


Space Details

Key:	CRUCIBLE
Name:	Crucible 1.2
Description:	Latest documentation for Crucible: do your code reviews, your way
Creator (Creation Date):	(Oct 04, 2007)
Last Modifier (Mod. Date):	rosie@atlassian.com (Dec 04, 2007)

Available Pages

- Crucible Documentation Home 
 - About Crucible
 - Background Reading
 - Crucible Administrator's Guide
 - Backups and Review Data
 - Creating a Permission Scheme
 - Associating a Permission Scheme with a Project
 - Creating a Project
 - Crucible and FishEye
 - Customising the Defect Classifications
 - Crucible Development Hub
 - Crucible Installation & Upgrade Guide
 - Crucible Installation Guide
 - 1. System Requirements
 - 2. Installing Crucible
 - 3. Configuring Crucible
 - Crucible Release Notes
 - Crucible 1.1 Release Notes
 - Crucible 1.1 Changelog
 - Crucible 1.1 Upgrade Guide
 - Crucible 1.2 Release Notes
 - Crucible 1.2 Upgrade Guide
 - Crucible Upgrade Guide
 - Upgrading from FishEye to Crucible
 - Upgrading to a New Version of Crucible
 - Crucible Knowledge Base
 - Troubleshooting
 - Crucible User's Guide
 - 1. Using the Dashboard
 - 2. Requesting and Conducting a Review
 - 2.1 Creating a Review
 - 2.1.1 Creating a Review from FishEye
 - 2.1.2 Creating a Review within Crucible
 - 2.1.3 Creating a Review from JIRA
 - 2.1.4 Creating a Review from a URL
 - 2.1.5 Selecting the Files for the Review

- 2.1.6 Adding Reviewers
 - 2.1.7 Creating a Patch Review
 - 2.2 Issuing a Review
 - 2.3 Performing the Review
 - 2.3.1 Adding Comments
 - 2.3.2 Flagging Defects
 - 2.3.3 Completing your Review
 - 2.4 Summarising and Closing the Review
- 3. Changing your User Profile
- Glossary
 - approve
 - author
 - code review
 - comment
 - creator
 - defect
 - moderator
 - participant
 - permission
 - permission scheme
 - project
 - reviewer
 - role
 - state
 - statement of objective
 - user

Crucible Documentation Home

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

Getting Started with Crucible 1.2	Using/Administering Crucible 1.2
About Crucible Feature Tour Download Crucible Installation Guide Release Notes Upgrade Guide Glossary	Quick Start: Conducting a Review Administrator's Guide User's Guide FishEye Documentation Support FAQ Development Hub
Previous Versions	PDF Format
Crucible 1.1 documentation Crucible 1.0 documentation	You can download the Crucible documentation in PDF format.

Error formatting macro: recently-updated: java.lang.NullPointerException

About Crucible

This page last changed on Oct 15, 2007 by [smaddox](#).

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.

To run Crucible you need a FishEye-compatible source code repository setup. At the present time that is 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

This page last changed on Oct 25, 2007 by rosie@atlassian.com.

The following resources are recommended for background reading on peer code reviews:

- [White paper](#) on effective code review by Karl Wiegers.
- Book, [Peer Reviews in Software: A Practical Guide](#) by Karl Weigers.
- Software Engineering Institute web page: [Software Inspections](#).
- NASA Software Assurance Technology Center web page: [Software Formal Inspections](#).

Crucible Administrator's Guide

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

Once you have [installed](#) and [configured](#) Crucible, you can access the Administration pages at <http://HOSTNAME:8060/admin/>.

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

For more information on administering FishEye, please refer to the [FishEye documentation](#).

Information in the Crucible Administrator's Guide:

- [Backups and Review Data](#)
- [Creating a Permission Scheme](#)
- [Creating a Project](#)
- [Crucible and FishEye](#)
- [Customising the Defect Classifications](#)

Backups and Review Data

This page last changed on Oct 11, 2007 by [smaddox](#).

Currently all review data is stored in an embedded relational database found at `FISHEYE_HOME/var/data/`. It can be safely copied whilst Crucible is running. In addition to backing up `crudb` before upgrading, you should make it part of your backup regime. For more details view the [FishEye backup guide](#).

Review data will be automatically upgraded on startup of a newer version. Therefore it is imperative that you back up your data before upgrading. The review database can be upgraded, but not downgraded. In the unlikely event that a serious bug is introduced, you will need a backup to roll back to the previous version.

Creating a Permission Scheme

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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](#), i.e.:
 - ['moderator'](#)
 - ['author'](#)
 - ['reviewer'](#)
 - ['creator'](#)

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.

The following permissions are available:

Permission	Description	Default Assignees
'Edit'	Ability to edit a review's details and change the set of revisions being reviewed.	'Moderator' 'Creator'
'View'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users 'Moderator' 'Creator' 'Author' 'Reviewer'
'Abandon'	Ability to abandon (i.e. cancel) a review.	'Moderator' 'Creator'
'Re-Open'	Ability to re-open a closed or abandoned review.	'Moderator' 'Creator'
'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.	'Moderator' 'Creator' 'Author' 'Reviewer'
'Approve'		'Moderator'

	Ability to approve a review (i.e. issue it to the reviewers).	
'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'
'Summarise'	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	'Moderator'
'Create'	Ability to create a review.	All logged-in users
'Recover'	Ability to resurrect an abandoned (i.e. cancelled) review.	'Moderator' 'Creator'

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 2 below).
- i** Your new permission scheme will have the default assignees shown in the table above.
5. To edit the assignees for a permission, click the 'Edit' link corresponding to the permission. The 'Edit Permission Scheme' page will be displayed (see screenshot 3 below).
 6. Choose the appropriate assignee(s) for this permission:

! Note: for ongoing ease of management, it is recommended that you grant permissions to [groups](#) or [participants](#) rather than to individual users.

 - To assign this permission to [anonymous users](#), select the 'Allow Anonymous users' check-box.
 - To assign this permission to all logged-in users, select the 'Allow All logged in users' check-box.
 - To assign this permission to a particular user, type their username into the 'Individual users' field (hint: you can type just part of the name, then press <Enter> to select from a list of matching usernames).
 - To assign this permission to a particular group of users, type the group name into the 'Groups' field (hint: you can type just part of the group name, then press <Enter> to select from a list of matching groups).
 - To assign this permission to users who belong to a particular [participant](#) ('Reviewer' / 'Moderator' / 'Author' / 'Creator'), select the corresponding check-box.
 7. Click the 'Save' button.

i Next step: see [Associating a Permission Scheme with a Project](#).

Screenshot 1: Add Permission Scheme 'Top Secret'

Name	Projects using this scheme	
default	Default Project	edit copy
Name: <input type="text" value="Top Secret"/>	<input type="button" value="create"/>	<input type="button" value="cancel"/>

Screenshot 2: Edit Permission Scheme 'Top Secret'

Edit Permission Scheme - Top Secret		
<div>Top Secret</div> <div>Rename</div>		
Permissions	Users / Groups / Review Roles	
Edit Review Details Ability to change review details including the set of revisions being reviewed.	<ul style="list-style-type: none"> Anonymous users: false All logged in users: false Individual users: Groups: Review Roles: Moderator Creator 	edit
View Ability to view a review.	<ul style="list-style-type: none"> Anonymous users: true All logged in users: true Individual users: Groups: Review Roles: Moderator Reviewer Author Creator 	edit
Abandon Ability to abandon (i.e. cancel) a review.	<ul style="list-style-type: none"> Anonymous users: false All logged in users: false Individual users: Groups: Review Roles: Moderator Creator 	edit
Re-Open Ability to re-open a closed review.	<ul style="list-style-type: none"> Anonymous users: false All logged in users: false Individual users: Groups: Review Roles: Moderator Creator 	edit
Uncomplete Ability to indicate they have not completed a review, after indicating they have completed a review.	<ul style="list-style-type: none"> Anonymous users: false All logged in users: false Individual users: Groups: Review Roles: Reviewer 	edit
Reject Ability to reject a review submitted for approval.	<ul style="list-style-type: none"> Anonymous users: false All logged in users: false Individual users: 	edit

Screenshot 3: Edit Permission 'View' on Permission Scheme 'Top Secret'

Edit User Action "View" on scheme "Top Secret"	
View: Ability to view a review.	
Allow Anonymous users:	<input checked="" type="checkbox"/>
Allow All logged in users:	<input checked="" type="checkbox"/>
Individuals: <input type="text"/> Start typing a user name then press enter to select.	
Groups: <input type="text"/> Start typing a group name then press enter to select.	
Review Participants:	<input checked="" type="checkbox"/> Reviewer <input checked="" type="checkbox"/> Moderator <input checked="" type="checkbox"/> Author <input checked="" type="checkbox"/> Creator
<div>Save</div>	

Associating a Permission Scheme with a Project

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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](#), i.e.:
 - ['moderator'](#)
 - ['author'](#)
 - ['reviewer'](#)
 - ['creator'](#)

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.

