


## Space Details

Key:	CRUCIBLE
Name:	Crucible 1.6
Description:	Latest documentation for Crucible: do your code reviews, your way
Creator (Creation Date):	(Oct 04, 2007)
Last Modifier (Mod. Date):	edawson (Sep 23, 2008)

### Available Pages

- [.bookmarks](#)
- [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](#)
      - [Setting Crucible to Store all Revisions](#)
    - [Crucible and FishEye](#)
    - [Customising Email Notifications](#)
      - [Freemarker Data Model for Email Templates](#)
    - [Customising the Defect Classifications](#)
    - [Customising the Welcome Message](#)
    - [Deleting a Project](#)
    - [Trusted Applications](#)
  - [Crucible Development Hub](#)
    - [Crucible Plugins](#)
      - [Crucible Reporting plugin](#)
    - [Crucible REST API](#)
      - [Data Types](#)
      - [Project Service](#)
      - [Repository Service](#)
      - [Review Service](#)
    - [Crucible REST API Usage Example](#)
  - [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 Changelog
  - Crucible 1.2 Upgrade Guide
- Crucible 1.5 Release Notes
  - Crucible 1.5 Changelog
- Crucible 1.6 Release Notes
- Crucible Release Summary
- Crucible Upgrade Guide
  - Upgrading from FishEye to Crucible
  - Upgrading to a New Version of Crucible
- Crucible Knowledge Base
  - Do I need a FishEye licence to run Crucible?
  - Troubleshooting
- Crucible User's Guide
  - 0. Getting Started with Crucible
  - 1. Using the Dashboard
    - Using the Project 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
        - Removing Reviewers From An Active Review
      - 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
    - 2.5 Deleting an Abandoned 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
- \_\_newreleaseCrucible
- TreeNavigation

## .bookmarks

---

This page last changed on Jan 17, 2008 by .



### Crucible 1.2 Bookmarks

This page is a container for all the bookmarks in this space. Do not delete or move it or you will lose all your bookmarks.

[Bookmarks in Crucible 1.2](#) | [Links for Crucible 1.2](#)



## The 15 most recent bookmarks in Crucible 1.6

There are no bookmarks to display.

## Crucible Documentation Home

This page last changed on Sep 23, 2008 by [edawson](#).

Getting Started with Crucible 1.6	Using/Administering Crucible 1.6
<a href="#">About Crucible</a> <a href="#">Feature Tour</a> <a href="#">Download Crucible</a> <a href="#">Installation Guide</a> <a href="#">Release Notes</a>   <a href="#">Upgrade Guide</a> <a href="#">Glossary</a>	<a href="#">Quick Start: Conducting a Review</a> <a href="#">Administrator's Guide</a> <a href="#">User's Guide</a> <a href="#">FishEye Documentation</a> <a href="#">Support</a>   <a href="#">Forums</a>   <a href="#">FAQ</a> <a href="#">Development Hub</a>
Previous Versions	PDF Format
<a href="#">Crucible 1.1 documentation</a> <a href="#">Crucible 1.0 documentation</a>	You can <a href="#">download the Crucible documentation</a> in PDF format.



### Recently Updated

by Edwin Dawson [Atlassian] (31 minutes ago)



[Crucible 1.6 Release Notes](#)

by Edwin Dawson [Atlassian] (44 minutes ago)



[\\_newreleaseCrucible](#)

by Edwin Dawson [Atlassian] (44 minutes ago)



[Crucible 1.6](#)

by Edwin Dawson [Atlassian] (45 minutes ago)



[Crucible Documentation Home](#)

by Edwin Dawson [Atlassian] (46 minutes ago)



[Creating a Permission Scheme](#)

by Edwin Dawson [Atlassian] (48 minutes ago)



[3. Configuring Crucible](#)

by Edwin Dawson [Atlassian] (51 minutes ago)



[Crucible REST API](#)

by Edwin Dawson [Atlassian] (52 minutes ago)



[Setting Crucible to Store all Revisions](#)

by Edwin Dawson [Atlassian] (53 minutes ago)



[Deleting a Project](#)

by Edwin Dawson [Atlassian] (55 minutes ago)



[2.1.7 Creating a Patch Review](#)

## 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](mailto: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](mailto: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 Email Notifications](#)
- [Customising the Defect Classifications](#)
- [Customising the Welcome Message](#)
- [Deleting a Project](#)
- [Trusted Applications](#)



## Backups and Review Data

---

This page last changed on Dec 18, 2007 by [edawson](#).

The processes for backing up and restoring Crucible data are identical to the [FishEye processes](#) — follow the instructions on [this page](#).

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 Sep 23, 2008 by [edawson](#).

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 <a href="#">set of revisions</a> being reviewed.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'View'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users ' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> ' ' <a href="#">Author</a> ' ' <a href="#">Reviewer</a> '
'Abandon'	Ability to abandon (i.e. cancel) a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Re-Open'	Ability to re-open a closed or abandoned review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Uncomplete'	Ability of a reviewer to change their individual review status from 'Complete' to 'Uncomplete'.	' <a href="#">Reviewer</a> '
'Reject'	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	' <a href="#">Moderator</a> '
'Complete'	Ability of a reviewer to change their individual review status to 'Complete'.	' <a href="#">Reviewer</a> '
'Comment'	Ability to add or remove a comment to or from a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> ' ' <a href="#">Author</a> ' ' <a href="#">Reviewer</a> '
'Approve'		' <a href="#">Moderator</a> '

	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).	' <a href="#">Creator</a> ' ' <a href="#">Author</a> '
'Close'	Ability to close a review once it has been summarised.	' <a href="#">Moderator</a> '
'Delete'	Ability to delete a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Summarise'	Ability to summarise a review. (Normally this would be done after all reviewers have completed their review.)	' <a href="#">Moderator</a> '
'Create'	Ability to create a review.	All logged-in users
'Recover'	Ability to resurrect an abandoned (i.e. cancelled) review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '

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).
-  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.

 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 <a href="#">edit</a>   <a href="#">copy</a>
Name: <input type="text" value="Top Secret"/>	<a href="#">create</a> <a href="#">cancel</a>

Screenshot 2: Edit Permission Scheme 'Top Secret'

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

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

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

## Associating a Permission Scheme with a Project

---

This page last changed on Dec 04, 2007 by [rosie@atlassian.com](mailto: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 Jan 15, 2008 by [edawson](#).


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 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'.)

Screenshot: 'Create Project'

**Admin Menu**
**Repository Settings**

- Repository List
- Repository Defaults

**Global Settings**

- Server Settings
- Security
- Users
- Groups
- ViewCVS URL Mappings
- Change Admin Password
- Customize Crucible Defect Classifications
- Projects
- Permission Schemes

**System**

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

**Edit Project**
**Identification**

Name: 

Key: 
**Repository**

Default Repository: 
**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:

## Setting Crucible to Store all Revisions

This page last changed on Sep 23, 2008 by [edawson](#).

When creating a project or editing a project's properties, you can set Crucible to save all revisions that are associated with a review to Crucible's database. This allows you to be able to view that file content whether or not the repository is online or accessible to Crucible. It also creates an enhanced audit trail should you require it, saving the review content regardless of whether or not it is deleted or lost from the repository.

**i** Note that the storage of revisions must be set per-project. Also, the storage only applies to reviews created after Revision Storage is enabled. This means that for existing projects, pre-existing reviews will not be stored unless you look at them again after Revision Storage is enabled.

### Enabling Revision Storage on a new project

To enable Revision Storing on a new project,

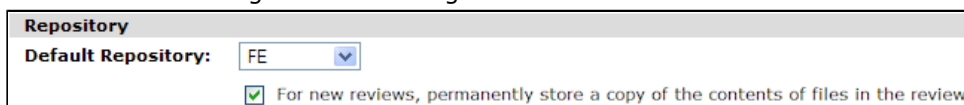
1. When creating a new project, you have the option to turn on revision storing on the 'Create New Project' page.
2. Under 'Repository', Click the checkbox labelled 'For new reviews, permanently store a copy of files in the review'.
3. Click 'Save' to finish.

### 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 'Repository', Click the checkbox labelled 'For new reviews, permanently store a copy of files in the review'.
4. Click 'Save' to finish.

Screenshot: Enabling Revision Storage



**Repository**

Default Repository: FE

☒ For new reviews, permanently store a copy of the contents of files in the review



## 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 Email Notifications

---

This page last changed on Apr 14, 2008 by [edawson](#).

Email notifications in Crucible can be customised to change their formatting, by editing template files. This page contains instructions for this process.

### Editing Crucible Email Templates

Template files for Crucible are stored in the `FISHEYE_HOME/templates/` folder.

For Crucible, the set of templates is for plain-text email only. Note that these templates do not support embedding full diffs into notifications. They are only for changing the appearance and order of certain content inside the messages.



If you edit the templates of an operational Crucible instance, you may disrupt notifications that are being sent at that time. To avoid this, shut Crucible down during template editing.

#### Editing the Subject Line

1. Open the 'crucible-notification-subject.ftl' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Type in your new text for the email subject, ensuring that all of the content is contained within line 1 of the template. 'crucible-notification-subject.ftl' is used as the subject template for all Crucible email notifications.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

#### Editing the Header

Header information will be included at the beginning of the email body text.

