</strong></td>
<td>This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.</td>
</tr>
<tr>
<td><strong>Diff to Last Reviewed Version</strong></td>
<td>This creates a review with a diff to the last reviewed changeset.</td>
</tr>
<tr>
<td><strong>Diff to... (a particular revision)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Whole file</strong></td>
<td>The review content is not limited by changesets, showing the entire file contents.</td>
</tr>
</tbody>
</table>

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*

![](image)

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.

1. 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,

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.

⚠️ 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*
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:

   ![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:**

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

⚠️ 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

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*

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*

**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
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.
  - 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**

**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**

**Flagging Defects**

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

**Screenshot: Defects**

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:

![Complete](image)

**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

- View drafts
- Delete drafts
- Post drafts

*Screenshot: Review complete*
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'.

---

**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.
Screenshot 1: The 'Email Review' option in Crucible

Screenshot 2: The 'Recipients' Screen in Crucible

Screenshot 3: The 'Message' Screen in Crucible
Using JIRA Integration in Crucible Reviews

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 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 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

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
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
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

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.
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**

![Create Issue](image)

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

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

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 Review

Summarize the review outcomes (optional)
Testing successful. Thanks for your time, everyone.

Review Closed

Review Summary
Testing successful. Thanks for your time, everyone.

Reviews requiring your attention:

<table>
<thead>
<tr>
<th>Task</th>
<th>Review</th>
<th>Owner</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review</td>
<td>CR.FE-2170</td>
<td></td>
<td>03 Jul</td>
</tr>
</tbody>
</table>
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**

![Screenshot: Deleting a Review in Crucible](image)

**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**

**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*

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.

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

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

**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.

<table>
<thead>
<tr>
<th>Sub-Tab Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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*
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*
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
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*
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
- Adding Items to Your Favourites
  - Adding a Review to Your Favourites
  - Adding a Review Comment Thread to Your Favourites
  - Adding a Project to Your Favourites
  - Adding a Person to Your Favourites
  - Adding a Changeset to Your Favourites
  - Adding a File or Folder to Your Favourites
  - Adding a Repository to Your Favourites
- Viewing Your Favourite Items
- Renaming an Item In Your Favourites
- Removing an Item From Your Favourites

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
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

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
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.

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.
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

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

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
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

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**

---

**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

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Opens reference list of keyboard shortcuts.</td>
</tr>
<tr>
<td>escape</td>
<td>Closes reference list of keyboard shortcuts.</td>
</tr>
<tr>
<td>alt</td>
<td>Hold down then click and drag to select source line contents.</td>
</tr>
<tr>
<td>shift + f</td>
<td>Toggle full screen review mode.</td>
</tr>
</tbody>
</table>

### Comment Navigation Shortcuts

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Go to next review comment.</td>
</tr>
<tr>
<td>p</td>
<td>Go to previous review comment.</td>
</tr>
<tr>
<td>shift + p</td>
<td>Go to first review comment.</td>
</tr>
<tr>
<td>shift + n</td>
<td>Go to last review comment.</td>
</tr>
<tr>
<td>l</td>
<td>Go to next thread (skips replies).</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1.Biggest heading</td>
<td>Turns text into a heading at size 1.</td>
</tr>
<tr>
<td></td>
<td><strong>Biggest Text</strong></td>
</tr>
<tr>
<td>h2.Bigger heading</td>
<td>Turns text into a heading at size 2.</td>
</tr>
<tr>
<td></td>
<td><strong>Bigger heading</strong></td>
</tr>
<tr>
<td>h3.Big heading</td>
<td>Turns text into a heading at size 3.</td>
</tr>
<tr>
<td></td>
<td><strong>Big heading</strong></td>
</tr>
<tr>
<td>h4.Normal heading</td>
<td>Turns text into a heading at size 4.</td>
</tr>
<tr>
<td></td>
<td><em>Normal heading</em></td>
</tr>
<tr>
<td>h5.Small heading</td>
<td>Turns text into a heading at size 5.</td>
</tr>
<tr>
<td></td>
<td><strong>Small heading</strong></td>
</tr>
<tr>
<td>h6.Smallest heading</td>
<td>Turns text into a heading at size 6.</td>
</tr>
<tr>
<td></td>
<td><strong>Smallest heading</strong></td>
</tr>
</tbody>
</table>

### Text Effects

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>bold</em></td>
<td>Makes text appear <strong>bold</strong>.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Makes text appear in <em>italics</em>.</td>
</tr>
<tr>
<td>+underline+</td>
<td>Makes text appear underlined.</td>
</tr>
<tr>
<td>??citation??</td>
<td>Makes text appear in <em>citation</em> form.</td>
</tr>
<tr>
<td>-strikethrough-</td>
<td>Makes text appear struck through.</td>
</tr>
<tr>
<td>^superscript^</td>
<td>Makes text appear in superscript.</td>
</tr>
<tr>
<td><del>subscript</del></td>
<td>Makes text appear in subscript.</td>
</tr>
</tbody>
</table>
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

**Example:**

```
here is quoteable content to be quoted
```

**{color:red}**

look ma, red text!

**Example:** look ma, red text!

### Text Breaks

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(empty line)</td>
<td>Produces a new paragraph</td>
</tr>
</tbody>
</table>
| \
| Creates a line break. Not often needed, most of the time the wiki renderer will guess new lines for you appropriately. |

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>Produces em dash — symbol.</td>
</tr>
<tr>
<td>–</td>
<td>Produces en dash – symbol.</td>
</tr>
</tbody>
</table>

### Links

Creating links is easy with Wiki Markup.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="CR-FE-100">Crucible Issue CR-FE-100</a></td>
<td>Creates a link to a Crucible review or FishEye artifact using the internal key reference for the item.</td>
</tr>
</tbody>
</table>
Link Examples:

- Link to an anchor on this page:
  - anchor link to top

  [Atlassian Crucible](http://atlassian.com)

Examples:

- [http://www.atlassian.com/crucible](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](mailto:mail@example.com)

  Creates a link to an email address.

  Example:
  - mail@example.com

- [file:///c:/temp/foo.txt](file:///c:/temp/foo.txt)
  - [file:///z:/file/on/network/share.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".

- [My Page#here](#anchor)

  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.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
</table>
| * A bulleted list
  * Second item
  ** indented item 1
  ** indented item 2
| Examples:
  * A bulleted list
  * Second item
  ** indented item 1
  ** indented item 2 |
| # A numbered 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.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="http://www.host.com/image.gif" alt="http://www.host.com/image.gif" /></td>
<td>The image will be displayed from the remote source.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The code above produces a table that looks like this:

<table>
<thead>
<tr>
<th>heading 1</th>
<th>heading 2</th>
<th>heading 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>col A1</td>
<td>col A2</td>
<td>col A3</td>
</tr>
<tr>
<td>col B1</td>
<td>col B2</td>
<td>col B3</td>
</tr>
</tbody>
</table>

**Advanced Formatting**

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{noformat}</td>
<td>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:</td>
</tr>
</tbody>
</table>

This is a no-formatted piece of text, so *no* _formatting_ is done here.
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:

```plaintext
Some text in a basic panel
```

```plaintext
My Title
Some text with a title
```

```plaintext
My Title
a block of text surrounded with a panel
yet another line
```

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**:

```plaintext
Bar.java
```

```plaintext
// Some comments here
public String getFoo()
{
    return foo;
}
```

**A basic display with XML code**:

```plaintext
<test>
    <another tag="attribute"/>
</test>
```
**Miscellaneous Markup Features**

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\X</td>
<td>Escape special character X (i.e. {)</td>
</tr>
<tr>
<td>:)</td>
<td>Graphical emoticons (smileys): 😊, 😊</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notation</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>:)</td>
<td>😊</td>
</tr>
<tr>
<td>:(</td>
<td>😞</td>
</tr>
<tr>
<td>:P</td>
<td>😤</td>
</tr>
<tr>
<td>:D</td>
<td>😞</td>
</tr>
<tr>
<td>;)</td>
<td>😊</td>
</tr>
<tr>
<td>(y)</td>
<td>😊</td>
</tr>
<tr>
<td>(n)</td>
<td>😞</td>
</tr>
<tr>
<td>(i)</td>
<td>😞</td>
</tr>
<tr>
<td>(/)</td>
<td>😞</td>
</tr>
<tr>
<td>(x)</td>
<td>😊</td>
</tr>
<tr>
<td>(l)</td>
<td>😞</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notation</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+)</td>
<td>🔴</td>
</tr>
<tr>
<td>(-)</td>
<td>🔴</td>
</tr>
<tr>
<td>(?)</td>
<td>🔴</td>
</tr>
<tr>
<td>(on)</td>
<td>🔴</td>
</tr>
<tr>
<td>(off)</td>
<td>🔴</td>
</tr>
<tr>
<td>(*)</td>
<td>🔴</td>
</tr>
<tr>
<td>(*r)</td>
<td>🔴</td>
</tr>
<tr>
<td>(*g)</td>
<td>🔴</td>
</tr>
<tr>
<td>(*b)</td>
<td>🔴</td>
</tr>
<tr>
<td>(*y)</td>
<td>🔴</td>
</tr>
</tbody>
</table>

**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.
   - 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
The Crucible Command Line Backup Process

1. Your Crucible instance must be running during the backup.

2. Open a command line interface on the Crucible server computer.
3. Navigate to the FISHEYE_HOME/bin/ directory.
4. Run the backup command on the command line with the desired options.
5. 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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Purpose</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Database</td>
<td>Refers to the SQL content database (used by both FishEye and Crucible and containing all user profile data, reviews and their comments).</td>
<td>Backed up by default.</td>
</tr>
<tr>
<td>Cache</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>Plugins</td>
<td>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).</td>
<td>Configuration data for all plugins are backed up by default, as well as all plugins installed in FISHEYE_INST/var/plugins/user.</td>
</tr>
</tbody>
</table>
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
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Switch</th>
<th>Description</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiet mode</td>
<td>-q OR --quiet</td>
<td>Suppresses output</td>
<td>No</td>
</tr>
<tr>
<td>Output filename</td>
<td>-f OR --file</td>
<td>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.</td>
<td>FISHEYE_INST is the default</td>
</tr>
<tr>
<td>Compression level</td>
<td>--compression OR -c</td>
<td>Sets the Zip compression level, from 1-9. Runs at level 6 if no argument is passed.</td>
<td>Yes (6)</td>
</tr>
</tbody>
</table>
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 default, the cache data is excluded from 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