To associate a permission scheme with a project,

1. From the 'Admin Menu', click 'Projects'.
2. The 'Projects List' page will be displayed. Locate the project with which you wish to associate your permission scheme, and click the corresponding 'Edit' link.
3. The 'Edit Project' page will be displayed. In the
4. In the 'Permission Scheme' drop-down list, select your permission scheme.
5. Click the 'Save' button.

Creating a Project

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

THIS PAGE IS CURRENTLY IN DRAFT


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 'Projects'.
2. The 'Projects List' page will be displayed. Click the 'Create a New Project' link, which appears below the list of existing projects.
3. The 'Create Project' page will be displayed (see screenshot below).
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 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'.)

Screenshot: 'Create Project'

Repository List	
Identification	
Name:	<input type="text"/>
Key:	<input type="text"/>
Repository	
Default Repository:	svn <input type="button" value="v"/>
Moderator	
Default Moderator:	<input type="text"/> <i>Start typing a user name then press enter to select.</i>
Default Reviewers	
<input type="checkbox"/> Let allowed review participants join a review	
Users:	<input type="text"/> <i>Start typing a user name then press enter to select.</i>
Groups:	<input type="text"/> <i>Start typing a group name then press enter to select.</i>
Allowed Review Participants	
Users:	<input type="text"/> <i>Start typing a user name then press enter to select.</i>
Groups:	<input type="text"/> <i>Start typing a group name then press enter to select.</i>
Project Permissions	
Permission Scheme:	default <input type="button" value="v"/>
<input type="button" value="Save"/>	

Crucible and FishEye

This page last changed on Oct 15, 2007 by [smaddox](#).

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.



Your Crucible installation includes a FishEye instance

Crucible includes all the functionality of FishEye. If you are running Crucible it is not necessary to run FishEye separately.

Purchasing and Installing Crucible/FishEye

When you buy Crucible, you get FishEye too.

- If you install Crucible, there is no need to do a separate installation of FishEye.
- If you already have an existing FishEye installation, you don't need to keep it. Refer to the guide on [upgrading from FishEye to Crucible](#).
- To run Crucible, you need both the FishEye and the Crucible licenses.



But for evaluation purposes, you can use just the Crucible evaluation license.

FISHEYE_HOME and FISHEYE_INST

Crucible makes use of the FishEye environment variables:

- FISHEYE_HOME – the location of the FishEye application.
- FISHEYE_INST – the location of the FishEye data.

Refer to the FishEye documentation for more about the [environment variables](#) and how they are used in the [FishEye installation](#).

Detailed Documentation

You can find more information in:

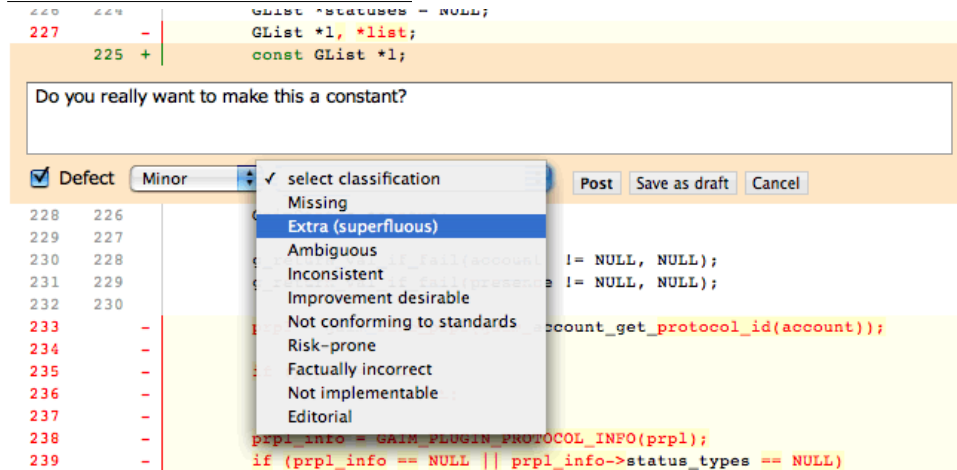
- [Crucible Installation Guide](#)
- [FishEye Installation Guide](#)
- Complete [Crucible documentation](#)
- Complete [FishEye documentation](#)

Customising the Defect Classifications

This page last changed on Oct 11, 2007 by [smaddox](#).

[Defects](#) are comments made by [reviewers](#) that indicate a problem in a review. Defects can be classified by rank and type.

Screenshot: Defect classifications




The above screenshot shows the default classifications.

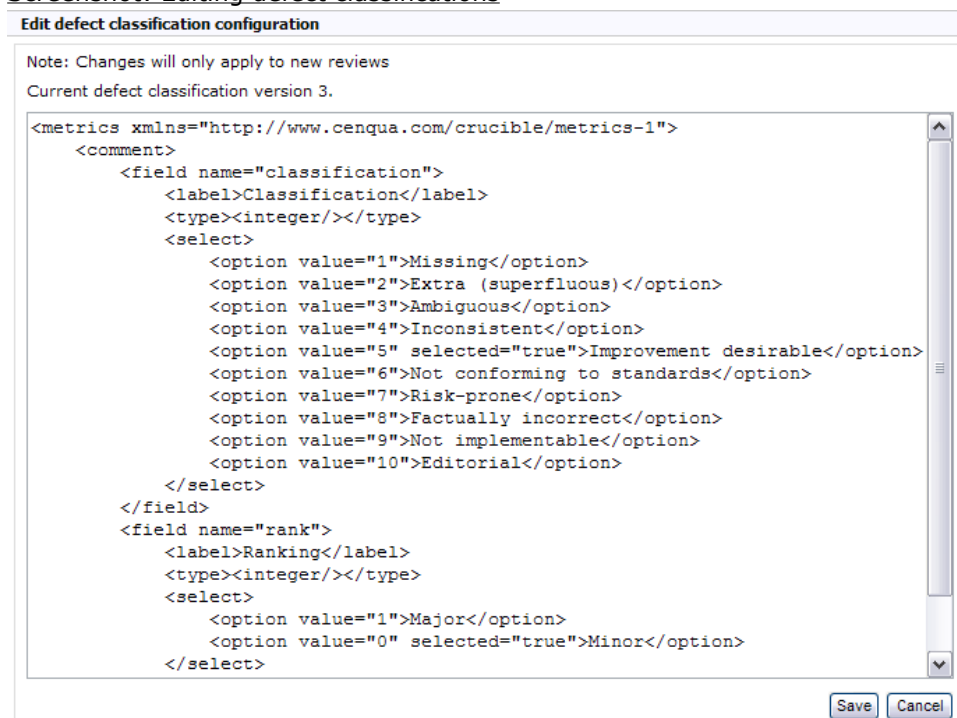
To change the default classifications:

1. Open the [FishEye Administration screens](#).
2. Click 'Customize Crucible Defect Classifications' under 'Global Settings' in the 'Admin Menu'.

Only the administrator has access to this screen.

 Any changes made within 'Customize crucible defect classifications' will only affect new reviews.

Screenshot: Editing defect classifications



Crucible Development Hub

This page last changed on Dec 03, 2007 by rosie@atlassian.com.



Warning:

The Crucible API is currently in development and is subject to change. It is not recommended for production use.

The Crucible Development Hub contains some handy resources for developers.

- [Javadocs](#)

Also see the [FishEye Development Hub](#).

Crucible Installation & Upgrade Guide

This page last changed on Oct 04, 2007 by smaddox.


- [Crucible Installation Guide](#)
 - [1. System Requirements](#)
 - [2. Installing Crucible](#)
 - [3. Configuring Crucible](#)
- [Crucible Release Notes](#)
 - [Crucible 1.1 Release Notes](#)
 - [Crucible 1.2 Release Notes](#)
- [Crucible Upgrade Guide](#)
 - [Upgrading from FishEye to Crucible](#)
 - [Upgrading to a New Version of Crucible](#)

Crucible Installation Guide

This page last changed on Oct 11, 2007 by [smaddox](#).

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.

- [1. System Requirements](#)
- [2. Installing Crucible](#)
- [3. Configuring Crucible](#)

1. System Requirements

This page last changed on Oct 11, 2007 by [smaddox](#).

These are the requirements for Crucible and FishEye. See the [FishEye requirements](#) for more specific details.