1. Open the 'crucible-notification-header.ftl' template file from `FISHEYE_HOME/templates/` in a text editor.
2. Add your new header content. 'crucible-notification-header.ftl' is used as the header template for all Crucible email notifications.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

#### Editing the Footer

Footer information will be included at the end of the email body text.

1. Open the 'crucible-notification-footer.ftl' template file from `FISHEYE_HOME/templates/` in a text editor. 'crucible-notification-footer.ftl' is used as the footer template for all Crucible email notifications.
2. Add your new footer content.
3. Save and close the file.
4. Restarting Crucible will activate the new templates.

After an edit, the change to the email template will take place immediately. No restart is required.



Try and avoid editing the live template file, as Crucible may try to use it while you are editing. This could have unpredictable results. Instead, back up the template file (it's wise to keep original versions of all these files), edit a copy you have made, then overwrite the 'live' template once you have finished.

### Advanced Editing of Crucible Email Templates

The email notification templates use the [Freemarker](#) format. Freemarker is a general templating engine enabling automated content.

If you are familiar with Freemarker, more advanced customisations can be made to the email notification templates. However, you make such adjustments at your own risk.

Note: In Crucible, email notifications are limited to plain-text format only.

### Crucible Email Template File List

The following template files for Crucible notification are stored in the `FISHEYE_HOME/templates/` folder.

Template filename	Purpose
crucible-notification-subject.ftl	Subject template
crucible-notification-header.ftl	Header template
crucible-notification-footer.ftl	Footer template
state-closed-notification.ftl	State Closed template
all-completed-notification.ftl	All Completed template
state-changed-notification.ftl	State Changed template
completed-notification.ftl	Completed template
general-notification.ftl	General notification template
uncompleted-notification.ftl	'Uncompleted' template
all-no-longer-completed-notification.ftl	All-No-Longer-Completed template
comment-notification.ftl	Comment template
reply-notification.ftl	Reply template
review-precis-plain.ftl	Precis template

See also [Customising FishEye Email Notifications](#).

## Freemarker Data Model for Email Templates

---

This page last changed on Mar 31, 2008 by [edawson](#).

### Customising Crucible email templates with Freemarker

See the [Freemarker documentation](#) for instructions on Freemarker syntax. Use the templates that ship with Crucible as a guide to the properties available on each object.

Specific email types will have extra data associated with them, and this data will be available in that particular template (but not in others).

### Example

The syntax to access the data-model, using the data model object 'link' as an example, place this code into the email at the desired position.

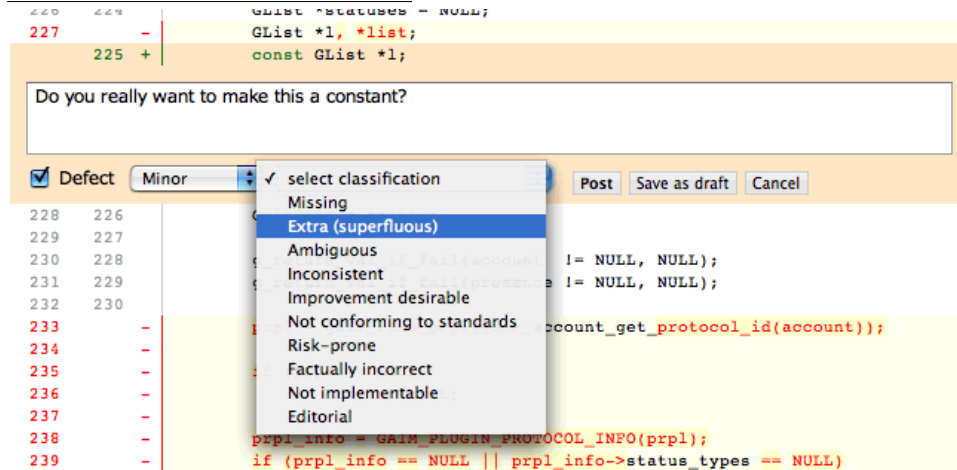
```
${notification.link}
```

## Customising the Defect Classifications

This page 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.

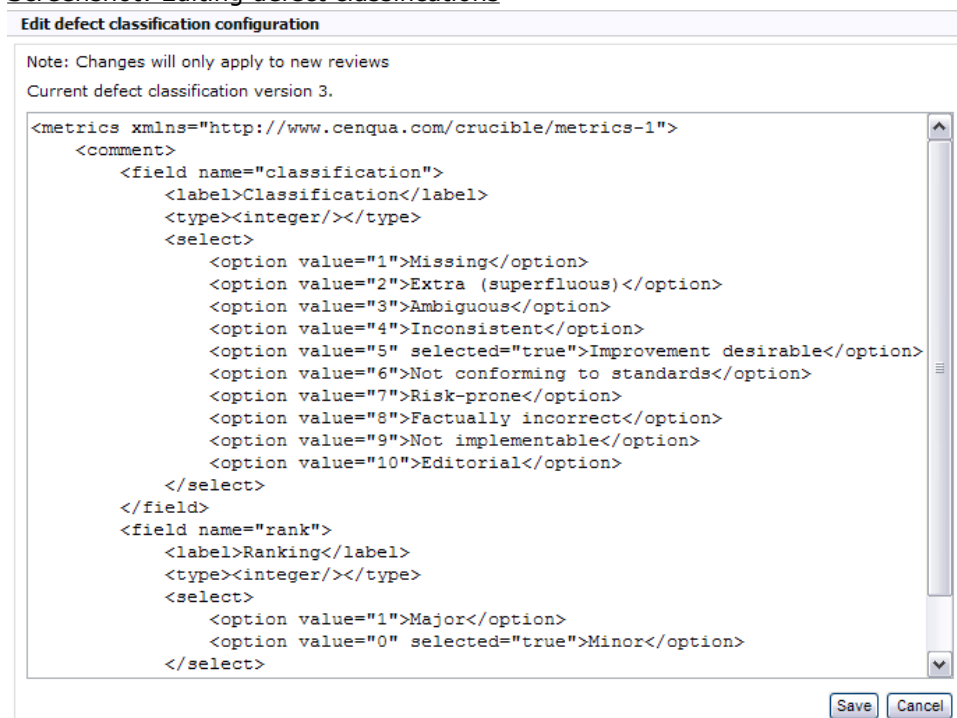
To change the default classifications:

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



## Customising the Welcome Message

---


This page last changed on Feb 21, 2008 by [edawson](#).

The welcome and support messages are [configured via FishEye's admin screen](#).

## Deleting a Project

This page last changed on Sep 23, 2008 by [edawson](#).

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 'Projects'.
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

Projects List						
Key	Name	Default Repository	Default Moderator	Default Reviewer Users	Default Reviewer Groups	
CR	Default Project					<a href="#">Edit</a>
CR-DEL	Delete	FE_tmp				<a href="#">Edit</a> <a href="#">Delete</a>
CR-CH	Confluence Hosted	CH				<a href="#">Edit</a> <a href="#">Delete</a>

## Trusted Applications

---

This page last changed on Jan 15, 2008 by [edawson](#).

Trusted Applications are supported in Crucible/FishEye from Crucible version 1.2.2.

For information on how to configure Trusted Applications in Crucible/FishEye, see the [FishEye documentation](#).



## Crucible Development Hub

---

This page last changed on Mar 24, 2008 by [edawson](#).

Documentation for Crucible plugin authors is contained in the [FishEye Development Hub](#). Visit that page for more information.

The Crucible [Javadocs](#) may also be of use:

- [Crucible Plugins](#)
  - [Crucible Reporting plugin](#) — A plugin for [Crucible](#) that facilitates the generation of a consolidated report for a specific review.
- [Crucible REST API](#)
  - [Data Types](#) — Definitions of data types used by the REST API.
  - [Project Service](#) — Provides access to the projects defined in a Crucible instance.
  - [Repository Service](#) — Provides information about the repositories configured in a Crucible instance.
  - [Review Service](#) — The Review Service allows you to list, examine, create and modify reviews.
- [Crucible REST API Usage Example](#)

## Crucible Plugins

---

This page last changed on Jun 19, 2008 by [jnolen](#).

- [Crucible Reporting plugin](#) — A plugin for [Crucible](#) that facilitates the generation of a consolidated report for a specific review.

## Crucible Reporting plugin


This page last changed on Sep 22, 2008 by [piaross@gmail.com](mailto:piaross@gmail.com).