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

Backup including cache data (also includes all default components)

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

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

```bash
$ ./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.

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

**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 archive 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.

<table>
<thead>
<tr>
<th>Option name</th>
<th>Description</th>
<th>Allowed Values</th>
</tr>
</thead>
</table>


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

**Screenshot: Scheduling Backups in FishEye and Crucible**

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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Switch</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress output</td>
<td>--quiet OR -q</td>
<td>Suppress the output messages from the restore program on the command line.</td>
<td>No</td>
</tr>
<tr>
<td>Choose file to restore from</td>
<td>--file PATH/FILENAME OR -f PATH/FILENAME</td>
<td>Restore the backup from PATH/FILENAME.</td>
<td>Yes (required)</td>
</tr>
<tr>
<td>Show inline help</td>
<td>--help OR -h</td>
<td>Displays help for options on the command line.</td>
<td>No</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restore FishEye cache</td>
<td>--cache</td>
<td>Restore the repository cache backup.</td>
</tr>
<tr>
<td>Restore plugins</td>
<td>--plugins</td>
<td>Restore 3rd-party plugins and their configuration data.</td>
</tr>
<tr>
<td>Restore templates</td>
<td>--templates</td>
<td>Restore freemarker templates from the backup (the restored instance will use the built-in templates).</td>
</tr>
<tr>
<td>Restore uploads</td>
<td>--uploads</td>
<td>Restore uploads (e.g. patch files uploaded into Crucible and contents of files under review).</td>
</tr>
<tr>
<td>Restore Crucible reviews</td>
<td>--sql</td>
<td>Restore the SQL database containing user profiles, reviews and review comments.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set database type</td>
<td>--dbtype OR -t</td>
<td>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.</td>
</tr>
<tr>
<td>Set JDBC URL</td>
<td>--jdbcurl OR -j</td>
<td>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').</td>
</tr>
<tr>
<td>Set JDBC username</td>
<td>--username OR -u</td>
<td>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').</td>
</tr>
<tr>
<td>JDBC password</td>
<td>--password OR -p</td>
<td>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').</td>
</tr>
<tr>
<td>JDBC class</td>
<td>--driver OR -d</td>
<td>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 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'.)</td>
</tr>
</tbody>
</table>

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

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)

```bash
$ ./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
- Creating a Permission Scheme
- Editing a Permission Scheme
- List of Crucible Permissions

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).
5. Your new permission scheme will have the default assignees shown in the table above.
6. To edit the assignees for a permission, click the 'Edit' link corresponding to the permission. The 'Edit Permission Scheme' page will be displayed.
7. 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
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
List of Crucible Permissions

The following permissions are available:

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
<th>Default Assignees</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Edit'</td>
<td>Ability to edit a review's details and change the set of revisions being reviewed.</td>
<td>'Creator' 'Moderator'</td>
</tr>
<tr>
<td>'View'</td>
<td>Ability to view a review. (People without this permission will not know that the review exists.)</td>
<td>Anonymous users All logged-in users 'Creator' 'Author' 'Reviewer' 'Moderator'</td>
</tr>
<tr>
<td>'Abandon'</td>
<td>Ability to abandon (i.e. cancel) a review.</td>
<td>'Moderator' 'Creator'</td>
</tr>
<tr>
<td>'Re-Open'</td>
<td>Ability to re-open a closed or abandoned review.</td>
<td>'Creator' 'Moderator'</td>
</tr>
<tr>
<td>'Uncomplete'</td>
<td>Ability of a reviewer to change their individual review status from 'Complete' to 'Uncomplete'.</td>
<td>'Reviewer'</td>
</tr>
<tr>
<td>'Reject'</td>
<td>Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).</td>
<td>'Moderator'</td>
</tr>
<tr>
<td>'Complete'</td>
<td>Ability of a reviewer to change their individual review status to 'Complete'.</td>
<td>'Reviewer'</td>
</tr>
<tr>
<td>'Comment'</td>
<td>Ability to add or remove a comment to or from a review.</td>
<td>'Creator' 'Author' 'Reviewer' 'Moderator'</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>'Approve'</td>
<td>Ability to approve a review (i.e. issue it to the reviewers).</td>
<td>'Moderator'</td>
</tr>
<tr>
<td>'Submit'</td>
<td>Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).</td>
<td>'Creator' 'Author'</td>
</tr>
<tr>
<td>'Close'</td>
<td>Ability to close a review once it has been summarised.</td>
<td>'Moderator'</td>
</tr>
<tr>
<td>'Delete'</td>
<td>Ability to delete a review.</td>
<td>'Creator' 'Moderator'</td>
</tr>
<tr>
<td>'Summarise'</td>
<td>Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)</td>
<td>'Moderator'</td>
</tr>
<tr>
<td>'Create'</td>
<td>Ability to create a review.</td>
<td>All logged-in users</td>
</tr>
<tr>
<td>'Recover'</td>
<td>Ability to resurrect an abandoned (i.e. cancelled) review.</td>
<td>'Creator' 'Moderator'</td>
</tr>
</tbody>
</table>

### 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
- Overview of the Permission Schemes Bundled with Crucible
  - Default Permission Scheme Settings
  - Agile Permission Scheme Settings
- Related Links

#### 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.
   - 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.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Anonymous</th>
<th>All Logged In</th>
<th>Individuals</th>
<th>Groups</th>
<th>Review Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandon</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Moderator</td>
</tr>
<tr>
<td>Approve</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Moderator</td>
</tr>
<tr>
<td>Close</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Moderator</td>
</tr>
<tr>
<td>Comment</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
<tr>
<td>Complete</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer</td>
</tr>
<tr>
<td>Create</td>
<td>false</td>
<td>true</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Delete</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Moderator</td>
</tr>
<tr>
<td>Edit Review Details</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Moderator</td>
</tr>
<tr>
<td>Recover</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Moderator</td>
</tr>
<tr>
<td>Reject</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Moderator</td>
</tr>
<tr>
<td>Re-Open</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Moderator</td>
</tr>
<tr>
<td>Submit</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator</td>
</tr>
<tr>
<td>Summarize</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Moderator</td>
</tr>
<tr>
<td>Uncomplete</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer</td>
</tr>
<tr>
<td>View</td>
<td>false</td>
<td>true</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Anonymous</th>
<th>All Logged In</th>
<th>Individuals</th>
<th>Groups</th>
<th>Review Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandon</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Author, Moderator</td>
</tr>
<tr>
<td>Approve</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Author, Moderator</td>
</tr>
<tr>
<td>Close</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
<tr>
<td>Comment</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
<tr>
<td>Complete</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer</td>
</tr>
<tr>
<td>Create</td>
<td>false</td>
<td>true</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Delete</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Author, Moderator</td>
</tr>
<tr>
<td>Edit Review Details</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
<tr>
<td>Recover</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Reviewer, Creator, Author, Moderator</td>
</tr>
<tr>
<td>Reject</td>
<td>false</td>
<td>false</td>
<td>None</td>
<td>None</td>
<td>Creator, Author, Moderator</td>
</tr>
</tbody>
</table>
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.
   - 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**
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.

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
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.

<table>
<thead>
<tr>
<th>Template filename</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>crucible-notification-subject.ftl</td>
<td>Subject template</td>
</tr>
<tr>
<td>crucible-notification-header.ftl</td>
<td>Header template</td>
</tr>
<tr>
<td>crucible-notification-footer.ftl</td>
<td>Footer template</td>
</tr>
<tr>
<td>state-closed-notification.ftl</td>
<td>State Closed template</td>
</tr>
<tr>
<td>all-completed-notification.ftl</td>
<td>All Completed template</td>
</tr>
<tr>
<td>state-changed-notification.ftl</td>
<td>State Changed template</td>
</tr>
<tr>
<td>completed-notification.ftl</td>
<td>Completed template</td>
</tr>
<tr>
<td>general-notification.ftl</td>
<td>General notification template</td>
</tr>
<tr>
<td>uncompleted-notification.ftl</td>
<td>'Uncompleted' template</td>
</tr>
<tr>
<td>all-no-longer-completed-notification.ftl</td>
<td>All-No-Lighter-Completed template</td>
</tr>
<tr>
<td>comment-notification.ftl</td>
<td>Comment template</td>
</tr>
<tr>
<td>reply-notification.ftl</td>
<td>Reply template</td>
</tr>
<tr>
<td>review-precis-plain.ftl</td>
<td>Precis template</td>
</tr>
</tbody>
</table>

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.

```java
${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*
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.

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing</td>
<td>The defect applies to code or information that is missing (absent).</td>
</tr>
<tr>
<td>Extra (superfluous)</td>
<td>The defect applies to code or information that should be removed.</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>The defect applies to code or information that is not clear or easy to understand.</td>
</tr>
<tr>
<td>Inconsistent</td>
<td>The defect applies to code or information that is applied in several different ways.</td>
</tr>
<tr>
<td>Improvement desirable</td>
<td>The defect applies to code or information that needs to be revised.</td>
</tr>
<tr>
<td>Not conforming to standards</td>
<td>The defect applies to code or information that breaks established conventions.</td>
</tr>
<tr>
<td>Risk-prone</td>
<td>The defect applies to code or information that takes unacceptable risks.</td>
</tr>
<tr>
<td>Factualy incorrect</td>
<td>The defect applies to code or information that is wrong.</td>
</tr>
<tr>
<td>Not implementable</td>
<td>The defect applies to code or information that may be impossible to create.</td>
</tr>
<tr>
<td>Editorial</td>
<td>The defect applies to code or information where the classification as a defect may be subject to personal opinion.</td>
</tr>
</tbody>
</table>

Screenshot: Editing Defect Classifications in Crucible
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 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.

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
Using HTML

The content in the welcome screen can be arranged using basic HTML tables, image references or anchor tags such as the following:

```html
<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`:
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

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**
### Admin Menu

- Repository Settings
  - Repository List (new...)
  - Repository Defaults
- Project Settings
  - Project List (new...)
- Global Settings
  - Server Settings
  - Security
  - Users
  - User Mapping
  - Avatar Settings
  - Groups
  - Administrators
  - ViewCVS URL Mappings
  - Change Admin Password
  - Customize Crucible Defect Classifications
  - Permission Schemes
  - Trusted Applications
  - JIRA Servers
  - Customize Front Page

### System

- Database Configuration
- Sys-Info/Support
- Content
- Backup
- Plugins
- Shutdown

---

### Admin: Edit Project

#### Identification

- **Name:** Clover Eclipse Plugin
- **Key:** ORCEP

#### Default Review Content Repository

- **Default Repository:** GLOV

  - **Store the contents of files in reviews**

#### Moderator

- **Default Moderator:** installation

#### 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

- **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.