Java Runtime	A JDK or JRE version 1.5 or greater. You can download a Java Runtime for Windows/Linux/Solaris here . On MacOSX the JVM is available as part of the OS install
Source code Repository	To run Crucible you need a FishEye compatible source code repository. At the present time that is CVS, Subversion, or Perforce. ClearCase is expected during 2007.
Web Browser	<p>Crucible has been tested in the latest releases of Firefox 2, Internet Explorer 7, and Safari 2. Crucible is known to work with IE6 and FF1.5 and should work on any modern browser.</p> <div> Font size (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. FF2, IE7 and Safari now all have excellent font rendering. It is worth the time to tweak your fonts for the best experience.</div>
Operating System	Crucible is a pure Java application and should run on any platform provided the above requirements are satisfied.

Known Limitations

All artifacts to be reviewed must be checked in to a source code management system supported by [FishEye](#).

2. Installing Crucible

This page last changed on Oct 11, 2007 by [smaddox](#).

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.
3. If you intend to use Crucible/FishEye with [Subversion](#), please ensure you read the [System Requirements](#), [Subversion client setup](#), and [granting permission to FishEye](#) to scan your repository.
4. If you intend to use Crucible/FishEye with [Perforce](#), please ensure you read the [System Requirements](#) and [Perforce Client setup](#).

Next Step - Initial Configuration

See the guidelines on [configuring Crucible](#).

3. Configuring Crucible

This page last changed on Dec 04, 2007 by rosie@atlassian.com.



Read-only access for Crucible/FishEye

We recommend you run Crucible/FishEye as a user that has only read access to your repository.

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](#).

Running Crucible

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 8080 for HTTP requests. It also listens on 127.0.0.1:8059 as a control port. You can configure both of these in the FishEye 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 FishEye Administration screens.
 2. A license key. Please note your [server ID](#). You can then get a Crucible evaluation license key [here](#).
- During evaluation, only a Crucible license is needed. Once you install a commercial Crucible license, a commercial FishEye license will be required.

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 [add a repository](#).

Setting Up 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.

Setting Up Users

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

To add users:

1. Open the [FishEye Administration screens](http://HOSTNAME:8060/admin/) at <http://HOSTNAME:8060/admin/>.
2. Click 'Users/Security' under 'Global Settings' in the 'Admin Menu'.

Read more details about the different ways of [creating users](#).

Setting Up SMTP

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

First, you must [set up the SMTP Server](#).

Using Crucible

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

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

Stopping Crucible

To stop the Crucible server:

- On Windows:

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

- On Unix-based systems:

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

Crucible Release Notes

This page last changed on Oct 11, 2007 by [smaddox](#).



Crucible 1.2 has now been released. See the [Release Notes](#).

Installation

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](#).

- 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 Release Notes

- [Crucible 1.2 Release Notes](#)
- [Crucible 1.1 Release Notes](#)
- For changes prior to 1.1, see the [1.0.x Changelog](#)

Crucible 1.1 Release Notes

This page last changed on Dec 04, 2007 by rosie@atlassian.com.



Crucible 1.2 has now been released. See 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

This page last changed on Nov 08, 2007 by [cmaconeill](#).

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 were 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

This page last changed on Nov 08, 2007 by [cmacneill](#).

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 1.2 Release Notes

This page last changed on Dec 06, 2007 by [edawson](#).

December 5, 2007.

The Atlassian Crucible team is delighted to present Crucible 1.2. Crucible release 1.2 brings you a host of popular new features. You can now group your reviews into projects (similar to [JIRA](#) projects) and authorise your users via project permission schemes.

New user management screens make the administrator's job a lot easier. The new built-in integration with Atlassian [Crowd](#) extends your authentication and authorisation capabilities. You can now include users and groups from one or more Crowd directories, and provide single sign-on (SSO) across Atlassian products plus any other applications that support SSO.

Crucible's integration with [JIRA](#) and [FishEye](#) is now closer than ever before. Read the details below.

Highlights of this release:

- [Reviews grouped into projects](#)
- [Customisable permission schemes](#)
- [Plugin API](#)
- [Enhancements to user management](#)
- [JIRA integration](#)
- [Crucible 1.2 includes FishEye 1.4](#)
- [Plus over 20 improvements and bug-fixes](#)

Responding to your feedback:

★ 8 new feature requests/improvements implemented

★ 9 votes satisfied

Your [\[votes and issues\]](#)<http://jira.atlassian.com/browse/CRUC> help us keep improving our products, and are much appreciated.

Upgrading to Crucible 1.2

You can now download Crucible from [here](#). If upgrading from a previous version, please follow the [Upgrade Guide](#).

Highlights of Crucible 1.2

1

Reviews grouped into projects

- Crucible now supports [projects](#) - every review will belong to a project.
- Each project has a unique key (prefix), modelled on [JIRA](#) keys.
- You can add your own projects via the new administration screens.
- You can specify review defaults per project, such as the default users for each role and a default repository.
- And you can restrict the users/groups who can perform a particular role, e.g. only team leaders can be moderators.
- Each project has its own permission scheme (see below).

Repository List

Identification

Name:

Key:

Repository

Default Repository:

svn

Moderator

Default Moderator:

Start typing a user name then press enter to select.

Default Reviewers

☐ Let allowed review participants join a review

Users:

Start typing a user name then press enter to select.

Groups:

Start typing a group name then press enter to select.

Allowed Review Participants

Users:

Start typing a user name then press enter to select.

Groups:

Start typing a group name then press enter to select.

Project Permissions

Permission Scheme:

default

Save

2

Customisable permission schemes

- A [permission scheme](#) is a set of actions which a user can perform (e.g. create a review, approve a review, etc).
- Each project can have its own custom permission scheme — or you can use the same scheme for multiple projects.
- The permission scheme for a review is determined by the review's project.

Edit Permission Scheme - new scheme

Top Secret

Rename

Permissions	Users / Groups / Review Roles
<div>Notify file</div> <div>Ability to change the set of revisions being reviewed.</div>	<div> <div>Anonymous users:</div> <div>false</div> </div> <div> <div>All logged in users:</div> <div>false</div> </div> <div> <div>Individual users:</div> <div></div> </div> <div> <div>Groups:</div> <div></div> </div> <div> <div>Roles: Moderator: Creator</div> <div></div> </div> <div> <div>edit</div> </div>
<div>Close</div> <div>Ability to close a review once it has been summarized.</div>	<div> <div>Anonymous users:</div> <div>false</div> </div> <div> <div>All logged in users:</div> <div>false</div> </div> <div> <div>Individual users:</div> <div></div> </div> <div> <div>Groups:</div> <div></div> </div> <div> <div>Roles: Moderator</div> <div></div> </div> <div> <div>edit</div> </div>
<div>View Review</div> <div>Ability to view a review.</div>	<div> <div>Anonymous users:</div> <div>true</div> </div> <div> <div>All logged in users:</div> <div>true</div> </div> <div> <div>Individual users:</div> <div></div> </div> <div> <div>Groups:</div> <div></div> </div> <div> <div>Roles:</div> <div></div> </div> <div> <div>edit</div> </div>
<div>Uncomplete</div> <div>Ability to indicate they have not completed a review, after indicating they have completed a review.</div>	<div> <div>Anonymous users:</div> <div>false</div> </div> <div> <div>All logged in users:</div> <div>false</div> </div> <div> <div>Individual users:</div> <div></div> </div> <div> <div>Groups:</div> <div></div> </div> <div> <div>Roles: Reviewer</div> <div></div> </div> <div> <div>edit</div> </div>
<div>Re-Open</div> <div>Ability to re-open a closed review.</div>	<div> <div>Anonymous users:</div> <div>false</div> </div> <div> <div>All logged in users:</div> <div>false</div> </div> <div> <div>Individual users:</div> <div></div> </div> <div> <div>Groups:</div> <div></div> </div> <div> <div>Roles: Moderator</div> <div></div> </div> <div> <div>edit</div> </div>

3

Plugin API

- A new plugin Crucible programming interface (API), in beta for this release, supports the following functionality:
 - Create or modify reviews and comments.
 - Add files, patches, etc to reviews.
 - Invoke state transitions.
 - Add custom servlet handlers.
- More [information](#).

4

Enhancements to user management

In Crucible 1.1.2, we introduced support for public signup (self-registration). Now in Crucible 1.2:

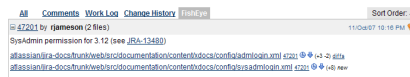
- Administrator can make the email address for [self-signups](#) optional.
- Improved user interface makes user administration easier.
- [Groups](#) are supported.
- Read the [FishEye documentation](#).

5

JIRA integration

The new version 1.2 of the [FishEye-for-JIRA plugin](#) includes some useful 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:



6

Crucible 1.2 includes FishEye 1.4

... and provides closer integration than ever before.