Name	Crucible Reporting plugin
Version	1.0.0
Product Versions	1.5.x, 1.6.x (beta)
Author(s)	<a href="#">Ross Rowe</a>
Homepage	<a href="http://confluence.atlassian.com/display/COGEIST/Crucible+Reporting+plugin">http://confluence.atlassian.com/display/COGEIST/Crucible+Reporting+plugin</a>
Price	Free
License	<a href="#">BSD</a>
JavaDocs	<a href="#">crucible-export-plugin-javadoc.zip</a>
IssueTracking	N/A
Subversion URL	<a href="https://svn.atlassian.com/svn/public/contrib/crucible/crucible-export-plugin">https://svn.atlassian.com/svn/public/contrib/crucible/crucible-export-plugin</a> or browse via <a href="#">fisheye</a>
Download JAR	<a href="#">crucible-export-1.0.0.jar</a> (for Crucible 1.5.x) / <a href="#">crucible-export-1.6.0.jar</a> (for Crucible 1.6.x)
Download Source	<a href="#">crucible-export.zip</a>

## Description/Features

A plugin for [Crucible](#) that facilitates the generation of a consolidated report for a specific review. This is especially useful if you are required to keep hard copies of your code review (like in the case of an audit 😊)

## Usage

 This plugin requires Crucible 1.5 or 1.6

### Crucible 1.5 installation

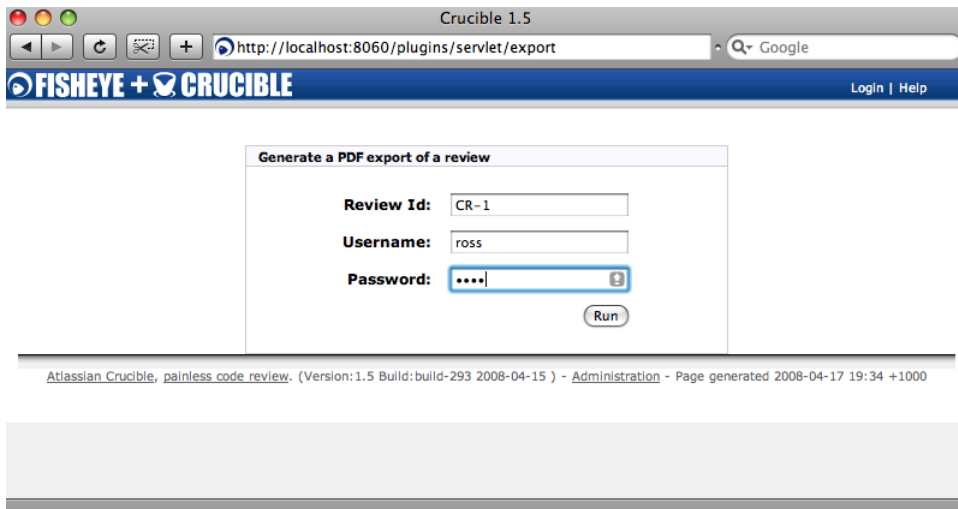
The plugin can be installed by copying the [crucible-export-1.0.0.jar](#) file into the CRUCIBLE\_HOME/var/plugins directory. You will also need to copy the [iText](#) jar file (available from <http://www.lowagie.com/iText/download.html>) into the CRUCIBLE\_HOME/lib directory.

### Crucible 1.6 installation

The plugin can be installed by copying the [crucible-export-1.6.0.jar](#) file into the CRUCIBLE\_HOME/var/plugins/user directory.

### Running the plugin

As Crucible does not currently have a mechanism to include user interface components via it's plugin api, the export mechanism can be opened by visiting <http://YourCrucibleHost/plugins/servlet/export>. From this page, the user must enter their username, password and the review id they wish to export.



Once the details are entered and the 'Run' button is clicked, a PDF including the Crucible Review details is generated. This report includes the summary information of the review, as well as any general and specific file comments.

## Version History

### Crucible 1.6 support

Version	Date	Description
1.6.0	22 Sep 2008	Updated plugin to support Crucible 1.6.0 beta

### Crucible 1.5 support

Version	Date	Description
1.0.0	9 May 2008	Updated plugin to support Crucible 1.5.1
0.0.4	4 May 2008	Added i18n support
0.0.3	29 April 2008	Updated unit tests
0.0.2	21 April 2008	Updated plugin to support Crucible 1.5
0.0.1	23 Mar 2008	Initial plugin version

## Screenshots

Error trying to draw image gallery org.apache.velocity.exception.MethodInvocationException: Invocation of method 'requireResource' in class \$Proxy40 threw exception java.lang.NullPointerException @ /com/atlassian/confluence/plugins/macros/advanced/gallery-default.vm[1,21]

## Crucible REST API

---

This page last changed on Sep 23, 2008 by [edawson](#).

These pages contain information relating to the REST API for Crucible.

A list of available services and a detailed example page are currently documented.

Crucible REST API documentation:

- [Data Types](#) — Definitions of data types used by the REST API.
- [Project Service](#) — Provides access to the projects defined in a Crucible instance.
- [Repository Service](#) — Provides information about the repositories configured in a Crucible instance.
- [Review Service](#) — The Review Service allows you to list, examine, create and modify reviews.

## Data Types

This page last changed on Sep 15, 2008 by [tom@atlassian.com](mailto:tom@atlassian.com).

Definitions of data types used by the REST API.

Name	Description
ReviewData	<pre>&lt;reviewData&gt;   &lt;allowReviewersToJoin&gt;false&lt;/ allowReviewersToJoin&gt;   &lt;author&gt;     &lt;displayName&gt;Matt Quail&lt;/displayName&gt;     &lt;userName&gt;matt&lt;/userName&gt;   &lt;/author&gt;   &lt;createDate&gt;2008-08-25T12:38:14.603+1000&lt;/ createDate&gt;   &lt;creator&gt;     &lt;displayName&gt;Matt Quail&lt;/displayName&gt;     &lt;userName&gt;matt&lt;/userName&gt;   &lt;/creator&gt;   &lt;description&gt;Review things and stuff&lt;/ description&gt;   &lt;metricsVersion&gt;1&lt;/metricsVersion&gt;   &lt;moderator&gt;     &lt;displayName&gt;Matt Quail&lt;/displayName&gt;     &lt;userName&gt;matt&lt;/userName&gt;   &lt;/moderator&gt;   &lt;name&gt;Test review 1&lt;/name&gt;&lt;permaId&gt; &lt;id&gt;CR-1&lt;/id&gt;&lt;/permaId&gt;   &lt;projectKey&gt;CR&lt;/projectKey&gt;   &lt;state&gt;Review&lt;/state&gt; &lt;/reviewData&gt;</pre>
DetailedReviewData	<p>Note that the <code>reviewItems</code> element is empty when multiple reviews are retrieved via REST. To include the <code>reviewItems</code> in a <code>detailedReviewData</code> structure you must retrieve a single review via the URL <code>/rest-service/reviews-v1/&lt;review id&gt;/details</code>.</p> <pre>&lt;detailedReviewData&gt;   &lt;allowReviewersToJoin&gt;false&lt;/ allowReviewersToJoin&gt;   &lt;author&gt;     &lt;displayName&gt;Matt Quail&lt;/ displayName&gt;     &lt;userName&gt;matt&lt;/userName&gt;   &lt;/author&gt;    &lt;createDate&gt;2008-09-16T10:50:26.862+1000&lt;/ createDate&gt;   &lt;creator&gt;     &lt;displayName&gt;Matt Quail&lt;/ displayName&gt;     &lt;userName&gt;matt&lt;/userName&gt;   &lt;/creator&gt;   &lt;description/&gt;</pre>

```

        <metricsVersion>1</metricsVersion>
        <moderator>
            <displayName>Matt Quail</
displayName>
            <userName>matt</userName>
        </moderator>
        <name/>
        <permaId>
            <id>CR-1</id>
        </permaId>
        <projectKey>CR</projectKey>
        <state>Draft</state>
        <actions>
            <actionData>

<name>action:abandonReview</name>
            </actionData>
            <actionData>

<name>action:closeReview</name>
            </actionData>
            <actionData>

<name>action:submitReview</name>
            </actionData>
            <actionData>

<name>action:reopenReview</name>
            </actionData>
            <actionData>

<name>action:summarizeReview</name>
            </actionData>
            <actionData>

<name>action:rejectReview</name>
            </actionData>
            <actionData>

<name>action:deleteReview</name>
            </actionData>
            <actionData>

<name>action:approveReview</name>
            </actionData>
            <actionData>

<name>action:modifyReviewFiles</name>
            </actionData>
            <actionData>

<name>action:viewReview</name>
            </actionData>
            <actionData>

<name>action:commentOnReview</name>
            </actionData>
            <actionData>

<name>action:recoverReview</name>
            </actionData>
            <actionData>

<name>action:createReview</name>
            </actionData>
        </actions>

```

```

        <generalComments/>
        <reviewItems>
            <reviewItem>
                <permId>
                    <id>CFR-1</id>
                </permId>
                <authorName>admin</authorName>
                <commitDate>2008-08-27T10:19:17.000+1000</commitDate>
                <commitType>Modified</commitType>
                <fileType>File</fileType>
                <fromPath>ds/Home</fromPath>
                <fromRevision>1</fromRevision>
                <repositoryName>localhost</repositoryName>
                <toPath>ds/Home</toPath>
                <toRevision>2</toRevision>
            </reviewItem>
            <reviewItem>
                <permId>
                    <id>CFR-2</id>
                </permId>
                <authorName>tomd</authorName>
                <commitDate>2008-09-09T16:42:28.786+1000</commitDate>
                <commitType>Added</commitType>
                <fileType>File</fileType>
                <fromPath/>
                <fromRevision/>
                <repositoryName>mylocalsvn</repositoryName>
                <toPath>aaa/bbb/qqq.txt</toPath>
                <toRevision>3</toRevision>
            </reviewItem>
        </reviewItems>
        <reviewers/>
        <transitions>
            <transitionData>
                <name>action:approveReview</name>
            </transitionData>
            <transitionData>
                <name>action:abandonReview</name>
            </transitionData>
        </transitions>
        <versionedComments/>
    </detailedReviewData>

```



## Project Service

This page last changed on Aug 21, 2008 by [tom@atlassian.com](mailto:tom@atlassian.com).