---

Save

Preview
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

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
- Adding a New JIRA Server
- Editing Default JIRA Server Mappings
- Operations on Existing Servers
  - Edit settings for an existing JIRA server
  - Edit mappings for an existing JIRA server
  - Delete an existing JIRA server
- Obtaining Subtask Values for Crucible Configuration
  - Obtaining the Subtask ID Values
- Known Issues
- See Also

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**

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**
A number of fields and options must be filled out or selected on this page. See the table below for information on each field.

<table>
<thead>
<tr>
<th>Option</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Text Field</td>
<td>A descriptive name for the JIRA server.</td>
<td>Yes</td>
</tr>
<tr>
<td>URL</td>
<td>Text Field</td>
<td>The base URL of the JIRA server, e.g. {<a href="http://jira.atlassian.com%7D">http://jira.atlassian.com}</a></td>
<td>Yes</td>
</tr>
<tr>
<td>Username</td>
<td>Text Field</td>
<td>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).</td>
<td>Yes</td>
</tr>
<tr>
<td>Password</td>
<td>Text Field</td>
<td>The password for the account on the JIRA instance.</td>
<td>Yes</td>
</tr>
<tr>
<td>Include in Activity Streams</td>
<td>Check Box</td>
<td>Allows JIRA information to appear in the Activity Streams.</td>
<td>No</td>
</tr>
<tr>
<td>Authenticate as Trusted Application</td>
<td>Check Box</td>
<td>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.</td>
<td>No</td>
</tr>
</tbody>
</table>
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**

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*

**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*
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
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
• **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:

   ```sql
   CREATE DATABASE crucible CHARACTER SET utf8 COLLATE utf8_bin;
   ```

3. You will also need to set the Server Character set 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]`:

   ```ini
   [mysqld]
default-character-set=utf8
   ```

4. Use the **status** command to verify database character encoding information:

   Screenshot: Using the MySQL Status Command
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):

```sql
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*
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
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:*

- Adding a Trusted Application
  - Configuring Identification Settings
    - URL field
    - Id field
  - Configuring Access Permissions
    - URL Patterns field
    - IP Address Patterns field
    - Certificate Timeout field
- Editing Trusted Application Settings

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.

ℹ️ 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.

ℹ️ 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.

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 is not necessary to specify anything for this field; in fact, a blank value is a sensible default. The default behavior 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`, `/foo/bar` and `/foo/bar/foobar`. 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*

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**

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**
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

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

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**

**Screenshot: Simplified Review Navigation**
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

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

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*

*Screenshot: Timeline Mode in the Crucible Review History Dialog*
"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
Threaded Comments

Crucible comments now support fully threaded discussions. Threads can also be collapsed.

Screenshot: Crucible Threaded Comments
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

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:

<table>
<thead>
<tr>
<th>JIRA Issues (3 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-2738</td>
</tr>
<tr>
<td>CRUC-2713</td>
</tr>
<tr>
<td>CRUC-2620</td>
</tr>
</tbody>
</table>

From 2.1.0 to 2.1.1

17th November 2009

This is a bugfix release.

The following issues are addressed by this release:

<table>
<thead>
<tr>
<th>JIRA Issues (25 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-2710</td>
</tr>
<tr>
<td>CRUC-2707</td>
</tr>
<tr>
<td>CRUC-2705</td>
</tr>
<tr>
<td>CRUC-2704</td>
</tr>
<tr>
<td>CRUC-2703</td>
</tr>
<tr>
<td>CRUC-2702</td>
</tr>
</tbody>
</table>

Don't try to create a revisions drop down on the review contents tab when the
<table>
<thead>
<tr>
<th>CRUC-Identification</th>
<th>Description of Issue</th>
<th>Resolution Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-2700</td>
<td>revision is in a light scm repo which doesn't implement DirectoryBrowser</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2698</td>
<td>Lucene Index for Review State Changes</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2645</td>
<td>something wrong with python syntax highlighting</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2191</td>
<td>Document supported database versions</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2118</td>
<td>Defect properties on comments are being lost</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2095</td>
<td>Activity stream says &quot;finished&quot; instead of &quot;uncompleted&quot;</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2085</td>
<td>FE PERF - fixing the people page. Based on above perf data. Goal, all cases &lt; 10s, ideally &lt; 2s</td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1818</td>
<td>Double slashes appended incorrectly at the end of the SVN URL</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1687</td>
<td>upgrade to latest jquery</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1673</td>
<td>GenericJDBCException thrown on atlaseye</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1603</td>
<td>Due date display is too exact.</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1601</td>
<td>find out the css differences between the review dir tree and the existing content dir tree</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1599</td>
<td>display settings for reviews aren't intuitive and aren't sticky</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1597</td>
<td>Need Performance Benchmarks for Crucible and Fisheye</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1449</td>
<td>Largish review beachballs on resize and when expanding FRXs</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1432</td>
<td>Page refreshes when preferences pane is closed, even if there has been no change</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1385</td>
<td>Add &quot;Change Diff&quot; button for file revision chosen from Light SCM plugins similar to files chosen from Fisheye Repositories</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1334</td>
<td>Hot Deployment of Plugin does not work</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-874</td>
<td>Documentation for Web Fragment Plugin Points</td>
<td>Closed</td>
</tr>
</tbody>
</table>

**Crucible 2.0 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.

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
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
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 it's especially useful to filter and highlight just the new comments when you come back to a review that's underway.

_Screenshot: Unread comments_
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*
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
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
Crucible 2.1 Documentation

External Databases and Backup
Crucible now supports MySQL and Postgress in addition to the embedded HSQL database. Backup and restore capabilities have also been greatly enhanced.

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:

<table>
<thead>
<tr>
<th>JIRA Issues (9 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-2317</td>
</tr>
<tr>
<td>CRUC-2295</td>
</tr>
<tr>
<td>CRUC-2214</td>
</tr>
<tr>
<td>CRUC-2203</td>
</tr>
<tr>
<td>CRUC-2135</td>
</tr>
<tr>
<td>CRUC-1856</td>
</tr>
<tr>
<td>CRUC-1742</td>
</tr>
<tr>
<td>CRUC-1734</td>
</tr>
<tr>
<td>CRUC-269</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>JIRA Issues (14 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-2160</td>
</tr>
<tr>
<td>CRUC-2128</td>
</tr>
</tbody>
</table>
### JIRA Issues (14 issues)

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-2051</td>
<td>ie7: can't view draft reviews</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2050</td>
<td>java.util.ConcurrentModificationException thrown during when removing participants</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-2015</td>
<td>RSS should have absolute URLs, not relative</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1940</td>
<td>Frequent Error When Starting A Review : error.unauthorized.stateChange.description</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1927</td>
<td>The copied reviewed file should be shown in orange color</td>
<td></td>
<td>Closed</td>
</tr>
</tbody>
</table>

**From 2.0.2 to 2.0.3**

**18th August 2009**

This is a bugfix release which includes the following issues:
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-1911</td>
<td>Case sensitivity in username</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1906</td>
<td>[git] annotation view skips chunks</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1899</td>
<td>Review updated ajax calls should be chained</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1845</td>
<td>Cannot delete abandoned review</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1833</td>
<td>NPE thrown in getIcDOFromComment</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1828</td>
<td>Make the number of connections to a db easily configurable</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1814</td>
<td>Review Deletion is not working</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1094</td>
<td>Strange problems while trying to use SCM plugin for SVN with multiple repositories</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-854</td>
<td>Author Mapping doesn't always work</td>
<td></td>
<td>Closed</td>
</tr>
</tbody>
</table>

**From 2.0.1 to 2.0.2**

**24th July 2009**

This is a bugfix release which includes the following issues:

<table>
<thead>
<tr>
<th>JIRA Issues (15 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>CRUC-1886</td>
</tr>
<tr>
<td>CRUC-1831</td>
</tr>
<tr>
<td>CRUC-1791</td>
</tr>
<tr>
<td>CRUC-1781</td>
</tr>
<tr>
<td>CRUC-1780</td>
</tr>
<tr>
<td>CRUC-1779</td>
</tr>
<tr>
<td>CRUC-1771</td>
</tr>
<tr>
<td>CRUC-1770</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Key</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>CRUC-1747</td>
</tr>
<tr>
<td>CRUC-1746</td>
</tr>
<tr>
<td>CRUC-1739</td>
</tr>
<tr>
<td>CRUC-1729</td>
</tr>
<tr>
<td>CRUC-1720</td>
</tr>
<tr>
<td>CRUC-1715</td>
</tr>
<tr>
<td>CRUC-1384</td>
</tr>
</tbody>
</table>

**From 2.0 to 2.0.1**

**14th July 2009**

This is a bugfix release which includes the following issues:

**JIRA Issues (29 issues)**

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-1743</td>
<td>Select Reviewers textbox on the Edit Review-&gt;Review Details screen will not add user to a review if the username contains &quot;.&quot; or&quot;@&quot; symbols</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1716</td>
<td>[Regression] Suggested reviewers are not appearing in the list of reviewers when they are added from the pop-up</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1703</td>
<td>Arrow &quot;Previous Comment Thread&quot; in Crucible Review throws script exception &quot;CRUCOMMENT is undefined&quot;</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1701</td>
<td>Fail to add user as reviewer in Crucible if the user's username contains dot (.)</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1691</td>
<td>Poll for state changes</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1690</td>
<td>User and Crucible links don't have hovers in Review Comment search results</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1686</td>
<td>Editing Permission Schemes throws javax.el.PropertyNotFoundException</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1683</td>
<td>Non-empty folder doesn't have comment count in tree</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1682</td>
<td>Opening a review with a 2mb patch causes the review to take 1 minute to open</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1676</td>
<td>Empty folders are incorrectly ordered</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1672</td>
<td>&quot;Create Review&quot; on browse page does not add-to-review, but always creates a new review</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1671</td>
<td>Empty folder doesn't have folder icon in tree</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>CRUC-1670</td>
<td>FRXs with the same path are collapsed in the navigation tree</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1669</td>
<td>Patch reviews shouldn't allow deletion of frxrevisions</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1664</td>
<td>Backup admin page includes all items when de-selecting all</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1657</td>
<td>draft comments are shown as unread from the reviews list</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1655</td>
<td>inline comments need revisiting in side-by-side diff mode</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1654</td>
<td>Restore user to previous position in review when reloading from a review update</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1653</td>
<td>Show &quot;reviews to do&quot; table when closing a review</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1650</td>
<td>edit revisions allows you to try and add a revision already in the review, then complains</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1638</td>
<td>terminology - should use &quot;unreviewed&quot; or &quot;not reviewed&quot;</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1636</td>
<td>inconsistent state names</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1630</td>
<td>Grey out menu items when not applicable</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1619</td>
<td>due dates aren't updated when new revisions are added</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1606</td>
<td>Sort indicator on &quot;Age&quot; column of reviews dashboard points in the wrong direction</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1598</td>
<td>Add linked reviews to the review sidebar.</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1579</td>
<td>Editing a User in crucible standalone will render an sterisk next to their name</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1567</td>
<td>Comment hovers only on text in IE8</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1443</td>
<td>Safari issue - Tools dropdown causes 'invisible scrollbars'</td>
<td>Closed</td>
<td></td>
</tr>
</tbody>
</table>

**From 2.0 Beta3 to 2.0**

**30th June 2009**

Full list of issues in this release:

**JIRA Issues (200 issues)**
<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-1749</td>
<td>XSS issue in filter parameter</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1661</td>
<td>File marked as unreviewed for defect author</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1656</td>
<td>RC3 - NPE when trying to create first project on empty CRU install</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1651</td>
<td>clicking [add] branch in SVN admin symbolic screen causes page-pop and 404</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1649</td>
<td>toggling Display Preferences &gt; Show Source in a review does nothing</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1644</td>
<td>Some display preferences don't update when clicked</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1641</td>
<td>error mapping JIRA servers</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1623</td>
<td>Admin: Default reviewers are not re-saved</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1616</td>