- FishEye screens include links to existing Crucible reviews. So you can see which files/changesets have been reviewed.
- EyeQL allows you to search for Crucible data. For example, you can search for files that have not yet been reviewed.
- Crucible now has built-in [Crowd/SSO](#) support.
- See the [FishEye 1.4 Release Notes](#).

7

Plus over 20 improvements and bug-fixes

Atlassian JIRA (32 issues)			
Key	Summary	Pr	Status
CRUC-181	Old screenshot in 1.2 release notes	↑	Closed
CRUC-175	LHS abandon button on Crucible review screen broken	↑	Resolved
CRUC-166	Some files not work	↑	Resolved
CRUC-152	sysinfo admin screen should	↑	Resolved

	show both CRU and FE license string			
CRUC-146	Add review information to ALL search result pages	↑	➡	Resolved
CRUC-143	add review-constraints in the EyeQL where clause	↑	➡	Resolved
CRUC-140	ensure "linked" reviews are exported thru data in remote api	↑	➡	Resolved
CRUC-122	Move review to another project	↑	➡	Resolved
CRUC-121	EyeQL return clause: reviews	↑	➡	Resolved
CRUC-113	Allow creation of reviews for multiple changesets through / cru/create URL	↑	➡	Resolved
CRUC-109	Presented with Post/delete drafts buttons when no comments drafted made	↑	➡	Resolved
CRUC-107	Ability to change username	↑	➡	Resolved
CRUC-104	AnnotatorTag error reviewing patch	↑	➡	Resolved
CRUC-102	Create Project Model object and Hibernate DB upgrade	↑	➡	Resolved
CRUC-101	Add 'Projects'	↑	➡	Resolved
CRUC-93	Crucible should preserve request params/	↑	➡	Resolved

	URLs through login redirects			
CRUC-90	More Administrator Options	↑		Resolved
CRUC-89	Add 'Default Reviewers'	↑		Resolved
CRUC-88	Review Groups	↑		Resolved
CRUC-86	add "allow anyone" as a per-project default	↑		Resolved
CRUC-80	Should be able to create a review when there are no configured repositories	↑		Resolved
CRUC-78	merge cru/fe src and content trees	↑		Resolved
CRUC-73	Generate HEAD review from directory	↑		Resolved
CRUC-68	Dragging to deselect source lines no longer works	↑		Resolved
CRUC-65	beta plugin api	↑		Resolved
CRUC-56	Self-registration	↑		Resolved
CRUC-43	Webservice API for Reviews	↑		Resolved
CRUC-95	Change Diff Buttons Losing State	↓		Resolved
CRUC-94	Show Existing Reviews In Fisheye	↓		Resolved
CRUC-61	Author should be able to "complete"	↓		Resolved
CRUC-36	Should my review status go back to incomplete if I start adding comments?	↓		Resolved
CRUC-115	Accept Patch	↓		Resolved

	Review From Clipboard
--	---

Crucible 1.2 Upgrade Guide

This page last changed on Dec 03, 2007 by rosie@atlassian.com.

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 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 Upgrade Guide

This page last changed on Oct 11, 2007 by [smaddox](#).

- [Upgrading from FishEye to Crucible](#)
- [Upgrading to a New Version of Crucible](#)

Upgrading from FishEye to Crucible

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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](#).

Before you Start

We strongly recommend you make a backup of your data before following the steps below. Refer to the FishEye documentation on [making a backup](#).

Upgrade Procedure

Follow method 1 below if you have FishEye configured to use a `FISHEYE_INST` directory. Follow method 2 if you are not using a `FISHEYE_INST` directory.

 Read more about the `FISHEYE_INST` [environment variable](#).

Method 1 - Using a FISHEYE_INST Directory

1. [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/`.
2. Leave your `FISHEYE_INST` environment variable set to its existing location.
3. Start Crucible from the new installation by running `NEW_FISHEYE/bin/run.sh`. (Use `run.bat` on Windows).
4. Follow the initial configuration steps outlined below.

Method 2 - Without a FISHEYE_INST Directory

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 (or move) the `/OLD_FISHEYE/var` directory to `/NEW_FISHEYE/var`.
7. If you have a Cenqua-issued FishEye license, copy `/OLD_FISHEYE/fisheye.license` to `/NEW_FISHEYE/`. (Atlassian-issued licenses are included within `config.xml`.)
8. Follow any version-specific instructions found in the [Release Notes](#).
9. Follow the initial configuration steps outlined below.

Initial Configuration

1. You can access Crucible immediately by going to <http://HOSTNAME:8060/> in a browser.
2. The first time you run Crucible, you will be asked for a Crucible license key. You can view your license key [here](#).
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](#).

Upgrading to a New Version of Crucible

This page last changed on Oct 15, 2007 by [smaddox](#).

 Read about how your [Crucible installation works with FishEye](#).

Before you Start

- Before upgrading you should always read the [Release Notes, Changelog and Upgrade Guides](#) 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

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

1. Extract the new Crucible version to a directory.
2. Leave your `FISHEYE_INST` environment variable set to its existing location.
3. Start Crucible from the new installation.
4. Follow any version-specific instructions found in the [Release Notes](#).

 Read more about the `FISHEYE_INST` [environment variable](#).

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](#).

Crucible Knowledge Base

This page last changed on Oct 15, 2007 by [smaddox](#).



Most setup issues are likely to be related to the FishEye component of Crucible. Refer to the FishEye documentation:

- [FishEye documentation](#)
- [FishEye knowledge base](#)

Do you still have a question, or need help with Crucible? Please [create a support request](#).

[Crucible Knowledge Base](#)

Answers to commonly raised questions about configuring and using Crucible.

- [Troubleshooting](#)

Troubleshooting

This page last changed on Oct 12, 2007 by [smaddox](#).

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](#).

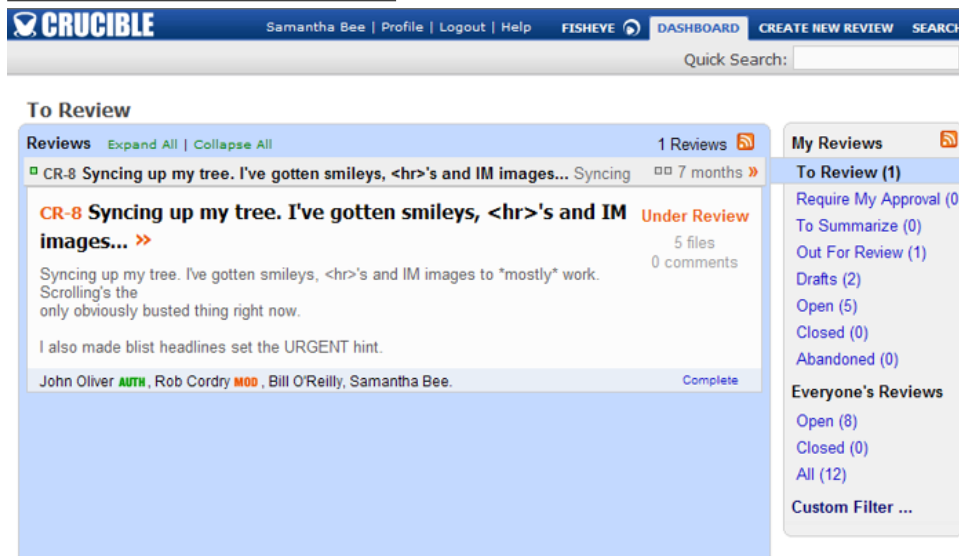
This page last changed on Dec 04, 2007 by rosie@atlassian.com.

- [1. Using the Dashboard](#)
- [2. Requesting and Conducting a Review](#)
 - [2.1 Creating a Review](#)
 - [2.1.1 Creating a Review from FishEye](#)
 - [2.1.2 Creating a Review within Crucible](#)
 - [2.1.3 Creating a Review from JIRA](#)
 - [2.1.4 Creating a Review from a URL](#)
 - [2.1.5 Selecting the Files for the Review](#)
 - [2.1.6 Adding Reviewers](#)
 - [2.1.7 Creating a Patch Review](#)
 - [2.2 Issuing a Review](#)
 - [2.3 Performing the Review](#)
 - [2.3.1 Adding Comments](#)
 - [2.3.2 Flagging Defects](#)
 - [2.3.3 Completing your Review](#)
 - [2.4 Summarising and Closing the Review](#)
- [3. Changing your User Profile](#)

1. Using the Dashboard

This page last changed on Nov 14, 2007 by rosie@atlassian.com.

The Dashboard is the home screen for Crucible and allows users to manage their reviews.
Screenshot: Crucible Dashboard



Managing Reviews

To find a particular review, use the side panel or the search box (Quick Search), as described below.