Provides access to the projects defined in a Crucible instance. At present this interface is read-only.

Method	URL	Description	Example XML
GET	/projects-v1	Returns a list of projects.	<pre>&lt;projects&gt;   &lt;projectData&gt;      &lt;allowReviewersToJoin&gt;false&lt;/allowReviewersToJoin&gt;     &lt;id&gt;1&lt;/id&gt;     &lt;key&gt;CR&lt;/key&gt;     &lt;name&gt;Default       Project&lt;/name&gt;      &lt;permissionSchemeId&gt;1&lt;/permissionSchemeId&gt;   &lt;/projectData&gt;   &lt;projectData&gt;     ...   &lt;/projectData&gt; &lt;/projects&gt;</pre>

## Repository Service

This page last changed on Sep 21, 2008 by [tom@atlassian.com](mailto:tom@atlassian.com).

Provides information about the repositories configured in a Crucible instance.



### Notes

Repositories belonging to **Light SCM Plugins** are not reported. ([CRUC-581](#))  
Detailed information is only available for SVN repositories. ([CRUC-582](#))

Method	URL	Description	Example XML
GET	/repositories-v1	Get a list of all the repositories.	<pre>&lt;repositories&gt;   &lt;repoData&gt;     &lt;enabled&gt;true&lt;/enabled&gt;     &lt;name&gt;local&lt;/name&gt;     &lt;type&gt;svn&lt;/type&gt;   &lt;/repoData&gt;   &lt;repoData&gt;     &lt;enabled&gt;true&lt;/enabled&gt;     &lt;name&gt;test&lt;/name&gt;     &lt;type&gt;cvs&lt;/type&gt;   &lt;/repoData&gt; &lt;/repositories&gt;</pre>
GET	/repositories-v1/ <repositoryName>/svn	If the repository with the name attribute which equals repositoryName is an SVN repository, return its details, otherwise return an error	<pre>&lt;svnRepositoryData&gt;   &lt;enabled&gt;true&lt;/enabled&gt;   &lt;name&gt;local&lt;/name&gt;   &lt;type&gt;svn&lt;/type&gt;   &lt;path&gt;&lt;/path&gt;   &lt;url&gt;file:///Users/tomd/dev/svn&lt;/url&gt; &lt;/svnRepositoryData&gt;</pre>

## Review Service

---

This page last changed on Sep 21, 2008 by [tom@atlassian.com](mailto:tom@atlassian.com).

The Review Service allows you to list, examine, create and modify reviews.

See the [Data Types](#) Page for the structure of the `reviewData` and `detailedReviewData` tags.

Click an item in the list below to see full details and an example.  
On this page:

- [Reviews](#)
  - [Get All Reviews](#)
  - [Get Version Info](#)
  - [Get All Reviews](#)
  - [Get Reviews by Filter](#)
  - [Get Review Details](#)
  - [Get Reviews by Path](#)
  - [Get Review Details by Path](#)
  - [Get Review](#)
  - [Delete Review](#)
  - [Get Single Review Details](#)
  - [Get Allowed Review Actions](#)
  - [Get Allowed Review Transitions](#)
  - [Create Review](#)
  - [Add Changeset To Review](#)
  - [Add Patch Revisions To Review](#)
- [Reviewers](#)
  - [Get Reviewers](#)
  - [Get Finished Reviewers](#)
  - [Get Incomplete Reviewers](#)
  - [Add Reviewers](#)
  - [Remove Single Reviewer](#)
- [Review Items](#)
  - [Get Review Items](#)
  - [Add Revision to Review](#)
  - [Remove Revision from Review](#)
  - [Get Single Revision Details](#)
- [Comments](#)
  - [Get Review Comments](#)
  - [Get General Comments](#)
  - [Get Comments on Files](#)
  - [Get Comments on a Review Item](#)
  - [Add General Comment to a Review](#)
  - [Add Comment to a Review Item](#)
  - [Update a Comment](#)
  - [Remove a Comment](#)
  - [Get the Replies to a Comment](#)
  - [Add a Reply to a Comment](#)
  - [Add a Reply to a Reply](#)
  - [Delete a Reply](#)
  - [Publish All Draft Comments](#)
  - [Publish a Single Draft Comment](#)
- [Workflow](#)
  - [Complete a Review](#)
  - [Uncomplete a Review](#)
  - [Move a Review to a New State](#)
  - [Close a Review](#)
- [Miscellaneous](#)
  - [Get Metrics](#)

## Reviews

### Get All Reviews

URL:

```
GET /reviews-v1?state=<states>
```

Description:

Get all reviews as a list of ReviewData structures. Note that this may return a lot of data, so using /reviews-v1/filter/<filter> (see below) is usually better. The state parameter is a comma separated list of state names from the set Draft, Approval, Review, Summarize, Closed, Dead, Rejected, Unknown.

Example Return Data:

```
<reviews>
  <reviewData>
    ...
  </reviewData>
  ...
</reviews>
```

### Get Version Info

URL:

```
GET /reviews-v1/versionInfo
```

Description:

Returns the release number and build date of Crucible.

Example Return Data:

```
<versionInfo>
  <buildDate>2008-08-25</buildDate>
  <releaseNumber>1.6.0</releaseNumber>
</versionInfo>
```

### Get All Reviews

URL:

```
GET /reviews-v1/details?state=<states>
```

Description:

Get all reviews as a list of DetailedReviewData structures. This may return a lot of data, so using /reviews-v1/filter/<filter>/details (see below) is usually better. The state parameter is a comma separated list of state names from the set Draft, Approval, Review, Summarize, Closed, Dead, Rejected, Unknown.

Note that the reviewItems list in the detailedReviewData elements will not appear because this URL retrieves multiple reviews.

Example Return Data:

```
<detailedReviews>
  <detailedReviewData>
    ...
  </detailedReviewData>
  ...
</detailedReviews>
```

## Get Reviews by Filter

URL:

```
GET /reviews-v1/filter/<filter>
```

Description:

Get all the reviews which match the given filter, for the current user.

Filter names are:

- allReviews - All reviews for everyone.
- allOpenReviews - Open reviews for everyone.
- allClosedReviews - Closed reviews for everyone.
- draftReviews - Draft reviews for everyone.
- toReview - Reviews on which the current user is an uncompleted reviewer.
- requireMyApproval - Reviews waiting to be approved by the current user.
- toSummarize - Completed reviews which are ready for the current user to summarize.
- outForReview - Reviews with uncompleted reviewers, on which the current reviewer is the moderator.
- drafts - Draft reviews created by the current user.
- open - Open reviews created by the current user.
- closed - Closed reviews created by the current user.
- trash - Abandoned reviews created by the current user.

Example Return Data:

```
<reviews>
  <reviewData>
    ...
  </reviewData>
  ...
</reviews>
```

## Get Review Details

URL:

```
GET /reviews-v1/filter/<filter>/details
```

**Description:**

Get details of all the reviews which match the given filter. See above for filter names.

**Example Return Data:**

```
<detailedReviews>
  <detailedReviewData>
    ...
  </detailedReviewData>
  ...
</detailedReviews>
```

## Get Reviews by Path

**URL:**

```
GET /reviews-v1/search/<repository>?path=<path>
```

**Description:**

Return a list of Reviews which include items in the given repository, under the given path.

**Example Return Data:**

```
<reviews>
  <reviewData>
    ...
  </reviewData>
  ...
</reviews>
```

## Get Review Details by Path

**URL:**

```
GET /reviews-v1/search/<repository>/details?path=<path>
```

**Description:**

Return a list of Review details which include items in the given repository, under the given path.

**Example Return Data:**

```
<detailedReviews>
  <detailedReviewData>
    ...
```

```
</detailedReviewData>
...
</detailedReviews>
```

## Get Review

URL:

```
GET /reviews-v1/<id>
```

Description:  
Get a single review.

Example Return Data:

```
<reviewData>
...
</reviewData>
```

## Delete Review

URL:

```
DELETE /reviews-v1/<id>
```

Description:  
Delete a review. The review must have been abandoned.

## Get Single Review Details

URL:

```
GET /reviews-v1/<id>/details
```

Description:  
Get details of a single review.

Example Return Data:

```
<detailedReviewData>
...
</detailedReviewData>
```

## Get Allowed Review Actions

URL:

```
GET /reviews-v1/<id>/actions
```

### Description:

Get a list of the actions which the current user is allowed to perform on the review. This shows actions the user has permission to perform – the review may not be in a suitable state for all these actions to be performed.