<td>Tune polling period.</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1615</td>
<td>Reset the delay when actively using the review</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1612</td>
<td>Move the warning to a Gmail-style overlay</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1611</td>
<td>Commenting doesn't work in Safari 4</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1609</td>
<td>CAC documentation for iterative review info from CRUC-1536</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1481</td>
<td>Warning dialogs and polling when new stuff added since last refresh</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1445</td>
<td>STUPID: old style error page when trying to access fisheye pages</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1442</td>
<td>Crucible Standalone - Logout screen has FishEye icon &amp; link</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1413</td>
<td>Adding all changesets for new files with multiple commits should show file as &quot;Added&quot;, not &quot;Modified&quot;</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1412</td>
<td>&quot;Earlier changeset&quot; and &quot;Later changeset&quot; links should be grayed out when earliest/latest already listed.</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1408</td>
<td>Images in review blend in to crucible</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1404</td>
<td>Review throws 500 error NPE</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1403</td>
<td>Rename &quot;crucibledb&quot; to &quot;sql&quot;</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>CRUC-1399</td>
<td>Default Reviewer for project is not exposed through the SPI</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1398</td>
<td>overdue but review has not been started. Setting a due date in the past causes this.</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1393</td>
<td>Refactor Email Comments Page into a Dialog</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1392</td>
<td>fix remove file behaviour of current content panel</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1391</td>
<td>Use newDirTreeBox in the files pane</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1390</td>
<td>Fix Uploads</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1388</td>
<td>Missing css files on Send Email page / Layout needs love</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1386</td>
<td>fix suggest reviewer ui</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1383</td>
<td>suggest reviewers - turn blame calculation back on</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1382</td>
<td>defect metrics charts are broken</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1381</td>
<td>.png file not detected as binary</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1380</td>
<td>Email Review button from the dashboard doesn't work / Email review in tools menu gives a 404 - permaid is undefined</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1379</td>
<td>clicking HELP opens help twice / Help in Tools menu will open a new window and change your window</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1378</td>
<td>page load error &quot;The requested URL /atlaseye/static/kidi0j2static/script/fe/fisheye-profile.js was not found on this server.&quot;</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1377</td>
<td>wrong review actions (comments.txt and Email Review) are visible</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1376</td>
<td>we need to handle multiple errors in a smarter way</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1375</td>
<td>There is no jira quick link in the edit review dialog</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1374</td>
<td>Review info popup on the dashboard shouldn't show when clicking on a link such as the review permalid</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1373</td>
<td>Page load error &quot;The requested URL /atlaseye/static/kidi0j2static/images/sprite_arrows.png was not found on this server&quot;</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1371</td>
<td>External dialog 'View ...' links need to redirect to review and trigger filter</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1370</td>
<td>Show current content panel in manage files</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1369</td>
<td>make the side filters like the fisheye tabs</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1368</td>
<td>bring back the custom filter</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1367</td>
<td>make the pages consistent</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1366</td>
<td>Add warnings for incomplete FRXs and unresolved JIRAs to dialogs</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1365</td>
<td>Move review summary form into a dialog.</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1364</td>
<td>Move creation-time review suggestions into a dialog</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1363</td>
<td>Convert project selection screen to a dialog</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1362</td>
<td>Backup broke Crucible Blitz</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1361</td>
<td>General Issues</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1360</td>
<td>Manage Files</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1359</td>
<td>Edit review form</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1358</td>
<td>Interstitials/dialogs</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1357</td>
<td>Dashboard Fixes</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1356</td>
<td>Remove driver url parameter from Restore and use new DBInfo properties</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1354</td>
<td>Refactor programatic frx filtering to not invoke click events</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1351</td>
<td>Add path and base filename to backup UI</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1345</td>
<td>Extend QuartzManager to support persistent configuration data per job</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1344</td>
<td>Rest Api returns the wrong URL for download Patch</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1332</td>
<td>Going to full context doesn't work</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-ID</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>CRUC-1330</td>
<td>Put custom quartz job scheduling in a separate tab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-1329</td>
<td>Dynamically update general comment counts</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1328</td>
<td>Add general comments to the tree</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1327</td>
<td>Add the BackupItem selection to the scheduled backup</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1325</td>
<td>Add file type css place holders</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1324</td>
<td>FishEye only indicate Crucible review for a file with latest revision</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1319</td>
<td>Update doco for admin backup page</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-1317</td>
<td>Dynamically update comment counts</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1316</td>
<td>Add comment counts to the frx tree</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1305</td>
<td>Springify Backup Manager</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1303</td>
<td>Write unit test</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1294</td>
<td>Check start/stop no longer needed in restore</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1293</td>
<td>Render maintenance page for non-admin pages when a backup is being made</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1291</td>
<td>Add &quot;Change Diff&quot; dropdown to manipulate both fromRev and toRev independently</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-1290</td>
<td>Collapse common directories</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1289</td>
<td>Apply tree html structure and css styling to list</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1288</td>
<td>Treeify FRX list pane</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1286</td>
<td>add other filters</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1285</td>
<td>add filter - with comments</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1284</td>
<td>add UI for dropdown and expand/collapse</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1283</td>
<td>Drop-down filter for frx tree</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1275</td>
<td>Convert stored dates to long</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
<td>---</td>
<td>--------</td>
</tr>
<tr>
<td>CRUC-1274</td>
<td>Create FRX anchors and adjust the browser location as you navigate</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1273</td>
<td>FRXs should start expanded and maintain position when loading frxs above</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1272</td>
<td>Told that other participants have drafts, but also that all other participants have zero drafts</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1271</td>
<td>Move edit review box into AJS.Dialog</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1270</td>
<td>Integrate the summarise and close pages</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1254</td>
<td>Rearrange content</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1253</td>
<td>Review meta information sub pane</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1252</td>
<td>Move manage files tabs into AJS.Dialog</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1251</td>
<td>scroll frx pane when clicking on frx list item</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1250</td>
<td>create floating frx info subpane which updates on scroll</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1248</td>
<td>Modify overview header</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1247</td>
<td>3 Pane UI</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1242</td>
<td>Delete the old backup mechanism</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1241</td>
<td>Rewire the quartz job</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1235</td>
<td>Revision slider hardening/tweaking</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1233</td>
<td>Wrap option</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1232</td>
<td>Delete comment refactor</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1215</td>
<td>Quartz backup job must invoke the new xml-based backup</td>
<td></td>
<td>Resolved</td>
</tr>
<tr>
<td>CRUC-1208</td>
<td>reorganise contents of static directory</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1206</td>
<td>add new preference for new frxrevisions</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1205</td>
<td>add new notification for adding frxRevisions</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1201</td>
<td>displaying frxs does not take into account showAsDiff flag</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1179</td>
<td>Sort FRX - alpha server side sort</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1174</td>
<td>Add banner warning &quot;you have unresolved jiras&quot; to page</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1173</td>
<td>Add resolve all JIRAs button &amp; action to summarize page</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1171</td>
<td>XML-RPC bollocks</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1170</td>
<td>Delete a defect's linked JIRA subtask when deleting the defect</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1169</td>
<td>ui &amp; action</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1168</td>
<td>XML-RPC bollocks</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1167</td>
<td>Resolve a defect's linked JIRA subtask from Crucible</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1166</td>
<td>XML-RPC bollocks</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1165</td>
<td>UI display of jira link and status + post error creating</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1164</td>
<td>schema</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1163</td>
<td>Automatically add defects as subtasks to JIRA &amp; link to them</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1162</td>
<td>guessing + jira validation</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1161</td>
<td>edit details ui + actions</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1160</td>
<td>schema</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1159</td>
<td>Link a review to a single JIRA and enable auto defect creation</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1158</td>
<td>just do it</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1157</td>
<td>Configure a JIRA instance for use by Crucible</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1156</td>
<td>admin UI + create review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-1155</td>
<td>schema</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1154</td>
<td>Configure a default review duration for a project in week days so the due date can be pre-populated</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1153</td>
<td>highlighting overdue</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1152</td>
<td>sorting in the lists</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1151</td>
<td>Order any list of reviews by due date &amp; highlight any that are overdue</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1150</td>
<td>due date field in UI</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1149</td>
<td>schema</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1148</td>
<td>Set or change a due date on a review</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1143</td>
<td>just do it</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1142</td>
<td>Combine diff and annotation views</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1141</td>
<td>just do it</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1140</td>
<td>show the reviews you can’t update without buttons</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1139</td>
<td>Author colours</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1138</td>
<td>Invisible markers (between lines)</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1137</td>