Side Panel

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

To Review	Reviews that the user needs to review.
Require My Approval	The user has been assigned the role of moderator for these reviews and needs to approve them.
To Summarize	The user has been assigned the role of moderator for these reviews and needs to summarise and close them.
Out For Review	These reviews are currently in progress but do not require any action by the user.
Drafts	These are reviews created by the user that have not yet been moved to the 'Approval' or the 'Require Approval' states .
Open	These reviews have been created by the user and are at a state other than 'Closed'.
Closed	These are reviews that the user has been involved in and are now closed.

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


Open	All open reviews.
Closed	All closed reviews.
All	All reviews.

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

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

Review State: Below is a description of the checkboxes available under the heading 'Review State'. You can use the checkboxes with the above filters or on their own.

Draft	Reviews that are still in 'Review' state .
Pending Approval	Reviews that have been moved out of 'Draft' state and are now waiting for the moderator to approve .
Under Review	The review is now 'Under Review'.
Summarize	The review is now in 'Summarize' state.
Closed	Reviews that are now 'Closed'.
Abandoned	Any reviews that have been abandoned.
Rejected	Any reviews that a moderator has rejected.
Review needs fixing	A review will match this filter if the review enters into an undefined state because something went wrong with storing the review state . A moderator can use this filter to find the review and then change the state to something sensible.

 It is highly unlikely that any reviews will be found under the filter 'Review needs fixing'. But if reviews start appearing there regularly, please raise a [support request](#).

Search Box (Quick Search)

Quick Search is the search box in the top right-hand corner of the Dashboard.

You can find a review by searching for:

- Review IDs - must be in the format CR-xx
- Contents of Title
- Contents of Statement of Objective
- Contents of a comment

Changing your User Profile

Users can change Crucible settings such as passwords, notifications and display settings.

To view your user profile, log into Crucible, and click the 'Profile' link at the top of the screen.

Read [more information](#).

Subscribing to an RSS Feed

The Dashboard provides two RSS feed icons. Click an icon to set up an RSS feed containing reviews as follows:

- Icon on the right-hand panel - reviews that you have had interaction with.
- Icon on the left-hand panel - all reviews.

2. Requesting and Conducting a Review

This page last changed on Oct 10, 2007 by [smaddox](#).

This guide explains in detail how to create a review and manage the review through its lifecycle states until completion.

For an explanation of the different roles, see the Terminology guide

- [2.1 Creating a Review](#)
- [2.2 Issuing a Review](#)
- [2.3 Performing the Review](#)
- [2.4 Summarising and Closing the Review](#)

2.1 Creating a Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

The first step is to create the review. There are two ways to create a review:

- [Creating a review from FishEye](#)
Using FishEye it is easy to create a review for a single changeset by clicking on the Crucible icon in the FishEye changelog view.
- [Creating a review within Crucible](#)
For reviews involving multiple changesets or an arbitrary set of files, you can create a review from the Crucible dashboard.

Click the links below for more information:

- [2.1.1 Creating a Review from FishEye](#)
- [2.1.2 Creating a Review within Crucible](#)
- [2.1.3 Creating a Review from JIRA](#)
- [2.1.4 Creating a Review from a URL](#)
- [2.1.5 Selecting the Files for the Review](#)
- [2.1.6 Adding Reviewers](#)
- [2.1.7 Creating a Patch Review](#)



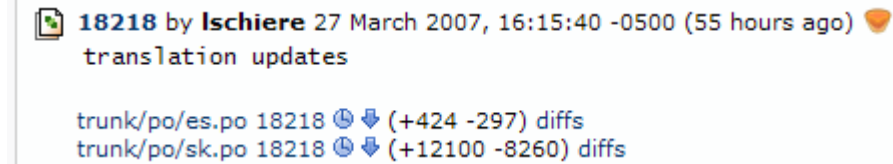
Note that only people with the ['Create' permission](#) can create a review.

2.1.1 Creating a Review from FishEye

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

To create a review from within FishEye, click the small Crucible icon next to the required changeset, and select the relevant [project](#) (if more than one project exists) in which to create your review.

Screenshot: Adding a Review in FishEye



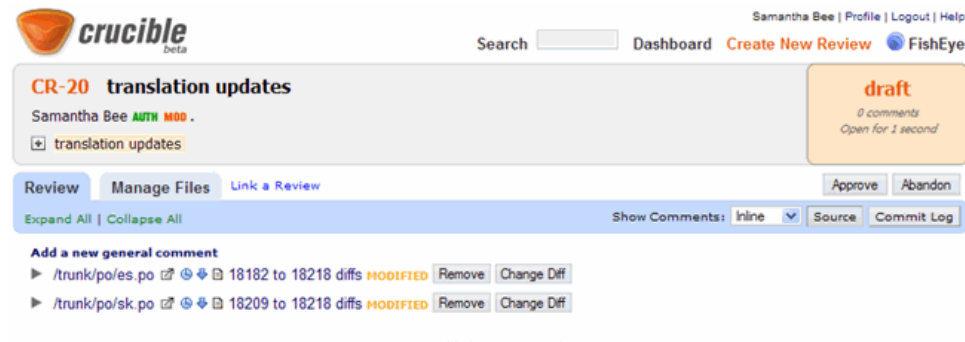
When you click the icon a new draft review will be created, including the following information:

- 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 and [Statement of Objective](#).

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

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

Screenshot: Review screen in Crucible



The next step is to [add reviewers](#).

2.1.2 Creating a Review within Crucible

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

Within Crucible, you can create a review by clicking the Create New Review link at the top right of every Crucible screen, and selecting the relevant [project](#) (if more than one project exists) in which to create your review. The Edit details screen appears, as shown below.

Screenshot: Adding a review in Crucible

CRUCIBLE Samantha Bee | Profile | Logout | Help FISHEYE DASHBOARD **CREATE NEW REVIEW** SEARCH

Quick Search:

Title:

Moderator:

Author:

Statement of Objectives:

Reviewers: ☐ Allow anyone to join
☐ Bill O'Reilly ☐ John Oliver
☐ Jon Stewart ☐ Rob Cordry
☐ Samantha Bee ☐ Stephen Colbert

draft
0 comments
Open for 1 second

Review **Manage Files** [Link a Review](#)

Repository: Author: Branch: Tag:

Changesets **Files** **Search** **Patch**

[Remove all revisions](#) | [« Earlier Changesets](#) | [later »](#)

Files in gaim:/

- ☐ **18236** created by sadrul on 06 April 2007, 23:06:48 -0500 (6 months ago)
Allow --with-ncurses-headers= configure option from here too
☐ /trunk/console/libgnt/configure.ac 18236 (+10 -3) diffs
- ☐ **18235** created by sadrul on 04 April 2007, 15:22:16 -0500 (6 months ago)
"elb: g_build_path is Glib 2.8"
☐ /trunk/console/libgnt/gntfilesel.c 18235 (+106 -1) diffs

Next steps:

- Use the 'Manage Files' tab to select the files to be reviewed.
- Use the 'Changesets', 'Files' or 'Search' tab to find the files to be added.

Read [more details](#) about selecting files for review.

2.1.3 Creating a Review from JIRA

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

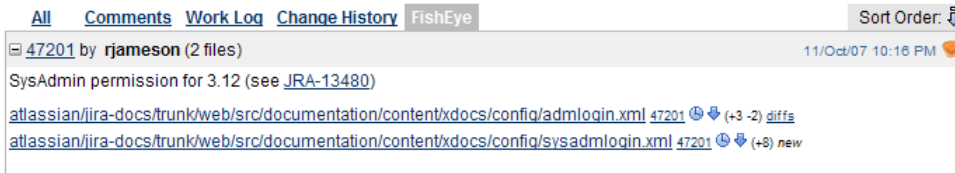
To create a review from within [JIRA](#), click the small Crucible icon



next to the required

changeset on the 'FishEye' tab.

Screenshot: Adding a Review from within JIRA



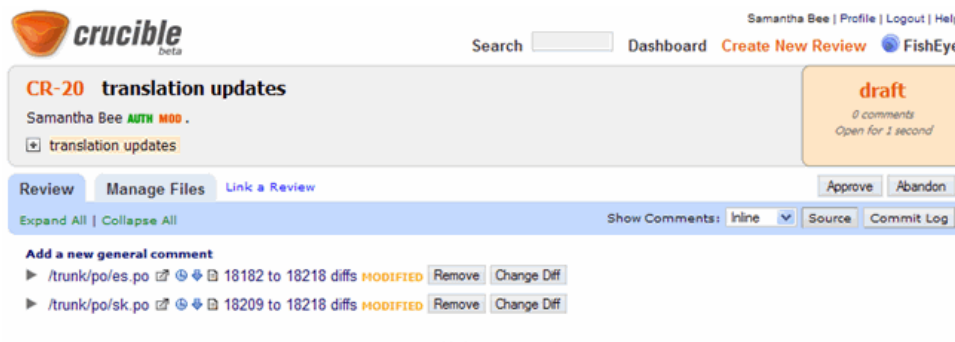
When you click the icon, you will be prompted to select the relevant [project](#) (if more than one project exists) in which to create your review. A new draft review will then be created, including the following information:

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

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

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

Screenshot: Review screen in Crucible



The next step is to [add reviewers](#).