### Example Return Data:

```
<actions>
  <actionData>
    <name>action:summarizeReview</name>
  </actionData>
  <actionData>
    <name>action:viewReview</name>
  </actionData>
  <actionData>
    <name>action:approveReview</name>
  </actionData>
  <actionData>
    <name>action:closeReview</name>
  </actionData>
  <actionData>
    <name>action:modifyReviewFiles</name>
  </actionData>
  <actionData>
    <name>action:rejectReview</name>
  </actionData>
  <actionData>
    <name>action:deleteReview</name>
  </actionData>
  <actionData>
    <name>action:createReview</name>
  </actionData>
  <actionData>
    <name>action:recoverReview</name>
  </actionData>
  <actionData>
    <name>action:commentOnReview</name>
  </actionData>
  <actionData>
    <name>action:reopenReview</name>
  </actionData>
  <actionData>
    <name>action:abandonReview</name>
  </actionData>
  <actionData>
    <name>action:submitReview</name>
  </actionData>
</actions>
```

## Get Allowed Review Transitions

URL:



```
GET /reviews-v1/<id>/transitions
```

**Description:**

Get a list of the actions which the current user can perform on this review, given its current state and the user's permissions.

**Example Return Data:**

```
<transitions>
  <transitionData>
    <name>action:summarizeReview</name>
  </transitionData>
  <transitionData>
    <name>action:abandonReview</name>
  </transitionData>
</transitions>
```

## Create Review

**URL:**

```
POST /reviews-v1
```

**Description:**

Create a review. A review can be created in one of three ways(see examples).

**Example Return Data:**

1. An empty review, i.e. no revisions in the review. This uses the XML:

```
<createReview>
  <reviewData>
    ...
  </reviewData>
</createReview>
```

2. A patch review, containing diffs from a patch file, e.g. created by `svn diff >patch.txt`. XML:

```
<createReview>
  <reviewData>
    ...
  </reviewData>
  <patch>
    <![CDATA[
... text of patch goes here ...
]]>
  </patch>
</createReview>
```

3. A review containing revisions from a set of changesets. XML:

```
<createReview>
  <reviewData>
    ...
  </reviewData>
  <changesets>
    <changesetData>
      <id>...the id...</id>
    </changesetData>
    ... more changesets ...
  </changesets>
</createReview>
```

In all these cases, the `reviewData` structure shouldn't have the `permaId` or `state` attributes set. The response is a `reviewData` structure fully populated.

## Add Changeset To Review

URL:

```
POST /reviews-v1/<review id>/addChangeset
```

Description:

Add the revisions in a set of changesets to an existing review.

Example Return Data:

```
<addChangeset>
  <repository>aRepositoryName</repository>
  <changesets>
    <changesetData>
      <id>...the id...</id>
    </changesetData>
    ... more change sets ...
  </changesets>
</addChangeset>
```

The response is the `reviewData` structure of the review.

## Add Patch Revisions To Review

URL:

```
POST /reviews-v1/<review id>/addPatch
```

Description:

Add the revisions in a patch to an existing review.

Example Return Data:

```
<addPatch>
  <repository>aRepositoryName</repository>
```

```
<patch>
  <![CDATA[
... text of patch goes here ...
  ]]>
</patch>
</addPatch>
```

The response is the `reviewData` structure of the review.

## Reviewers

### Get Reviewers

URL:

```
GET /reviews-v1/<review id>/reviewers
```

Description:

Return a list of the reviewers participating in the review.

Example Return Data:

```
<reviewers>
  <reviewer>
    <displayName>Conor MacNeill</displayName>
    <userName>conor</userName>
    <completed>false</completed>
  </reviewer>
  ... more reviewers ...
</reviewers>
```

### Get Finished Reviewers

URL:

```
GET /reviews-v1/<review id>/reviewers/completed
```

Description:

Return a list of the reviewers who have completed the review.

Example Return Data:

Return value as `/reviews-v1/<review id>/reviewers`, but only completed reviewers are included.

### Get Incomplete Reviewers

URL:

```
GET /reviews-v1/<review id>/reviewers/uncompleted
```

Description:

Return a list of the reviewers who have not yet completed the review.

Example Return Data:

Return value as `/reviews-v1/<review id>/reviewers`, but only incomplete reviewers are included.

## Add Reviewers

URL:

```
POST /reviews-v1/<review id>/reviewers
```

Description:

Add new reviewers to the review. Send a string of comma separated user names.

## Remove Single Reviewer

URL:

```
DELETE /reviews-v1/<review id>/reviewers/<username>
```

Description:

Remove a reviewer from a review.

## Review Items

### Get Review Items

URL:

```
GET /reviews-v1/<review id>/reviewitems
```

Description:

Get a list of the items in a review.

Example Return Data:

```
<reviewItems>
  <reviewItem>
    <permId>
      <id>CFR-1</id>
    </permId>
    <authorName>tomd</authorName>
    <commitDate>2008-01-29T14:41:43.202+1100</commitDate>
    <commitType>Modified</commitType>
    <fileType>File</fileType>
    <fromPath>foo.txt</fromPath>
    <fromRevision>21</fromRevision>
    <repositoryName>local</repositoryName>
    <toPath>foo.txt</toPath>
    <toRevision>22</toRevision>
  </reviewItem>
  ... more reviewItems ...
```

```
</reviewItems>
```

## Add Revision to Review

URL:

```
POST /reviews-v1/<review id>/reviewitems
```

Description:

Add a revision to a review. Send a `[reviewItem]` with the repository name, and from and to paths and revisions specified. Other values can be omitted. This returns the completed `reviewItem` structure.

## Remove Revision from Review

URL:

```
DELETE /reviews-v1/<review id>/reviewitems/<review item id>
```

Description:

Remove a revision from a review.

## Get Single Revision Details

URL:

```
GET /reviews-v1/<review id>/reviewitems/<review item id>
```

Description:

Get the details of a single revision in a review. Returns the `reviewItem` structure for the item.

# Comments

## Get Review Comments

URL:

```
GET /reviews-v1/<review id>/comments
```

Description:

Get all the comments made on a review. The `versionedLineCommentData` tag may contain `fromLineRange` and `toLineRange` tags, indicating that the comment was made against a specific range of lines.

Example Return Data:

```
<comments>  
  <generalCommentData>
```

```

    <createDate>2008-09-16T16:28:02.833+1000</createDate>
    <defectApproved>>false</defectApproved>
    <defectRaised>>false</defectRaised>
    <deleted>>false</deleted>
    <draft>>false</draft>
    <message>This is a general comment.</message>
    <metrics/>
    <permaIdAsString>CMT:1</permaIdAsString>
    <replies/>
    <user>
        <displayName>Matt Quail</displayName>
        <userName>matt</userName>
    </user>
    <permId>
        <id>CMT:1</id>
    </permId>
</generalCommentData>
<versionedLineCommentData>
    <createDate>2008-09-16T16:28:26.432+1000</createDate>
    <defectApproved>>false</defectApproved>
    <defectRaised>true</defectRaised>
    <deleted>>false</deleted>
    <draft>>false</draft>
    <message>This is a revision level defect.</message>
    <metrics>
        <entry>
            <key>classification</key>
            <value>
                <configVersion>1</configVersion>
                <value>Inconsistent</value>
            </value>
        </entry>
        <entry>
            <key>rank</key>
            <value>
                <configVersion>1</configVersion>
                <value>Major</value>
            </value>
        </entry>
    </metrics>
    <permaIdAsString>CMT:2</permaIdAsString>
    <replies/>
    <user>
        <displayName>Matt Quail</displayName>
        <userName>matt</userName>
    </user>
    <permId>
        <id>CMT:2</id>
    </permId>
    <reviewItemId>
        <id>CFR-4</id>
    </reviewItemId>
</versionedLineCommentData>
<versionedLineCommentData>
    <createDate>2008-09-16T16:28:54.604+1000</createDate>
    <defectApproved>>false</defectApproved>
    <defectRaised>>false</defectRaised>
    <deleted>>false</deleted>
    <draft>>false</draft>
    <message>This is a comment covering two lines of a revision.</message>
    <metrics/>
    <permaIdAsString>CMT:3</permaIdAsString>
    <replies/>
    <user>
        <displayName>Matt Quail</displayName>
        <userName>matt</userName>

```

```

        </user>
        <permaId>
            <id>CMT:3</id>
        </permaId>
        <reviewItemId>
            <id>CFR-4</id>
        </reviewItemId>
        <fromLineRange>2-3</fromLineRange>
        <toLineRange>3-4</toLineRange>
    </versionedLineCommentData>
</comments>

```

## Get General Comments

URL:

```
GET /reviews-v1/<review id>/comments/general
```

Description:

Get all the general comments made on a review – that is, comments which are not attached to a particular file in the review.

Example Return Data:

```

<comments>
    <generalCommentData>
        ...
    </generalCommentData>
</comments>

```

## Get Comments on Files

URL:

```
GET /reviews-v1/<review id>/comments/versioned
```

Description:

Get all the versioned comments made on a review – that is, comments which are on a particular file in the review.

Example Return Data:

```

<comments>
    <versionedLineCommentData>
        ...
    </versionedLineCommentData>
    ...
</comments>

```

## Get Comments on a Review Item

URL:

```
GET /reviews-v1/<review id>/reviewitems/<review item id>/comments
```

Description:

Get all the comments made on a review item.

Example Return Data:

```
<comments>
  <versionedLineCommentData>
    ...
  </versionedLineCommentData>
  ...
</comments>
```

## Add General Comment to a Review

URL:

```
POST /reviews-v1/<review id>/comments
```

Description:

Add a general comment to a review. Returns the completed `generalCommentData` structure.