<td>Mouseover</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1136</td>
<td>Improve comment marker behaviour (tetris bar)</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1134</td>
<td>replace yahoo shite if we can</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1133</td>
<td>event binding</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1132</td>
<td>error handling</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1131</td>
<td>namespacing &amp; multiple files</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1130</td>
<td>remove prototype</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>CRUC-1129</td>
<td>client side js model</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1128</td>
<td>refactor commentator.js and friends</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1127</td>
<td>just do it</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1126</td>
<td>handle modifier keypress</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1125</td>
<td>just do it</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1124</td>
<td>Popup a keyboard shortcut help overlay when i press ‘?’</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1123</td>
<td>add user pref</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1122</td>
<td>add complete status resetter to the defect notification</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1121</td>
<td>Reset my complete status when new defects are added to a review</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1119</td>
<td>Tweak comment scroll tracker behaviour</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1118</td>
<td>Improve revision slider behaviour</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1117</td>
<td>add unread comment counts per revision</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1114</td>
<td>fix rendering/script issue in IE</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1113</td>
<td>make the slider handles move in unison</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1108</td>
<td>Mark comment read when it hits top of the viewport</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1105</td>
<td>adding to review doesn’t work properly when you add the first file after a branch</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1097</td>
<td>Remove the use of half closed TCP connections</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1096</td>
<td>Change SPI to support multiple revisions per FRX</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1092</td>
<td>Show user’s lastname and firstname when selecting a reviewer</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1089</td>
<td>Crucible Project should move from Global Settings to own Section</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>Issue Key</td>
<td>Description</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>CRUC-1083</td>
<td>render author + participant results</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1062</td>
<td>api changes for multi-revisions and revision comments</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1060</td>
<td>Comment-counts displayed are not taking replies into consideration</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1057</td>
<td>Expand Review Summary Email</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-1048</td>
<td>Change the admin backup page to use the new backup/restore implementation</td>
<td>Resolved</td>
<td></td>
</tr>
<tr>
<td>CRUC-1044</td>
<td>When upgrading from fisheye to crucible, the UI does not allow you to enter the Crucible license</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-977</td>
<td>issues comments added on changed and unchanged code don't show up none or less context</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-971</td>
<td>CRU Defect-&gt; JIRA Subtask creation</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-940</td>
<td>Comments made on a line not in the diff is not shown</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-927</td>
<td>Add a warning in the product that re-sync in large LDAP servers can cause performance issue</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-922</td>
<td>Admin screens with config.xml write and create/migrate workflow</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-921</td>
<td>Set up maintenance Mode</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-920</td>
<td>set up tests to compare upgrades of each database type</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-918</td>
<td>Write tests to determine/compare expected results from database tables</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-915</td>
<td>Usage Documentation for web-based backup/restore feature</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-912</td>
<td>Integration Tests</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-910</td>
<td>Admin Backup Page</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-909</td>
<td>Backup Status Page</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-907</td>
<td>Online Backup/Restore</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-882</td>
<td>REST API return other people's drafts</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-861</td>
<td>Allow soft wrap on diff and annotated display</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>CRUC-860</td>
<td>Autosave has issues</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-858</td>
<td>&quot;Lines are missing&quot; divider graphics are no longer there</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-845</td>
<td>Describe this in the CAC docs.</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-825</td>
<td>remove abandon button from edit details section</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-824</td>
<td>If we haven't already we should automatically create a backup of our CRUDB (and other configuration files) before upgrading</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-823</td>
<td>Add an Expand Unchecked Files option</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-813</td>
<td>Side-by-side diff mode is too wide</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-770</td>
<td>Support file upload via REST</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-745</td>
<td>Setup page doesn't warn of invalid fe licence when valid cru licence is provided</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-741</td>
<td>Change moderator via REST</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-734</td>
<td>&quot;Side-by-side&quot; Diffs option does not update upon refreshing the Firefox browser</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-603</td>
<td>Support Anonymous Users Properly</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-581</td>
<td>Include Light SCM Repos in information returned by REST API</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-537</td>
<td>Summary Report &amp; Searching by File</td>
<td></td>
<td>Closed</td>
</tr>
</tbody>
</table>

**From 2.0 Beta2 to 2.0 Beta3**

**5th June 2009**

Full list of issues in this release:

<table>
<thead>
<tr>
<th>JIRA Issues (35 issues)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td>CRUC-1524</td>
<td>review actions from hovers sometimes aren't bound and dont work.</td>
</tr>
<tr>
<td>CRUC-1515</td>
<td>File time stamps are not preserved in the backup</td>
</tr>
<tr>
<td>CRUC-1513</td>
<td>Permalinks to hidden comments are broken</td>
</tr>
<tr>
<td>CRUC-1512</td>
<td>Crucible user preferences should be saved automatically when changed</td>
</tr>
<tr>
<td>CRUC-1508</td>
<td>Pressing ‘y’ in fullscreen mode skips an frx</td>
</tr>
<tr>
<td>CRUC-1507</td>
<td>Next/Prev FRX buttons</td>
</tr>
<tr>
<td>CRUC-1506</td>
<td>Pressing ‘y’ in review comments doesn't do anything.</td>
</tr>
<tr>
<td>CRUC-1505</td>
<td>Marking a file as reviewed should mark all frx comments as read</td>
</tr>
<tr>
<td>CRUC-1504</td>
<td>Error dialog box background css shows entire sprite</td>
</tr>
<tr>
<td>CRUC-1503</td>
<td>Defect subtasks have their assignee set to default assignee when resolved from Crucible</td>
</tr>
<tr>
<td>CRUC-1501</td>
<td>1.6 -&gt; 2.0 beta upgrade truncates timestamps</td>
</tr>
<tr>
<td>CRUC-1499</td>
<td>IDEA direct link port changed from 6666 to 51234</td>
</tr>
<tr>
<td>CRUC-1496</td>
<td>FRX reloads cause review model sort problems</td>
</tr>
<tr>
<td>CRUC-1493</td>
<td>Show a message in the frx pane when all files are filtered</td>
</tr>
<tr>
<td>CRUC-1492</td>
<td>Files from different repos breaks Manage Files</td>
</tr>
<tr>
<td>CRUC-1491</td>
<td>I think it should be sticky. i.e. set your preference each time you toggle it.</td>
</tr>
<tr>
<td>CRUC-1487</td>
<td>Need to show initial sort order on the Review dashboard</td>
</tr>
<tr>
<td>CRUC-1486</td>
<td>Adding a changeset to a review results in duplicate FRXs</td>
</tr>
<tr>
<td>CRUC-1484</td>
<td>Linked subtask in comment doesn't have a JIRA hover</td>
</tr>
<tr>
<td>CRUC-1480</td>
<td>Revision slider doesn't resize properly when you edit revisions</td>
</tr>
<tr>
<td>CRUC-1477</td>
<td>Clicking on a source line when creating a file comment kills the file comment</td>
</tr>
<tr>
<td>CRUC-1475</td>
<td>Moving slider on image diff causes NPE</td>
</tr>
<tr>
<td>CRUC-1474</td>
<td>Suggested reviews dialog uses old style inline review dialog</td>
</tr>
<tr>
<td>CRUC-1473</td>
<td>Review actions missing from inline review dialog</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-1488</td>
<td>Improve user interface in the frx tree</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1478</td>
<td>Single frx visible mode</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1476</td>
<td>File revision comments aren't autosaved</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1465</td>
<td>Synchronise comment buttons when states are changed</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1464</td>
<td>Fix rev slider labels so that slider can snap</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1463</td>
<td>User has no display name</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1462</td>
<td>Selecting changeset in Manage Files fails</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1458</td>
<td>Account signup pages still use old styles</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1454</td>
<td>Timezone problem in review popup</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1439</td>
<td>Crucible Standalone - Manage Files links to FishEye</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1427</td>
<td>Old error page for disabled FishEye Urls</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-1396</td>
<td>Empty reviews cannot be abandoned/closed/viewed</td>
<td></td>
<td>Closed</td>
</tr>
</tbody>
</table>
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.

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

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
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

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**

<table>
<thead>
<tr>
<th>Due</th>
<th>Reviewers</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 days ago</td>
<td><img src="image1" alt="Avatar" /></td>
</tr>
<tr>
<td>11 days ago</td>
<td><img src="image2" alt="Avatar" /></td>
</tr>
<tr>
<td>10 days ago</td>
<td><img src="image3" alt="Avatar" /> <img src="image4" alt="Avatar" /></td>
</tr>
<tr>
<td>3 days ago</td>
<td><img src="image5" alt="Avatar" /> <img src="image6" alt="Avatar" /></td>
</tr>
<tr>
<td>4 hours ago</td>
<td><img src="image7" alt="Avatar" /> <img src="image8" alt="Avatar" /></td>
</tr>
<tr>
<td>3 hours ago</td>
<td><img src="image9" alt="Avatar" /> <img src="image10" alt="Avatar" /></td>
</tr>
<tr>
<td>79 minutes ago</td>
<td><img src="image11" alt="Avatar" /> <img src="image12" alt="Avatar" /></td>
</tr>
</tbody>
</table>

**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**
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*
Plus numerous improvements and bug fixes

Visit our issue tracker 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.

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.

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.

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:

- Opening the Administration Screen for JIRA Integration
- Adding a New JIRA Server
  - Obtaining the Subtask Type ID
  - Obtaining the Subtask Resolution ID and Subtask Resolution Action ID
- Editing Default JIRA Server Mappings
- Operations on Existing Servers
  - Edit settings for an existing JIRA server
  - Edit mappings for an existing JIRA server
  - Delete an existing JIRA server

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**

<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
<th>Subtask ID</th>
<th>Subtask Resolution Action ID</th>
<th>Subtask Resolution ID</th>
<th>Allow Unassigned</th>
<th>Activity Stream Provider</th>
<th>Trusted Application</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>jira</td>
<td><a href="https://jira.atlassian.com">https://jira.atlassian.com</a></td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>Edit Mappings Delete</td>
</tr>
</tbody>
</table>

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**
A number of fields and options must be filled out or selected on this page. See the table below for information on each field.

<table>
<thead>
<tr>
<th>Option</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Text Field</td>
<td>A descriptive name for the JIRA server.</td>
<td>Yes</td>
</tr>
<tr>
<td>URL</td>
<td>Text Field</td>
<td>The Internet address of the JIRA server.</td>
<td>Yes</td>
</tr>
<tr>
<td>Subtask Type ID</td>
<td>Number</td>
<td>This is required to enable creating issues from a Crucible comment.</td>
<td>No</td>
</tr>
<tr>
<td>Subtask Resolution Action ID</td>
<td>Number</td>
<td>This is required to enable creating issues from a Crucible comment. See below for instructions on obtaining this number.</td>
<td>No</td>
</tr>
<tr>
<td>Subtask Resolution ID</td>
<td>Number</td>
<td>This is required to enable creating issues from a Crucible comment.</td>
<td>No</td>
</tr>
<tr>
<td>Allow Unassigned</td>
<td>True/False Button</td>
<td>Allow unassigned sub-tasks.</td>
<td>No</td>
</tr>
<tr>
<td>Username</td>
<td>Text Field</td>
<td>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).</td>
<td>Yes</td>
</tr>
<tr>
<td>Password</td>
<td>Text Field</td>
<td>The password for the account on the JIRA instance.</td>
<td>Yes</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>Include in Activity Streams Check Box</td>