2.1.4 Creating a Review from a URL

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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

The format of your URL is as follows:

`http://localhost:8060/cru/create?csid=%2F%2F&repo=a%2F1234&csid=%2F%2Frepob%2F7583`

The parameters are as follows:

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

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

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

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

The next step is to [add reviewers](#).

2.1.5 Selecting the Files for the Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

The 'Manage Files' tab is displayed when you are [creating a review](#). This tab allows you to select and modify which files make up the review.

The 'Manage Files' tab has the following sub-tabs:

- [Changesets](#)
- [Files](#)
- [Search](#)
- [Patch](#)

Constraints on Manage Files Tab

Before using the 'Changesets' or 'Files' tabs, you can narrow the files down by specifying constraints, as listed along the top of the 'Manage Files' tab:

Repository	This is a list of the repositories that contain the files that can be reviewed. If the repository you require is not in the list then it has not been added to FishEye. Please contact your Crucible/ FishEye administrator.
Author	This contains a list of all the authors who have made changes within the repository. When creating a review, this will default if possible to the username of the user authoring this review and will therefore show their changesets.
Branch	This will only show files and recent changes on that branch from the repository set above.
Tag	This will only show files and recent changes tagged.



To create a review that includes a number of revisions for a particular file, always choose the oldest revision first and then tick each revision of the file up to the most recent one. Only one version of the file will be reviewed but it will contain the diffs from the revisions chosen.

Changeset Tab

Screenshot: Changesets tab



By default, Crucible presents a list of the author's changesets in reverse chronological order. You can see other changesets by changing the constraints described [above](#).

Click the checkbox next to the changeset ID to add the entire changeset, or the checkboxes next to file names to add or remove individual files. Click 'Remove all revisions' to remove them.

Understanding Changesets in Detail

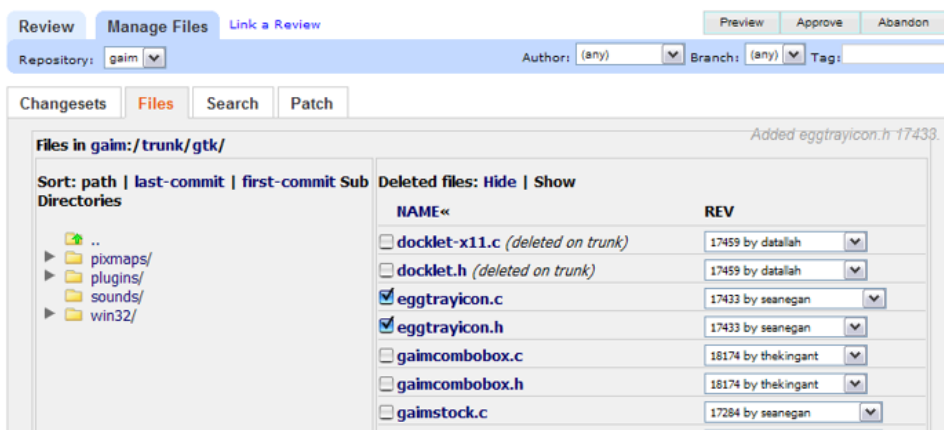
To help decide what files are to be placed under review, the [moderator](#) can click the icons next to the files to gain further information about them before they go out for review:

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

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

Files Tab

Screenshot: Files tab



Use the Files tab to find specific files

If you want to have specific files reviewed, rather than a changeset, it may be easier to find the files via the 'Files' tab.

To find a file, browse the folders by clicking the relevant folder. The folders by default are sorted by path name but can be sorted by last-commit or first-commit

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.

To choose a file for reviewing, click the checkbox to the left of the filename and if required the revision number.

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

Search Tab

Screenshot: Search tab

The screenshot shows the FishEye Search tab interface. At the top, there are tabs for 'Review', 'Manage Files', and 'Link a Review'. Below these are buttons for 'Preview', 'Approve', and 'Abandon'. A navigation bar includes 'Repository: gaim', 'Author: (any)', 'Branch: (any)', and 'Tag:'. Below the navigation bar are tabs for 'Changesets', 'Files', 'Search' (which is active), and 'Patch'. The main content area is titled 'gaim/trunk/gtk/'. It is divided into two columns: 'Search Criteria' and 'Search Results'. The 'Search Criteria' column contains a 'Search' button, checkboxes for 'Search all directories' and 'Only most recent versions', input fields for 'Comment:', 'Contents:', and 'File name:', a dropdown for 'Author(s)' with a list of names (aaronsheldon, amc_grim, chipx86, datallah, deryni9, eblanton, evands, faceprint), input fields for 'Branch:', 'Tag:', 'After:', and 'Before:', a date format '(YYYY-MM-DD ?)', radio buttons for 'Changeset', 'Revision', and 'Directory', and a 'Search' button. The 'Search Results' column is currently empty and contains the text 'Search your past.' and 'Fisheye has comprehensive search capabilities. Enter your search criteria to the left. (Note: content search currently searches only the HEAD of each file.)'

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

Choose the relevant search filters and then click the 'Search' button. If the simple filters are not enough, click 'Switch to Advanced Search'. You can then create an EyeQL query can be created to produce the results. Read about [EyeQL queries](#) in the FishEye documentation.

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

Read the FishEye documentation for more information about the [searching your repository](#).

Patch Tab

Read about [creating a patch review](#).

2.1.6 Adding Reviewers

This page last changed on Dec 04, 2007 by [rosie@atlassian.com](#).

The [moderator](#) adds reviewers during [review creation](#).

Before the review can be issued to reviewers, the [moderator](#) must decide who can review it.

Screenshot: Adding reviewers

CRUCIBLE Samantha Bee | Profile | Logout | Help FISHEYE DASHBOARD [CREATE NEW REVIEW](#) [SEARCH](#)

Quick Search:

Title:

Moderator:
Samantha Bee

Author:
Samantha Bee

Statement of Objectives:

Reviewers: ☐ Allow anyone to join
☐ Bill O'Reilly ☐ John Oliver
☐ Jon Stewart ☐ Rob Cordry
☐ Samantha Bee ☐ Stephen Colbert

draft
0 comments
Open for 57 minutes

[Save](#) [Cancel](#) [Abandon](#)

[Review](#) [Manage Files](#) [Link a Review](#) [Approve](#) [Abandon](#)

[Expand All](#) | [Collapse All](#) Show Comments: [Inline](#) [Source](#) [Commit Log](#)

Add a new general comment

- [/trunk/gtk/eggtrayicon.c](#) 16855 to 17433 diffs **MOVED TO** [Remove](#) [Change Diff](#)
- [/trunk/gtk/eggtrayicon.h](#) 16855 to 17433 diffs **MOVED TO** [Remove](#) [Change Diff](#)

The list of 'Reviewers' appears next to the Author and Moderator fields on the review details screen, as shown above.

- If there are fewer than seven potential reviewers, they are shown with checkboxes next to each name, as shown above.
- If there are seven or more potential reviewers, the list appears as a text box. Start typing the reviewer's name and then press the tab key to select the reviewer.

Click the 'Save' button to continue creating the review, or click the 'Approve' button (bottom right-hand corner) to issue the review to the reviewers.

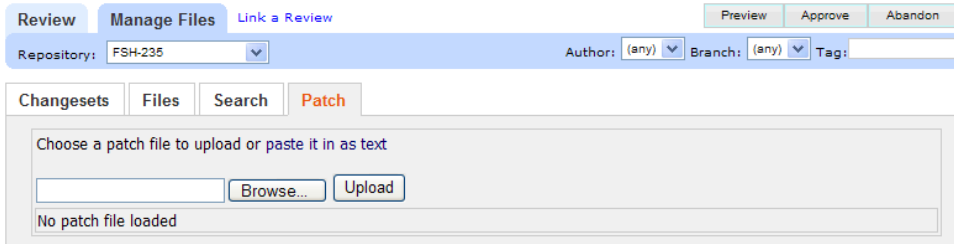
You can also decide to allow any user to add themselves as a reviewer in the review. To enable this option, put a check next to 'Allow anyone to join'. This feature is new in Crucible 1.1.2.