The POST data is a `generalCommentData` structure.

## Add Comment to a Review Item

URL:

```
POST /reviews-v1/<review id>/reviewitems/<review item id>/comments
```

Description:

Add a comment to a review. Returns the completed `versionedLineCommentData` structure.

The POST data is a `versionedLineCommentData` structure.

## Update a Comment

URL:

```
POST /reviews-v1/<review id>/comments/<comment id>
```

Description:

Update an existing comment.



The POST data is a `generalCommentData` structure.

## Remove a Comment

URL:

```
DELETE /reviews-v1/<review id>/comments/<comment id>
```

Description:

Remove an existing comment.

## Get the Replies to a Comment

URL:

```
GET /reviews-v1/<review id>/comments/<comment id>/replies
```

Description:

Get the replies to an existing comment.

Example Return Data:

```
<comments>
  <generalCommentData>
    ...
  </generalCommentData>
  ...
</comments>
```

## Add a Reply to a Comment

URL:

```
POST /reviews-v1/<review id>/comments/<comment id>/replies
```

Description:

Add a reply to an existing comment.

The POST data is a `generalCommentData` structure.

## Add a Reply to a Reply

URL:

```
POST /reviews-v1/<review id>/comments/<comment id>/replies/<reply id>
```

Description:

Add a reply to a reply to an existing comment.

The POST data is a `generalCommentData` structure.

## Delete a Reply

URL:

```
DELETE /reviews-v1/<review id>/comments/<comment id>/replies/<reply id>
```

Description:  
Delete a reply.

## Publish All Draft Comments

```
POST /reviews-v1/<review id>/publish
```

Description:  
Publish all the user's draft comments on the review.

## Publish a Single Draft Comment

```
POST /reviews-v1/<review id>/publish/<comment id>
```

Description:  
Publish a draft comments on the review.

## Workflow

### Complete a Review

```
POST /reviews-v1/<review id>/complete
```

Description:  
Indicate that the current user has completed the review.

### Uncomplete a Review

```
POST /reviews-v1/<review id>/uncomplete
```

Description:  
Indicate that the current user has not completed the review.

## Move a Review to a New State

```
POST /reviews-v1/<review id>/transition?action=<action>
```

### Description:

Change the state of the review.

### Valid actions are:

```
action:abandonReview
action:deleteReview
action:submitReview
action:approveReview
action:rejectReview
action:summarizeReview
action:closeReview
action:reopenReview
action:recoverReview
action:completeReview
action:uncompleteReview
```

## Close a Review

```
POST /reviews-v1/<review id>/close
```

### Description:

Close the review.

## Miscellaneous

### Get Metrics

#### URL:

```
GET /reviews-v1/metrics/<version>
```

### Description:

Get the replies to an existing comment.

### Example Return Data:

```
<?xml version="1.0"?>
<metrics>
  <metricsData>
    <configVersion>1</configVersion>
    <defaultValue>
      <name>Minor</name>
      <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">0</value>
    </defaultValue>
    <label>Ranking</label>
    <name>rank</name>
    <type>INTEGER</type>
    <values>
```

```

        <name>Major</name>
        <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">1</value>
        </values>
        <values>
            <name>Minor</name>
            <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">0</value>
            </values>
        </metricsData>
        <metricsData>
            <configVersion>1</configVersion>
            <defaultValue>
                <name>Improvement desirable</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">5</value>
            </defaultValue>
            <label>Classification</label>
            <name>classification</name>
            <type>INTEGER</type>
            <values>
                <name>Missing</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">1</value>
            </values>
            <values>
                <name>Extra (superfluous)</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">2</value>
            </values>
            <values>
                <name>Ambiguous</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">3</value>
            </values>
            <values>
                <name>Inconsistent</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">4</value>
            </values>
            <values>
                <name>Improvement desirable</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">5</value>
            </values>
            <values>
                <name>Not conforming to standards</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">6</value>
            </values>
            <values>
                <name>Risk-prone</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">7</value>
            </values>
            <values>
                <name>Factually incorrect</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">8</value>
            </values>
            <values>
                <name>Not implementable</name>
                <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">9</value>
            </values>
            <values>

```

```
        <name>Editorial</name>
        <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xsi:type="xs:int">10</value>
    </values>
</metricsData>
</metrics>
```

## Crucible REST API Usage Example

This page last changed on Jun 01, 2008 by [tom@atlassian.com](mailto:tom@atlassian.com).

This page describes using the Crucible REST API to retrieve comments from reviews in Crucible. It's an overview of using the API, not a comprehensive reference.

On this page:

- [Authentication](#)
- [Retrieving Reviews](#)
- [Retrieving Reviews in a Specific State](#)
- [Retrieving Comments From a Review](#)
- [Retrieving Properties of a File Under Review](#)

The Crucible REST API lives under the URL <http://HOSTNAME:PORT/CONTEXT/rest-service/>, where `HOSTNAME:PORT` is the IP address and port of your FishEye instance and `CONTEXT` is the web application context it is deployed under.

This page doesn't assume any particular REST client is being used – it just discusses the URLs to use and the responses which they will give. The information returned is in XML format.

This page assumes Crucible 1.5 – the example won't work with earlier versions.

### Authentication

Requests to the REST API are simply HTTP requests, which can use any of the normal Crucible authentication methods. An unauthenticated request will execute as the anonymous user.

Authentication options are:

- The normal Crucible login cookie. A cookie named 'remember' in the request with the token returned by the REST authentication service on <http://HOSTNAME:PORT/rest-service/auth-v1/login?userName=jim&password=jimspassword>. This will return `<?xml version="1.0" encoding="UTF-8" standalone="yes"?><loginResult><token>jim:2:4455f9a4387298a83aae6902e8843f89</token></loginResult>`. The value of the cookie should be set to `jim:2:4455f9a4387298a83aae6902e8843f89`.
- Trusted Applications. If Crucible trusts the application which is making the request, the user logged in to the trusted application will be authenticated in Crucible.
- Crowd. If Crucible is configured to use Crowd, then a request containing Crowd authentication will authenticate the Crowd user in Crucible.
- Basic Authentication. An RFC 2617 Basic Authentication header.

### Retrieving Reviews

This example will use the reviews service, at the URL <http://HOSTNAME:PORT/rest-service/reviews-v1>. A simple get on this URL will return every review in the system. The results will look like this:

```
<reviews>
  <reviewData>
    <author>pmcneil</author>
    <creator>pmcneil</creator>
    <description>14699: CRUC-230: allow links to be removed
14698: CRUC-230: don't allow linking cycles</description>
    <moderator>pmcneil</moderator>
    <name>CRUC-214: Generate comment/defect and open review report graphs</name>
    <permaId>
      <id>CR-FE-1</id>
    </permaId>
    <projectKey>CR-FE</projectKey>
    <repoName>FE</repoName>
    <state>Review</state>
  </reviewData>
  ...
```

```
</reviews>
```

## Retrieving Reviews in a Specific State

If you don't want to retrieve every review, you can specify a value for the `state` parameter: <http://HOSTNAME:PORT/rest-service/reviews-v1?state=Review,Summarize> to retrieve only those reviews in particular states.

The request only returns those reviews that the authenticated user is allowed to see.

Once you have the reviews you can use their `permaId` to get more details, so: <http://HOSTNAME:PORT/rest-service/reviews-v1/CR-FE-1> will return a single `reviewData` element, identical to the one shown above.

## Retrieving Comments From a Review

URLs like <http://HOSTNAME:PORT/rest-service/reviews-v1/CR-FE-1/thing> will return information about `thing` records belonging to the review.

So <http://HOSTNAME:PORT/rest-service/reviews-v1/CR-FE-1/comments> returns all the comments in the review:

```
<comments>
  <versionedComment>
    <createDate>2008-03-03T22:22:00.920+11:00</createDate>
    <defectApproved>false</defectApproved>
    <defectRaised>false</defectRaised>
    <deleted>false</deleted>
    <draft>false</draft>
    <message>why use roll instead of add??</message>
    <permaId>
      <id>CMT:200</id>
    </permaId>
    <reviewItemId>
      <id>CFR-281</id>
    </reviewItemId>
    <user>mquail</user>
    <fromLineRange>196-199, 230-233</fromLineRange>
    <toLineRange>206-209, 240-251</toLineRange>
  </versionedComment>
  ... more versioned comments ...
  <generalComment>
    <createDate>2008-03-25T17:15:20.380+11:00</createDate>
    <defectApproved>false</defectApproved>
    <defectRaised>true</defectRaised>
    <deleted>false</deleted>
    <draft>false</draft>
    <message>when there are no comments in the last week the vertical axis shows -5 as the starting
point</message>
    <user>pmcneil</user>
  </generalComment>
  ... more general comments ...
</comments>
```

## Retrieving Properties of a File Under Review

If you need more information about the file a versioned comment was on, the URL <http://HOSTNAME:PORT/rest-service/reviews-v1/CR-FE-1/reviewitems/CFR-281> gives more details:

```
<fisheyeReviewItemData>
  <permId>
    <id>CFR-281</id>
  </permId>
  <fromPath></fromPath>
  <fromRevision></fromRevision>
  <repositoryName>FE</repositoryName>
  <toPath>branches/iteration03/src/java/com/cenqua/crucible/reports/CommentsDefects/
CommentDatasetMaker.java</toPath>
  <toRevision>13947</toRevision>
</fisheyeReviewItemData>
```

That particular review item is a new file, so the `fromPath` and `fromRevision` elements are empty.