<td>Allows JIRA information to appear on the Dashboard.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Authenticate as Trusted Application Check Box</td>
<td>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.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

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**

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**

![Screenshot](image-url)
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

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
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

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.
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

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.
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.

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.
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.

Expanded API

The Crucible API now allows programmable review creation, along with a host of other additions. Read more.

Plus numerous improvements and bug fixes

<table>
<thead>
<tr>
<th>JIRA Issues (109 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CRUC-668</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>CRUC-664</td>
</tr>
<tr>
<td>CRUC-663</td>
</tr>
<tr>
<td>CRUC-662</td>
</tr>
<tr>
<td>CRUC-661</td>
</tr>
<tr>
<td>CRUC-659</td>
</tr>
<tr>
<td>CRUC-654</td>
</tr>
<tr>
<td>CRUC-647</td>
</tr>
<tr>
<td>CRUC-646</td>
</tr>
<tr>
<td>CRUC-645</td>
</tr>
<tr>
<td>CRUC-640</td>
</tr>
<tr>
<td>CRUC-639</td>
</tr>
<tr>
<td>CRUC-635</td>
</tr>
<tr>
<td>CRUC-634</td>
</tr>
<tr>
<td>CRUC-632</td>
</tr>
<tr>
<td>CRUC-626</td>
</tr>
<tr>
<td>CRUC-623</td>
</tr>
<tr>
<td>CRUC-622</td>
</tr>
<tr>
<td>CRUC-615</td>
</tr>
<tr>
<td>CRUC-610</td>
</tr>
<tr>
<td>CRUC-608</td>
</tr>
</tbody>
</table>

Improve the efficiency of checking whether a review has been completed by all
<table>
<thead>
<tr>
<th>CRUC-602</th>
<th>reviewers</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-600</td>
<td>REST API should return error when non-existent filter used</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-598</td>
<td>Emails always include a “Summary” section</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-597</td>
<td>Extend the Remote API to allow reviews to be deleted</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-594</td>
<td>Review description preview on dashboard swallows new lines from the review description</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-592</td>
<td>Filter names used in Cru do not match the menus</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-591</td>
<td>Renaming a confluence repository adds a new one instead of editing</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-589</td>
<td>Change revisionData getAdded/Removed lines an Integer</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-588</td>
<td>Search Subtab in Review-&gt;Manage Files throws PropertyNotFoundException for all searches</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-586</td>
<td>Remove Delete option from project drop down</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-579</td>
<td>Changing repository when certain review tabs are selected causes NPE/bad redirect</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-574</td>
<td>Allowed Review Participants left blank means what?</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-566</td>
<td>lightscm package should not be “fisheye”</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-565</td>
<td>Need get bulk version of getRevisionData() in SCMRepository</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-563</td>
<td>Toggle side-by-side and show per frx diff options on review page</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-562</td>
<td>REST API: createReview does not set Moderator and author correctly</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-561</td>
<td>Anchors to some comment types don’t work</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-560</td>
<td>REST-API: provide commit type information - esp. deleted file status</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-556</td>
<td>Next/previous comment arrows do not jump between file and general comments</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-554</td>
<td>Upgrade to non-beta atlassian-plugins</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-551</td>
<td>Confluence Light SCM plugin dependencies belong in plugin</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-549</td>
<td>Adjust IFRAME size when the admin page is resized</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-ID</td>
<td>Description</td>
<td>Status</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>CRUC-548</td>
<td>REST API: Handle allowReviewersToJoin flag on review</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-547</td>
<td>Make Repository source check if the engine is available.</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-545</td>
<td>NPE Closing Review</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-542</td>
<td>Use content manager to retrieve detailed file revisions</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-540</td>
<td>make metrics-config.xsd available online</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-539</td>
<td>Serious: &quot;Next comment&quot; does not always work right</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-526</td>
<td>Space character in installation path to Crucible disables remote API</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-519</td>
<td>Added files display &quot;no change&quot;</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-518</td>
<td>Crucible does not seem to be incrementally updating the FishEye cache</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-516</td>
<td>REST API: Return detailed information on review in one call - items, comments, reviewers, available actions</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-511</td>
<td>Added files when stored show &quot;No Change&quot;</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-507</td>
<td>Confluence: implement change set paging</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-505</td>
<td>Add “Search for Review” functionality to Crucible remote API</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-504</td>
<td>Confluence: get revision comments</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-499</td>
<td>Create separate API jar</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-496</td>
<td>Document store revisions with review</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-494</td>
<td>Don't make source type visible in URLs or elsewhere</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-488</td>
<td>clicking summarise without clicking close makes the review ‘disappear’ from the workflow</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-487</td>
<td>Page title says &quot;dead&quot; for a review that has been abandoned</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-484</td>
<td>you can link a review to itself</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-479</td>
<td>A comment which was a defect, but is no longer, still shows the defect attributes</td>
<td>Closed</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>CRUC-478</td>
<td>Ability to specify that Branches are not supported</td>
<td></td>
</tr>
<tr>
<td>CRUC-472</td>
<td>Changing tab while editing review deletes details</td>
<td></td>
</tr>
<tr>
<td>CRUC-469</td>
<td>Sort list of repositories alphabetically on &quot;Add project&quot; page</td>
<td></td>
</tr>
<tr>
<td>CRUC-461</td>
<td>Line comments are sometimes rendered twice</td>
<td></td>
</tr>
<tr>
<td>CRUC-457</td>
<td>show author name next to revision in revision dropdowns</td>
<td></td>
</tr>
<tr>
<td>CRUC-456</td>
<td>crucible doesn’t create default permission scheme when creating a blank db</td>
<td></td>
</tr>
<tr>
<td>CRUC-455</td>
<td>creating a review from a changeset with a non-existent repo keeps redirecting to the login screen</td>
<td></td>
</tr>
<tr>
<td>CRUC-453</td>
<td>Allow adding a screenshot as an attachment to a review</td>
<td></td>
</tr>
<tr>
<td>CRUC-439</td>
<td>Invite to review</td>
<td></td>
</tr>
<tr>
<td>CRUC-438</td>
<td>Not sending email notifications after Crucible and Fisheye upgrade</td>
<td></td>
</tr>
<tr>
<td>CRUC-436</td>
<td>create ‘remove all revision’ link for all tabs under ‘manage files’</td>
<td></td>
</tr>
<tr>
<td>CRUC-433</td>
<td>REST API: Get review list based on predefined and custom filters</td>
<td></td>
</tr>
<tr>
<td>CRUC-422</td>
<td>Create Schema Upgrade to remove foreign key constraint as reqd. by delete project work</td>
<td></td>
</tr>
<tr>
<td>CRUC-419</td>
<td>Typo in email &quot;reviewers are now complete&quot;</td>
<td></td>
</tr>
<tr>
<td>CRUC-407</td>
<td>Minimal OSGi Infrastructure</td>
<td></td>
</tr>
<tr>
<td>CRUC-404</td>
<td>Turn FE Off When only Crucible licence is present</td>
<td></td>
</tr>
<tr>
<td>CRUC-402</td>
<td>Provide Plugin Authors with somewhere to store properties</td>
<td></td>
</tr>
<tr>
<td>CRUC-401</td>
<td>Real Admin Users</td>
<td></td>
</tr>
<tr>
<td>CRUC-400</td>
<td>Add Configuration UI to SVN Plugin, and Polish</td>
<td></td>
</tr>
<tr>
<td>CRUC-399</td>
<td>Optionally Store Files for all FileRevisions</td>
<td></td>
</tr>
<tr>
<td>CRUC-397</td>
<td>Stored FRX Create UI</td>
<td></td>
</tr>
<tr>
<td>CRUC-396</td>
<td>Stored FRX Data Impl</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-394</td>
<td>Add Light SCM Revisions to Reviews</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-392</td>
<td>Filesystem Light SCM Plugin</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-391</td>
<td>Default Repository for a Project can be a Light SCM repo</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-389</td>
<td>Light SCM Plugin instances appear in source/repository drop-downs</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-388</td>
<td>Subversion LSCM plugin</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-387</td>
<td>Confluence LSCM Plugin</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-386</td>
<td>Modify File Browser to Use Light SCM Plugins</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-385</td>
<td>Modify Changeset Browser to use Light SCM Plugins</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-384</td>
<td>Create Light SCM Module Type</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-381</td>
<td>Admin Pages for Plugins</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-373</td>
<td>Add General Comment via REST API</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-362</td>
<td>Return replies to comments via API</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-361</td>
<td>Project RSS feed</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-349</td>
<td>REST method /rest-service/reviews-v1/CR-1/comments returns HTTP 500 error</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-303</td>
<td>Add REST method to retrieve all reviews which involve a given file</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-302</td>
<td>Add REST methods to allow review creation</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-297</td>
<td>Upload files for review (not patches)</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-292</td>
<td>javax.xml.bind.JAXBException generated when invoking the getGeneralComments web service method</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-244</td>
<td>Add revisions not diffs to a review</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-227</td>
<td>Cannot delete projects</td>
<td>Closed</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>JIRA Issues (2 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>CRUC-1406</td>
</tr>
<tr>
<td>CRUC-900</td>
</tr>
</tbody>
</table>

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) |
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:

<table>
<thead>
<tr>
<th>JIRA Issues (27 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>CRUC-926</td>
</tr>
<tr>
<td>CRUC-862</td>
</tr>
<tr>
<td>CRUC-846</td>
</tr>
<tr>
<td>CRUC-844</td>
</tr>
<tr>
<td>CRUC-837</td>
</tr>
<tr>
<td>CRUC-835</td>
</tr>
<tr>
<td>CRUC-834</td>
</tr>
<tr>
<td>CRUC-832</td>
</tr>
<tr>
<td>CRUC-827</td>
</tr>
<tr>
<td>CRUC-826</td>
</tr>
<tr>
<td>CRUC-821</td>
</tr>
<tr>
<td>CRUC-820</td>
</tr>
<tr>
<td>CRUC-809</td>
</tr>
<tr>
<td>CRUC-802</td>
</tr>
<tr>
<td>CRUC-793</td>
</tr>
<tr>
<td>CRUC-788</td>
</tr>
<tr>
<td>CRUC-787</td>
</tr>
<tr>
<td>CRUC-778</td>
</tr>
<tr>
<td>CRUC-765</td>
</tr>
<tr>
<td>CRUC-753</td>
</tr>
<tr>
<td>CRUC-748</td>
</tr>
<tr>
<td>CRUC-727</td>
</tr>
<tr>
<td>CRUC-582</td>
</tr>
<tr>
<td>CRUC-568</td>
</tr>
<tr>
<td>CRUC-398</td>
</tr>
<tr>
<td>CRUC-328</td>
</tr>
<tr>
<td>CRUC-255</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>JIRA Issues (78 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>CRUC-785</td>
</tr>
<tr>
<td>CRUC-784</td>
</tr>
<tr>
<td>CRUC-777</td>
</tr>
<tr>
<td>CRUC-776</td>
</tr>
<tr>
<td>CRUC-775</td>
</tr>
<tr>
<td>CRUC-773</td>
</tr>
<tr>
<td>CRUC-772</td>
</tr>
<tr>
<td>CRUC-771</td>
</tr>
<tr>
<td>CRUC-769</td>
</tr>
<tr>
<td>CRUC-768</td>
</tr>
<tr>
<td>CRUC-767</td>
</tr>
<tr>
<td>CRUC-764</td>
</tr>
<tr>
<td>CRUC-763</td>
</tr>
<tr>
<td>CRUC-759</td>
</tr>
<tr>
<td>CRUC-758</td>
</tr>
<tr>
<td>CRUC-757</td>
</tr>
<tr>
<td>CRUC-752</td>
</tr>
<tr>
<td>CRUC-ID</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>CRUC-750</td>
</tr>
<tr>
<td>CRUC-749</td>
</tr>
<tr>
<td>CRUC-747</td>
</tr>
<tr>
<td>CRUC-744</td>
</tr>
<tr>
<td>CRUC-742</td>
</tr>
<tr>
<td>CRUC-739</td>
</tr>
<tr>
<td>CRUC-736</td>
</tr>
<tr>
<td>CRUC-735</td>
</tr>
<tr>
<td>CRUC-733</td>
</tr>
<tr>
<td>CRUC-729</td>
</tr>
<tr>
<td>CRUC-728</td>
</tr>
<tr>
<td>CRUC-725</td>
</tr>
<tr>
<td>CRUC-724</td>
</tr>
<tr>
<td>CRUC-723</td>
</tr>
<tr>
<td>CRUC-722</td>
</tr>
<tr>
<td>CRUC-713</td>
</tr>
<tr>
<td>CRUC-712</td>
</tr>
<tr>
<td>CRUC-711</td>
</tr>
<tr>
<td>CRUC-710</td>
</tr>
<tr>
<td>CRUC-709</td>
</tr>
<tr>
<td>CRUC-708</td>
</tr>
<tr>
<td>CRUC-706</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>CRUC-705</td>
</tr>
<tr>
<td>CRUC-702</td>
</tr>
<tr>
<td>CRUC-701</td>
</tr>
<tr>
<td>CRUC-700</td>
</tr>
<tr>
<td>CRUC-699</td>
</tr>
<tr>
<td>CRUC-698</td>
</tr>
<tr>
<td>CRUC-697</td>
</tr>
<tr>
<td>CRUC-696</td>
</tr>
<tr>
<td>CRUC-695</td>
</tr>
<tr>
<td>CRUC-694</td>
</tr>
<tr>
<td>CRUC-693</td>
</tr>
<tr>
<td>CRUC-691</td>
</tr>
<tr>
<td>CRUC-689</td>
</tr>
<tr>
<td>CRUC-688</td>
</tr>
<tr>
<td>CRUC-687</td>
</tr>
<tr>