2.1.7 Creating a Patch Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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](#).

Screenshot: Patch tab

The screenshot shows the 'Patch' tab in the Crucible interface. At the top, there are tabs for 'Review', 'Manage Files', and 'Link a Review'. Below these are buttons for 'Preview', 'Approve', and 'Abandon'. A header bar contains a 'Repository' dropdown set to 'FSH-235', and 'Author', 'Branch', and 'Tag' dropdowns, all set to '(any)'. Below the header, there are tabs for 'Changesets', 'Files', 'Search', and 'Patch'. The 'Patch' tab is active, showing a text area with the prompt 'Choose a patch file to upload or paste it in as text'. Below the text area are 'Browse...' and 'Upload' buttons. At the bottom of the tab, it says 'No patch file loaded'.

CVS Patch Creation

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

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

SVN Patch Creation

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

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



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

Perforce Patch Creation

To create a patch in Perforce, you must ensure you have set [P4DIFF](#) to point to a GNU-compatible diff program.

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

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

Then do a `p4 opened` to find added and deleted files. For added files, append them individually to the patch using GNU diff:

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

You can follow a similar procedure with deleted files using `p4 print` to extract the previous version of the file.



A future version of Crucible will include helper tools to assist in this process.

2.2 Issuing a Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

Issuing a review in Crucible is known as approving the review.

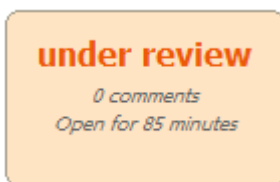
Once you have [selected the reviewers](#), the next stage is to notify the [reviewers](#) and the [author](#) (if different to the [moderator](#)) that they can start reviewing. The review has been in 'Draft' [state](#) until this point. Only the moderator has the permission to approve a review.

To issue the review:

- If you are 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.
- If you are the moderator of the review, click the 'Approve' button.

Once the review has been approved, the review [state](#) becomes:

Review state: under review



i Note that only people with the ['Approve' permission](#) can issue a review.

2.3 Performing the Review

This page last changed on Oct 15, 2007 by [smaddox](#).

Use the Crucible 'Dashboard' to manage your reviews. Read the overview on [managing your reviews](#).

Active reviews are listed on each [reviewer](#)'s dashboard under the default 'To Review' filter. For [authors](#) and [moderators](#), reviews are listed under 'Out for Review' until all reviewers indicate they are complete. Then the reviews move to the 'To Summarize' list of the moderator.



Your Open and Closed Reviews

All reviews that involve you in any role are listed under 'Open' or 'Closed' in the 'My Reviews' filters. 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 mailing is 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.



Deciding what needs to be reviewed

Crucible does not dictate how or what to review. It simply provides a mechanism to record comments. When asked to perform a review, you might like to read the 'Statement of Objective' by clicking the + sign under the review title to determine in more detail what the moderator requires you to review.

- [2.3.1 Adding Comments](#)
- [2.3.2 Flagging Defects](#)
- [2.3.3 Completing your Review](#)

2.3.1 Adding Comments

This page last changed on Dec 04, 2007 by [rosie@atlassian.com](#).

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

 Read about [flagging defects](#) too.

Screenshot: Comments



CR-5 Disable get_buddies_levels()

Jon Stewart **AUTH** **MOD**, Rob Cordry, Bill O'Reilly, Stephen Colbert.

I believe there's a bug in the get_buddies_levels() code that I don't have time to find. Temporarily disable

under review
4 comments
Open for 6 days

Review Complete

Expand All | Collapse All | Expand Commented Show Comments: Inline | Source | Commit Log

Bill O'Reilly [#permalink | 15 February 2007, 23:14] Reply

Over all a job well done. That's from the gut.

Add a new general comment

/trunk/libgaim/protocols/qq/buddy_list.c 17924 to 18179 diffs **MODIFIED**

/trunk/libgaim/protocols/qq/buddy_info.h 17924 to 18179 diffs **MODIFIED**

Add a revision comment (click on source to add inline comment) Show Full Source

```
93 93 void qq_process_get_info_reply(guint8 *buf, gint buf_len, GaimConnection *gc);
94 94 void qq_info_query_free(qq_data *qd);
95 95 void qq_send_packet_get_level(GaimConnection *gc, guint32 uid);
96 + /*
97 void qq_send_packet_get_buddies_levels(GaimConnection *gc);
98 + */
```

Rob Cordry [#permalink | (15 February 2007, 23:28)] Ranking: Minor Classification: Missing **DEFECT** Reply

If we're going to comment stuff out, I think a comment as to why would be warranted

Stephen Colbert [21 February 2007, 23:43] Edit Delete

Actually this could be deleted.

```
97 99 void qq_process_get_level_reply(guint8 *buf, gint buf_len, GaimConnection *gc);
98 100
99 101 #endif
```

/trunk/libgaim/protocols/qq/buddy_info.c 18179 **MODIFIED**

Add a revision comment (click on source to add inline comment)

```
1 /**
2  * #file buddy_info.c
3  *
```

- To add a comment that applies to the whole review, click the 'Add a new general comment' link.
- To add a comment to a revision/change, expand the specific source by clicking the toggle triangle then click 'Add a revision comment'.
- 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 reply to a comment, click the 'Reply' button on the right-hand top corner of any comment.

 Only people with the ['Comment' permission](#) can add comments.

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 remaining draft comments.

Screenshot: Draft comments



CR-5 Disable get_buddies_levels()

Jon Stewart **AUTH** **MOD**, Rob Cordry, Bill O'Reilly, Stephen Colbert.

I believe there's a bug in the get_buddies_levels() code that I don't have time to find. Temporarily disable it.

under review
4 comments
Open for 46 days

Warning - You have draft comments in this review

Rob Cordry [#permalink | 03 April 2007, 20:08] DRAFT Reply Edit Delete Post

I don't think this code is correct.

Please post any comments you want to keep, then press **Confirm Complete**

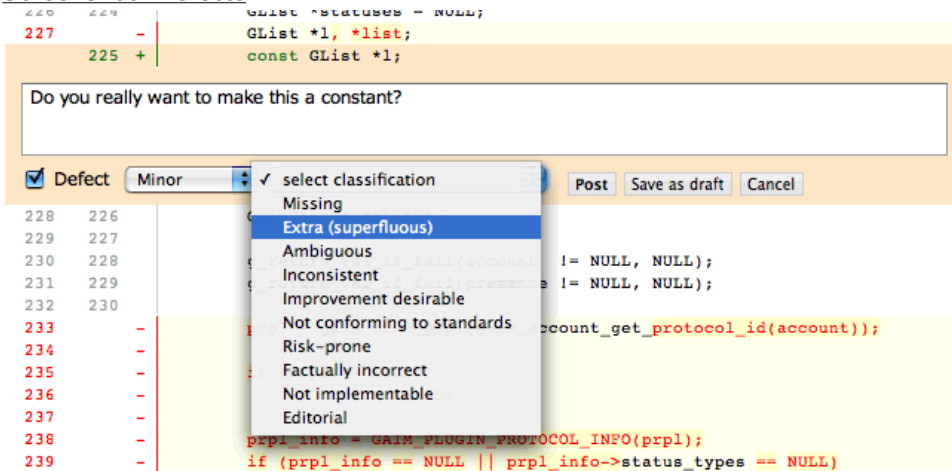
Confirm Complete Cancel (Return to Review)

2.3.2 Flagging Defects

This page last changed on Oct 14, 2007 by [smaddox](#).

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 administrator can [customise the defect classifications](#).

2.3.3 Completing your Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

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,

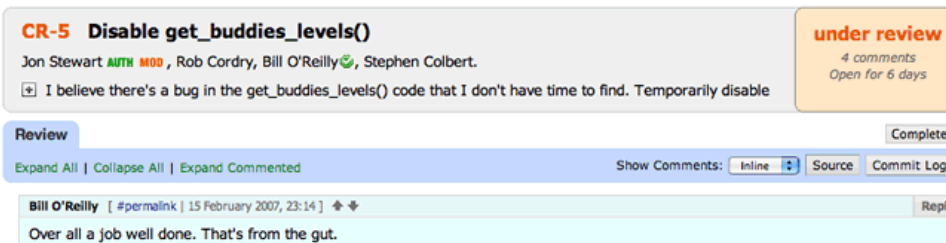
- either click the 'Complete' link at bottom right of the [Dashboard](#),
- or click the 'Complete' button at far right of the 'Review' screen.

i Only people with the ['Complete' permission](#) can complete a review.

Screenshot: The 'Complete' link on the Dashboard



Screenshot: The 'Complete' button on the Review screen



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



2.4 Summarising and Closing the Review

This page last changed on Dec 04, 2007 by rosie@atlassian.com.