## 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 1.5 Release Notes](#)
  - [Crucible 1.6 Release Notes](#)
  - [Crucible Release Summary](#)
- [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 Apr 14, 2008 by [edawson](#).

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

Java Runtime	A JDK or JRE version 1.5 or greater. You can download a Java Runtime for Windows/Linux/Solaris <a href="#">here</a> . 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. Currently, CVS, Subversion and Perforce are supported.
Web Browser	<p>Crucible has been tested in the latest releases of Firefox 2, Internet Explorer 7, and Safari 3. Crucible is known to work with Internet Explorer 6, FireFox 3 Beta and Safari 2. It should work on any modern browser.</p> <div> <b>Font size</b> (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.</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 May 28, 2008 by [edawson](#).

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](#).
5. If you intend to use Crucible/FishEye with [CVS](#), please ensure you read the [System Requirements](#) and [CVS Client setup](#).

Next Step - Initial Configuration

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


### 3. Configuring Crucible

This page last changed on Sep 23, 2008 by [edawson](#).

On this page:

- [Running Crucible](#)
- [Supplying Administration Password and License Key](#)
- [Accessing the Administration Pages](#)
- [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

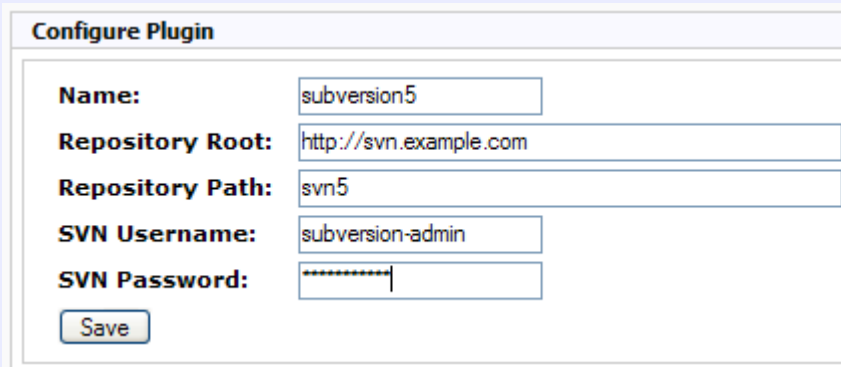
Once you have set up an administrator password (as described above), you can access the Administration screens at <http://HOSTNAME:8060/admin/>.

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 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 LightSVN repository plugin and click its Configure Plugin link.
5. The 'Configure Plugin' screen opens. Click 'Add Repository'.
6. The 'Add Repository' screen opens. Fill in the fields.



**Configure Plugin**


**Name:**

**Repository Root:**


**Repository Path:**

**SVN Username:**

**SVN Password:**

Field	What to enter
Name	Choose a unique name for the repository.
Repository Root	Enter the base URL for the repository, for example 'http://svn.example.com'.
Repository Path	Add the path on the base URL where your repository. For example, if you used the root URL above, and the full path to your Subversion instance is 'http://svn.example.com/svn5/', you would enter 'svn5' into this field.
SVN Username	Enter the username of the Subversion account that Crucible will use.  Note that this account should only have read-only access to the repository.
SVN Password	Enter the password of the Subversion account that Crucible will use.

1. Click 'Save'. The view will return to the list of repositories.
2. 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 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 Confluence repository plugin and click its Configure Plugin link.
5. The 'Configure Plugin' screen opens. Click 'Add Repository'.
6. The 'Add Repository' screen opens. Fill in the fields.



**Configure Plugin**

**Name:**


**URL:**

**Space Key:**

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

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

1. Click 'Save'. The view will return to the list of repositories.
2. Confluence is now set up as a code repository for Crucible. You will be able to select changesets from it when creating reviews, and browse the spaces and pages of the instance.

 You need to install the Confluence Crucible Plugin in the Confluence instance, and you must set up a trust relationship so that your Confluence instance trusts your Crucible instance.

## Setting Up the File System as a Code Repository For Crucible

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

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

Configure Plugin


Name: ServerFiles

Base Path: \code\java\

Save

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

## Setting Up a Repository via FishEye

 This section requires a working [FishEye](#) installation.

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 Users

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

To add users:

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

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

## Setting Up SMTP

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

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

## Using Crucible

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

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



## Stopping Crucible

To stop the Crucible server:

- On Windows:

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

- On Unix-based systems:

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

## Information on FishEye integration

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

## Crucible Release Notes

---

This page last changed on Apr 23, 2008 by [edawson](#).



Crucible 1.6 has now been released. Read the [Release Notes](#).

### Crucible Release Notes and Changelogs

- [Crucible Release Summary](#)
- [Crucible 1.6 Release Notes](#)
- [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 1.1 Release Notes

---

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



Crucible 1.6 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

---

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:

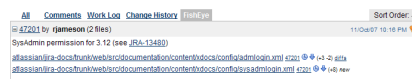
- Administrator can make the email address for [self-signups](#) optional.
- Improved user interface makes user administration easier.
- [Groups](#) are supported.
- Read the [FishEye documentation](#).

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
<a href="#">CRUC-181</a>	<a href="#">Old screenshot in 1.2 release notes</a>	↑	Closed
<a href="#">CRUC-175</a>	<a href="#">LHS abandon button on Crucible review screen broken</a>	↑	Resolved
<a href="#">CRUC-166</a>	<a href="#">Some files not work</a>	↑	Resolved
<a href="#">CRUC-152</a>	<a href="#">sysinfo admin screen should</a>	↑	Resolved



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

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

	<a href="#">Review</a> <a href="#">From</a> <a href="#">Clipboard</a>
--	---

## Crucible 1.2 Changelog

This page last changed on Feb 06, 2008 by [edawson](#).





































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](#)).




Atlassian JIRA (26 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-127</a>	<a href="#">Improperly stopping crucible causes data loss!</a>			Closed
<a href="#">CRUC-274</a>	<a href="#">Create IDE integration with IntelliJ IDEA</a>			Closed
<a href="#">CRUC-273</a>	<a href="#">Create IDE integration with Eclipse</a>			Closed
<a href="#">CRUC-256</a>	<a href="#">Check boxes for checking off files reviewed in a review are lost if navigate to raw text by "download raw text" icon</a>			Closed
<a href="#">CRUC-249</a>	<a href="#">Accept review comments in Japanese (unicode characters)</a>			Closed
<a href="#">CRUC-243</a>	<a href="#">race condition between crucible backup and repository scan</a>			Closed
<a href="#">CRUC-235</a>	<a href="#">File tabs should be kept open</a>			Closed
<a href="#">CRUC-210</a>	<a href="#">Create JIRA issues from code reviews</a>			Closed
<a href="#">CRUC-208</a>	<a href="#">Nullpointer exception when creating new review</a>			Closed
<a href="#">CRUC-202</a>	<a href="#">update 'Repositories' screenshot</a>			Closed
<a href="#">CRUC-196</a>	<a href="#">Error on attempting to create a Review.</a>			Closed
<a href="#">CRUC-180</a>	<a href="#">'Show Full Source' should continue to highlight the diff</a>			Closed
<a href="#">CRUC-177</a>	<a href="#">Comments on reviews are being sent but don't appear in the review when viewed online</a>			Closed
<a href="#">CRUC-165</a>	<a href="#">Folder Issue</a>			Closed
<a href="#">CRUC-139</a>	<a href="#">Can't add new versions of the same file to a review</a>			Closed
<a href="#">CRUC-112</a>	<a href="#">Point online help links to new Crucible doc space</a>			Closed
<a href="#">CRUC-106</a>	<a href="#">Add content to new documentation page 'Crucible and FishEye'</a>			Closed
<a href="#">CRUC-105</a>				Closed

	<a href="#">Upload PDF, XML and HTML versions of Crucible docs</a>			
<a href="#">CRUC-103</a>	<a href="#">Create admin page for Projects</a>	↑		Closed
<a href="#">CRUC-100</a>	<a href="#">Move Crucible docs to Confluence</a>	↑		Closed
<a href="#">CRUC-99</a>	<a href="#">Feature request: I would like to see statistics about number of major and minor issues found per developer.</a>	↑		Closed
<a href="#">CRUC-92</a>	<a href="#">Allow Anonymous Access Per Repository</a>	↑		Closed
<a href="#">CRUC-84</a>	<a href="#">Ability to create a review directly from JIRA</a>	↑		Closed
<a href="#">CRUC-50</a>	<a href="#">Allow review of entire file, not a particular diff</a>	↑		Closed
<a href="#">CRUC-28</a>	<a href="#">After Crucible forums are moved, update all links in docs</a>	↑		Closed
<a href="#">CRUC-258</a>	<a href="#">Allow all users to be able to a "Moderator" or "Author" in a review project</a>	↓		Closed

## From 1.2.1 to 1.2.2







This release contains some minor improvements and bug fixes.