<td>CRUC-680</td>
</tr>
<tr>
<td>CRUC-679</td>
</tr>
<tr>
<td>CRUC-678</td>
</tr>
<tr>
<td>CRUC-677</td>
</tr>
<tr>
<td>CRUC-671</td>
</tr>
<tr>
<td>CRUC-670</td>
</tr>
<tr>
<td>CRUC-667</td>
</tr>
<tr>
<td>CRUC-665</td>
</tr>
<tr>
<td>CRUC-658</td>
</tr>
<tr>
<td>CRUC-656</td>
</tr>
<tr>
<td>CRUC-655</td>
</tr>
<tr>
<td>CRUC-653</td>
</tr>
<tr>
<td>CRUC-627</td>
</tr>
<tr>
<td>CRUC-624</td>
</tr>
<tr>
<td>CRUC-536</td>
</tr>
<tr>
<td>CRUC-529</td>
</tr>
<tr>
<td>CRUC-509</td>
</tr>
<tr>
<td>CRUC-486</td>
</tr>
<tr>
<td>CRUC-475</td>
</tr>
<tr>
<td>CRUC-470</td>
</tr>
<tr>
<td>CRUC-395</td>
</tr>
<tr>
<td>CRUC-383</td>
</tr>
<tr>
<td>CRUC-380</td>
</tr>
<tr>
<td>CRUC-327</td>
</tr>
</tbody>
</table>

*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

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
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
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.

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

<table>
<thead>
<tr>
<th>JIRA Issues (38 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-344</td>
</tr>
<tr>
<td>CRUC-332</td>
</tr>
<tr>
<td>CRUC-324</td>
</tr>
<tr>
<td>CRUC-323</td>
</tr>
<tr>
<td>CRUC-318</td>
</tr>
<tr>
<td>CRUC-314</td>
</tr>
<tr>
<td>CRUC-307</td>
</tr>
<tr>
<td>CRUC-299</td>
</tr>
<tr>
<td>CRUC-290</td>
</tr>
<tr>
<td>CRUC-289</td>
</tr>
<tr>
<td>CRUC-285</td>
</tr>
<tr>
<td>CRUC-283</td>
</tr>
<tr>
<td>CRUC-282</td>
</tr>
<tr>
<td>CRUC-280</td>
</tr>
<tr>
<td>CRUC-266</td>
</tr>
<tr>
<td>CRUC-260</td>
</tr>
<tr>
<td>CRUC-248</td>
</tr>
<tr>
<td>CRUC-247</td>
</tr>
<tr>
<td>CRUC-246</td>
</tr>
<tr>
<td>CRUC-242</td>
</tr>
<tr>
<td>CRUC-241</td>
</tr>
<tr>
<td>CRUC-240</td>
</tr>
<tr>
<td>CRUC-238</td>
</tr>
<tr>
<td>CRUC-232</td>
</tr>
<tr>
<td>CRUC-230</td>
</tr>
<tr>
<td>CRUC-226</td>
</tr>
<tr>
<td>CRUC-220</td>
</tr>
<tr>
<td>CRUC-212</td>
</tr>
<tr>
<td>CRUC-206</td>
</tr>
<tr>
<td>CRUC-201</td>
</tr>
<tr>
<td>CRUC-194</td>
</tr>
<tr>
<td>CRUC-184</td>
</tr>
<tr>
<td>CRUC-172</td>
</tr>
<tr>
<td>CRUC-162</td>
</tr>
<tr>
<td>CRUC-147</td>
</tr>
</tbody>
</table>
CRUC-128  email notification of review closure prints literal "null" for absent field value

CRUC-85  Sometimes we need to see differences in UNICODE files.

CRUC-21  change icon for "View History In Fisheye"

**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)

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-638</td>
<td>How to (easily) add specific changeset to a review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-633</td>
<td>How to remove a reviewer from a review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-538</td>
<td>Comments weirdly duplicated on occasion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-513</td>
<td>If we display no change on a revision, comments made are hidden by default.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-503</td>
<td>Upgrade to crowd-integration-client 1.4.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-498</td>
<td>Draft comment replies are deleted when completing review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-454</td>
<td>two notification emails sent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-429</td>
<td>Prevent two projects with same key different case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-417</td>
<td>large var/data/crubb/crucible.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-284</td>
<td>AnnotatorTag error: Could not get blame for file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**From 1.5.2 to 1.5.3**

**23 June 2008**

This release contains bug fixes.

### JIRA Issues (1 issues)

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-462</td>
<td>&quot;File closed: var\data\data0.bin&quot; after backups</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# From 1.5.1 to 1.5.2

**27 May 2008**

This release contains bug fixes.

<table>
<thead>
<tr>
<th>JIRA Issues (16 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-434</td>
</tr>
<tr>
<td>CRUC-413</td>
</tr>
<tr>
<td>CRUC-406</td>
</tr>
<tr>
<td>CRUC-405</td>
</tr>
<tr>
<td>CRUC-376</td>
</tr>
<tr>
<td>CRUC-371</td>
</tr>
<tr>
<td>CRUC-370</td>
</tr>
<tr>
<td>CRUC-363</td>
</tr>
<tr>
<td>CRUC-358</td>
</tr>
<tr>
<td>CRUC-355</td>
</tr>
<tr>
<td>CRUC-354</td>
</tr>
<tr>
<td>CRUC-336</td>
</tr>
<tr>
<td>CRUC-335</td>
</tr>
<tr>
<td>CRUC-262</td>
</tr>
<tr>
<td>CRUC-233</td>
</tr>
<tr>
<td>CRUC-185</td>
</tr>
</tbody>
</table>
From 1.5.0 to 1.5.1

24 April 2008

This release contains bug fixes.

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-367</td>
<td>LOC exceptions filling Debug log to the tune of 3-6GB per day</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-364</td>
<td>Starting Crucible 1.5 Spits Lots of Errors Out</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-357</td>
<td>Sparklines have wrong content type</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-353</td>
<td>Crucible seems not to be able to handle uploaded git patches</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-350</td>
<td>javax.servlet.ServletException</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-333</td>
<td>In IE, “view diff to latest” partially obscured</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-304</td>
<td>Add REST method to retrieve review items for a given review</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-291</td>
<td>NullPointerException thrown when invoking the getAllRevisionComments web</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td>service method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRUC-161</td>
<td>Create document stating when a customer purchases Crucible that they do</td>
<td></td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td>not need to have the latest version of FishEye</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
**Crucible 2.1 Documentation**

- 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).
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.
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.

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.
**JIRA integration**

The new version 1.2 of the FishEye-for-JIRA plugin includes some useful improvements:

- 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: 🍀

---

**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.

---

**Plus over 20 improvements and bug-fixes**

<table>
<thead>
<tr>
<th>JIRA Issues (32 issues)</th>
<th>Priority</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-181 Old screenshot in 1.2 release notes</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-175 LHS abandon button on Crucible review screen broken</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-166 Some files not work</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-152 sysinfo admin screen should show both CRU and FE license string</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-146 Add review information to ALL search result pages</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-143 add review-constraints in the EyeQL where clause</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-140 ensure &quot;linked&quot; reviews are exported thru data in remote api</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-122 Move review to another project</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
<tr>
<td>CRUC-121 EyeQL return clause: reviews</td>
<td>🔄</td>
<td>🍀 Closed</td>
</tr>
</tbody>
</table>
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).

<table>
<thead>
<tr>
<th>JIRA Issues (26 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key</strong></td>
</tr>
<tr>
<td>CRUC-274</td>
</tr>
</tbody>
</table>

CRUC-115  Accept Patch Review From Clipboard
CRUC-113  Allow creation of reviews for multiple changesets through /cru/create URL
CRUC-109  Presented with Post/delete drafts buttons when no comments drafted made
CRUC-107  Ability to change username
CRUC-104  AnnotatorTag error reviewing patch
CRUC-102  Create Project Model object and Hibernate DB upgrade
CRUC-101  Add ‘Projects’
CRUC-95   Change Diff Buttons Losing State
CRUC-94   Show Existing Reviews In Fisheye
CRUC-93   Crucible should preserve request params/URLs through login redirects
CRUC-90   More Administrator Options
CRUC-89   Add “Default Reviewers’
CRUC-88   Review Groups
CRUC-86   add “allow anyone” as a per-project default
CRUC-80   Should be able to create a review when there are no configured repositories
CRUC-78   merge cru/fe src and content trees
CRUC-73   Generate HEAD review from directory
CRUC-68   Dragging to deselect source lines no longer works
CRUC-65   beta plugin api
CRUC-61   Author should be able to “complete”
CRUC-56   Self-registration
CRUC-43   Webservice API for Reviews
CRUC-36   Should my review status go back to incomplete if I start adding comments?
<table>
<thead>
<tr>
<th>CRUC-273</th>
<th>Create IDE integration with Eclipse</th>
<th>◼️</th>
<th>◼️</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRUC-258</td>
<td>Allow all users to be able to a &quot;Moderator&quot; or &quot;Author&quot; in a review project</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-256</td>
<td>Check boxes for checking off files reviewed in a review are lost if navigate to raw text by &quot;download raw text&quot; icon</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-249</td>
<td>Accept review comments in Japanese (unicode characters)</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-243</td>
<td>race condition between crucible backup and repository scan</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-235</td>
<td>File tabs should be kept open</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-210</td>
<td>Create JIRA issues from code reviews</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-208</td>
<td>Nullpointer exception when creating new review</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-202</td>
<td>update 'Repositories' screenshot</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-196</td>
<td>Error on attempting to create a Review.</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-180</td>
<td>'Show Full Source' should continue to highlight the diff</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-177</td>
<td>Comments on reviews are being sent but don't appear in the review when viewed online</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-165</td>
<td>Folder Issue</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-139</td>
<td>Can't add new versions of the same file to a review</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-127</td>
<td>Improperly stopping crucible causes data loss!</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-112</td>
<td>Point online help links to new Crucible doc space</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-106</td>
<td>Add content to new documentation page ' Crucible and FishEye'</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-105</td>
<td>Upload PDF, XML and HTML versions of Crucible docs</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-103</td>
<td>Create admin page for Projects</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-100</td>
<td>Move Crucible docs to Confluence</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
<tr>
<td>CRUC-99</td>
<td>Feature request: I would like to see statistics about number of major and minor issues found per developer.</td>
<td>◼️</td>
<td>◼️</td>
<td>Closed</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>JIRA Issues (3 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>CRUC-207</td>
</tr>
<tr>
<td>CRUC-204</td>
</tr>
<tr>
<td>CRUC-195</td>
</tr>
</tbody>
</table>

From 1.2 to 1.2.1

This is a small bug-fix release.

<table>
<thead>
<tr>
<th>JIRA Issues (6 issues)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
</tr>
<tr>
<td>CRUC-199</td>
</tr>
<tr>
<td>CRUC-198</td>
</tr>
<tr>
<td>CRUC-182</td>
</tr>
<tr>
<td>CRUC-179</td>
</tr>
<tr>
<td>CRUC-178</td>
</tr>
</tbody>
</table>
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 freezes unexpectedly
- JIRA Integration Issues
- Problems with very long comments and MySQL migration

**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

...  

...  

Caused by: java.sql.BatchUpdateException: Lock wait timeout exceeded; try restarting transaction

at com.mchange.v2.c3p0.impl.NewProxyPreparedStatement.executeBatch(NewProxyPreparedStatement.java:1723)
at org.hibernate.jdbc.BatchingBatcher.doExecuteBatch(BatchingBatcher.java:48)
at org.hibernate.jdbc.AbstractBatcher.executeBatch(AbstractBatcher.java:246)

... 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.DBEditHelper doGet
- 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.jmxremote
wrapper.java.additional.5=-Dcom.sun.management.jmxremote.port=42...
t.jmxremote.password.file=./wrapper/jmxremote.password
wrapper.java.additional.10=-Dwrapper.mbean.name="wrapper:type=Java
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:\path\to\FISHEYE_INST"
wrapper.java.additional.12=-XX:MaxPermSize=128m
wrapper.java.additional.13=-Xrs
```

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 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
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 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:

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
<th>Default Assignees</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Edit'</td>
<td>Ability to edit a review's details and change the set of revisions being reviewed.</td>
<td>'Creator' 'Moderator'</td>
</tr>
</tbody>
</table>
| '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
'Creator'
'Moderator' |
| '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:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>See Creating a Review.</td>
</tr>
<tr>
<td>Require Approval</td>
<td>Relevant only when the moderator is not the creator. See Issuing a Review.</td>
</tr>
<tr>
<td>Under Review</td>
<td>See Issuing a Review and Reviewing the Code.</td>
</tr>
<tr>
<td>Summarize</td>
<td>See Summarising and Closing the Review.</td>
</tr>
<tr>
<td>Closed</td>
<td>See Summarising and Closing the Review.</td>
</tr>
</tbody>
</table>

† 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:

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abandoned</td>
<td>This happens when a review is deleted.</td>
</tr>
<tr>
<td>Rejected</td>
<td>Any reviews that a moderator has rejected.</td>
</tr>
</tbody>
</table>

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.
- Software Engineering Institute web page: Software Inspections.
- NASA Software Assurance Technology Center web page: Software Formal Inspections.