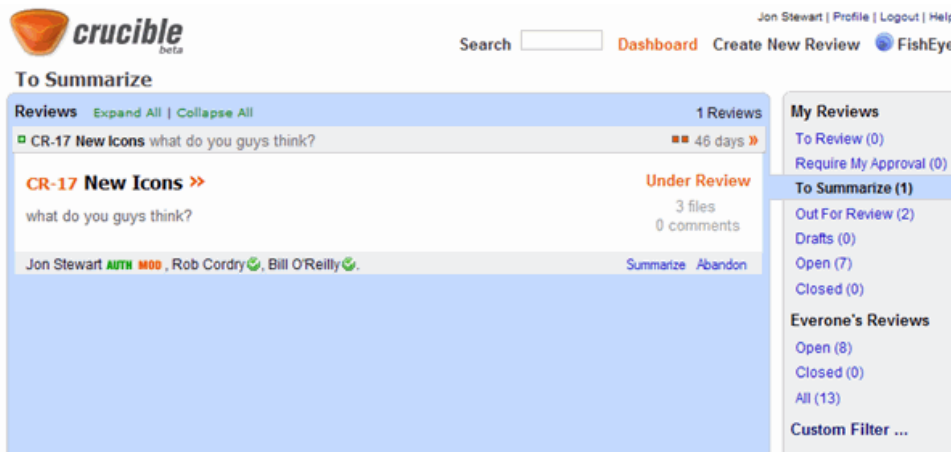
As the [moderator](#), you can choose to summarise a review at any time.

 Normally, we recommend that you wait for all reviewers to [complete their reviews](#).

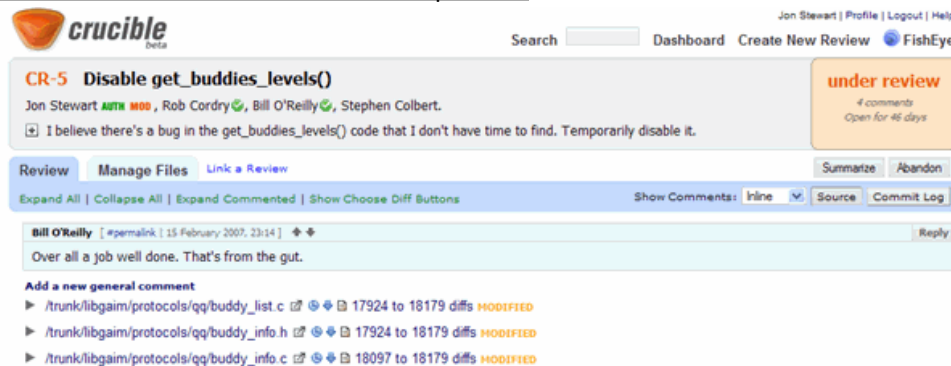
The reviews that the reviewers have completed will be in the 'To Summarize' menu on the [Dashboard](#). There are two ways to summarise a review:

- Either click the 'Summarize' link at bottom right of the review,
- Or preview the review, review the comments and then click on the 'Summarize' button.

Screenshot: 'Summarize' link on Review screen



Screenshot: 'Summarize' button on preview



The above review is not yet complete

We can see that Stephen Colbert 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 summarised 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: Review summary

crucible

Jon Stewart | Profile | Logout | Help

Search Dashboard Create New Review FishEye

Summarize the Review Outcomes (optional)

Close

CR-5 Disable get_buddies_levels()

Jon Stewart Rob Cordry Bill O'Reilly Stephen Colbert.

I believe there's a bug in the get_buddies_levels() code that I don't have time to find. Temporarily disable it.

summarize 4 comments Open for 46 days

Review Manage Files Link a Review Close Reopen Abandon

Expand All Collapse All Expand Commented Show Comments: Above Source Commit Log

Bill O'Reilly [15 February 2007, 23:14] ♦♦

Over all a job well done. That's from the gut.

Reply

Add a new general comment

▼ trunk/bugaim/protocols/qg/buddy_list.c @ 17924 to 18179 diffs MODIFIED

Bill O'Reilly [15 February 2007, 23:14] ♦♦

Ranking: Newer Classification: Not conforming to standards DEFECT

Please, no wars about brace styles. adopt the project coding standard

Jon Stewart [15 February 2007, 23:23]

that "is" the project standard Bill :-)

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.

Note that you need the '[Summarise](#)', '[Close](#)' or '[Re-Open](#)' permission to summarise, close or re-open a review.

3. Changing your User Profile

This page last changed on Oct 15, 2007 by [smaddox](#).

Users can change Crucible settings such as passwords, notifications and display settings.

To view your user profile, log into Crucible and click the 'Profile' link at the top of the screen.

Screenshot: Profile



FishEye > Profile

Display settings | Email | Change Password | Author mapping | Watches | Reviews

Send Notifications on

State change: ☒ Yes ☐ No

Comment added: ☒ Yes ☐ No

Participant finished: ☒ Yes ☐ No

General message: ☒ Yes ☐ No

My actions: ☐ Yes ☒ No

(emails sent to sam@nosspam.com)

save

Refer to the [FishEye documentation](#) for information about the tabs labelled 'Display settings', 'Email', 'Change Password' 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 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.

Glossary

This page last changed on Oct 10, 2007 by [smaddox](#).

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](#)

[reviewer](#)

[role](#)

[state](#)

[statement of objective](#)

[user](#)

approve

This page last changed on Oct 14, 2007 by [smaddox](#).

[Issuing a review to the reviewers](#) is known as approving the review.

author

This page last changed on Oct 10, 2007 by [smaddox](#).

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 change under review.

code review

This page last changed on Oct 15, 2007 by [smaddox](#).

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

This page last changed on Oct 15, 2007 by [smaddox](#).

A comment is a short textual note that is linked to a review, revision/diff, source line, or to another comment.

See [2.3.1 Adding Comments](#).

creator

This page last changed on Oct 15, 2007 by [smaddox](#).

The creator is the person who [creates the review](#). In most cases this role will be undertaken by the [moderator](#).

defect

This page last changed on Oct 15, 2007 by [smaddox](#).

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

This page last changed on Oct 15, 2007 by [smaddox](#).

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

This page last changed on Dec 02, 2007 by rosie@atlassian.com.

Crucible uses the terms [creator](#), [author](#), [moderator](#), and [reviewer](#) to describe the roles of review participants.

permission

This page last changed on Dec 02, 2007 by rosie@atlassian.com.

A permission is the ability to perform a particular action in Crucible, e.g. 'Create Review'. Permissions are assigned to particular users, groups or [review roles](#) by means of [permission schemes](#).

The following permissions are available:

Permission	Description	Default Assignees
'Edit'	Ability to edit a review's details and change the set of revisions being reviewed.	' Moderator ' ' Creator '
'View'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users ' Moderator ' ' Creator ' ' Author ' ' Reviewer '
'Abandon'	Ability to abandon (i.e. cancel) a review.	' Moderator ' ' Creator '
'Re-Open'	Ability to re-open a closed or abandoned review.	' Moderator ' ' Creator '
'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.	' Moderator ' ' Creator ' ' Author ' ' Reviewer '
'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 '
'Summarise'	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	' Moderator '
'Create'	Ability to create a review.	All logged-in users

'Recover'	Ability to resurrect an abandoned (i.e. cancelled) review.	' Moderator ' ' Creator '
-----------	--	--

permission scheme

This page last changed on Dec 02, 2007 by rosie@atlassian.com.

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](#), i.e.:
 - ['moderator'](#)
 - ['author'](#)
 - ['reviewer'](#)
 - ['creator'](#)

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

This page last changed on Dec 03, 2007 by rosie@atlassian.com.

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](#).

reviewer

This page last changed on Oct 15, 2007 by [smaddox](#).

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

This page last changed on Dec 02, 2007 by rosie@atlassian.com.


See [participant](#).

state

This page last changed on Oct 15, 2007 by [smaddox](#).

A Crucible review moves through the following states in the following sequence:

Draft	See Creating a Review .
Require Approval	Relevant only when the moderator is not the creator . See Issuing a Review .
Under Review	See Issuing a Review and Reviewing the Code .
Summarize	See Summarising and Closing the Review .
Closed	See Summarising and Closing the Review .

 Reviews can be re-opened, i.e. moved from Summarize or Closed back to Under Review.

A review may also be in the following states:

Abandoned	This happens when a review is deleted.
Rejected	Any reviews that a moderator has rejected.

statement of objective

This page last changed on Oct 10, 2007 by [smaddox](#).

A statement of objective is an optional text description of the review and any specific areas the [reviewers](#) should focus on.

user

This page last changed on Oct 10, 2007 by [smaddox](#).

A user is a person using Crucible.