- **Trusted Application Support**  
FishEye/Crucible now allows you to set up trusted communications with other Atlassian applications. At this point, the JIRA FishEye plugin supports Trusted Applications. The JIRA FishEye plugin can request information from FishEye on behalf of the currently logged-in user, and FishEye will not ask the user to log in again or to supply a password. Previously FishEye/Crucible would have used a single 'system' account to determine permissions. Now, FishEye/Crucible can apply the correct permission settings for the logged-in user.
- Hyphens are now allowed in project key names.

<a href="#">Atlassian JIRA</a> (3 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-207</a>	<a href="#">allow hyphens in Crucible project keys</a>	↑		Closed
<a href="#">CRUC-204</a>	<a href="#">Add project to dashboard filter</a>	↑		Closed
<a href="#">CRUC-195</a>	<a href="#">groups access control to repositories failed after configured External authentication (LDAP)</a>	↑		Closed

## From 1.2 to 1.2.1

This is a small bug-fix release.

Atlassian JIRA (6 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-199</a>	<a href="#">Top of project edit form says Repositories List</a>	↑		Resolved
<a href="#">CRUC-198</a>	<a href="#">multiple copies of revisions added to review if you add changesets in reverse chronological order</a>	↑		Resolved
<a href="#">CRUC-182</a>	<a href="#">Projects Don't Persist Through Pages</a>	↑		Resolved
<a href="#">CRUC-179</a>	<a href="#">Support addition of multiple revisions of the same file that appears in multiple changesets</a>	↑		Resolved
<a href="#">CRUC-178</a>	<a href="#">Hit NPE when perform backup</a>	↑		Resolved
<a href="#">CRUC-40</a>	<a href="#">Getting wrong changeset number causes diff to be missing</a>	↑		Closed

## Crucible 1.2 Upgrade Guide

---

This page last changed on Jan 09, 2008 by [edawson](#).

### 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 1.5 Release Notes

This page last changed on Apr 14, 2008 by [edawson](#).

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](#).

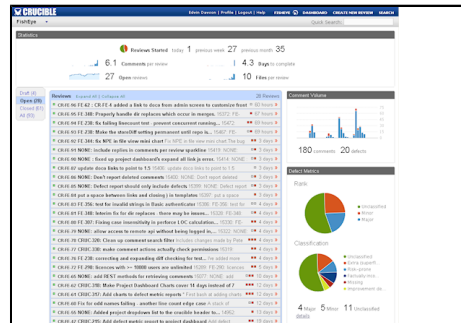
## Highlights of Crucible 1.5

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



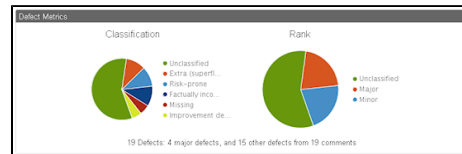


2

## Filtered comments & defects search, with statistical summary

Defects and comments are now searchable, easing the difficulty of finding a particular piece of work or revision (and its relevant comments). These search results now also show a very useful statistical summary. Also, a new defect metrics report is available.

Screenshot: Crucible Defect Metrics Report



3

## Customisable email templates

You can now customise the content and appearance of email notifications that get sent to Crucible users. For example you can append a legal disclaimer, alter the subject line or provide custom header text for all messages.

4















## Improvements to Crucible Plugin API beta

Now with REST support and the ability to upload patches, the Crucible Plugin API beta is for Crucible integrators who want to extend Crucible to interoperate with their enterprise infrastructure or processes.







## Plus numerous improvements and bug-fixes

Atlassian JIRA (38 issues)			
Key	Summary	Pr	Status
<a href="#">CRUC-344</a>	<a href="#">Defect pie chart on comment search page includes non-defect comments.</a>	↑	Closed
<a href="#">CRUC-332</a>	<a href="#">create a better looking indication icon</a>	↑	Closed
<a href="#">CRUC-324</a>	<a href="#">Add Project Dropdown</a>	↑	Closed

	<a href="#">to Search Pages</a>			
<a href="#">CRUC-323</a>	<a href="#">Create 'Reviews Started' Pie Chart for Project Dashboard</a>			Closed
<a href="#">CRUC-318</a>	<a href="#">Make Project Dashboard Charts cover 14 days instead of 7</a>			Closed
<a href="#">CRUC-314</a>	<a href="#">sidebyside diff line numbers must be aligned to the top</a>			Closed
<a href="#">CRUC-307</a>	<a href="#">Optimize chart queries</a>			Closed
<a href="#">CRUC-299</a>	<a href="#">link to project page</a>			Closed
<a href="#">CRUC-289</a>	<a href="#">permissions are not checked in rss feeds (but are in list html display)</a>			Closed
<a href="#">CRUC-285</a>	<a href="#">"project project" in Permission Screen in admin</a>			Closed
<a href="#">CRUC-283</a>	<a href="#">Escape HTML in Javascript properly</a>			Closed
<a href="#">CRUC-280</a>	<a href="#">remove debug log from Process Notification</a>			Closed
<a href="#">CRUC-247</a>	<a href="#">Flag files which have been modified since a review was created</a>			Closed
<a href="#">CRUC-246</a>	<a href="#">Parentheses should be valid URL characters</a>			Closed
<a href="#">CRUC-242</a>	<a href="#">Add RPC call to create review from patch</a>			Closed
<a href="#">CRUC-240</a>	<a href="#">Checking box in "manage</a>			Closed

	<a href="#">files" -&gt; "changesets" spins forever</a>			
<a href="#">CRUC-232</a>	<a href="#">Address all issues outline here</a>			Closed
<a href="#">CRUC-220</a>	<a href="#">permalink for a comment does not work when summarize mode is active</a>			Closed
<a href="#">CRUC-212</a>	<a href="#">Create Project Dashboard page</a>			Closed
<a href="#">CRUC-206</a>	<a href="#">URL recognition in Crucible comment does not always match whole string</a>			Closed
<a href="#">CRUC-201</a>	<a href="#">Allow project to be specified on create review URL</a>			Closed
<a href="#">CRUC-194</a>	<a href="#">Documentation: restoring Crucible backup data</a>			Closed
<a href="#">CRUC-184</a>	<a href="#">When Creating Review From FishEye, Default to Right Project</a>			Closed
<a href="#">CRUC-172</a>	<a href="#">Create 'Getting started with Crucible' document as part of User Guide revisions</a>			Closed
<a href="#">CRUC-147</a>	<a href="#">Add Permissions Checks to All Operations on Reviews</a>			Closed
<a href="#">CRUC-128</a>	<a href="#">email notification of review closure prints literal "null" for</a>			Closed

	<a href="#">absent field value</a>			
<a href="#">CRUC-85</a>	<a href="#">Sometimes we need to see differences in UNICODE files.</a>	↑		Closed
<a href="#">CRUC-21</a>	<a href="#">change icon for "View History In Fisheye"</a>	↑		Closed
<a href="#">CRUC-290</a>	<a href="#">Warnings during backup: File does not exist /data/crucible/inst/var/data/crddb/crucible.lck &amp; File does not exist /data/crucible/inst/var/data/crddb/crucible.log</a>	↓		Closed
<a href="#">CRUC-282</a>	<a href="#">Crucible fails to upload patch file with file information that contain blank spaces between file path and timestamp</a>	↓		Closed
<a href="#">CRUC-266</a>	<a href="#">let the subject of the email be modifiable via the notification template</a>	↓		Closed
<a href="#">CRUC-260</a>	<a href="#">How to customize the email notification content</a>	↓		Closed
<a href="#">CRUC-248</a>	<a href="#">Configurable email subject line</a>	↓		Closed
<a href="#">CRUC-241</a>	<a href="#">Incorrect dates in RSS feeds</a>	↓		Closed
<a href="#">CRUC-230</a>	<a href="#">Crucible allows circular review linking and can't delete</a>	↓		Closed

		<a href="#">a review link</a>			
	<a href="#">CRUC-226</a>	<a href="#">Title wrong for post-approval in manage files tab</a>			Closed
	<a href="#">CRUC-162</a>	<a href="#">From field on emails is incorrect or a least deceiving</a>			Closed
	<a href="#">CRUC-238</a>	<a href="#">Perforce unit tests do not work on Windows platform</a>			Closed

## Crucible 1.5 Changelog

This page last changed on Aug 01, 2008 by [edawson](#).











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.

Atlassian JIRA (10 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-638</a>	<a href="#">How to (easily) add specific changeset to a review</a>	↑		Closed
<a href="#">CRUC-513</a>	<a href="#">If we display no change on a revision, comments made are hidden by default.</a>	↑		Resolved
<a href="#">CRUC-503</a>	<a href="#">Upgrade to crowd-integration-client 1.4.4</a>	↑		Closed
<a href="#">CRUC-498</a>	<a href="#">Draft comment replies are deleted when completing review</a>	↑		Closed
<a href="#">CRUC-454</a>	<a href="#">two notification emails sent</a>	↑		Closed
<a href="#">CRUC-429</a>	<a href="#">Prevent two projects with same key different case</a>	↑		Closed
<a href="#">CRUC-284</a>	<a href="#">AnnotatorTag error: Could not get blame for file</a>	↑		Closed
<a href="#">CRUC-633</a>	<a href="#">How to remove a reviewer from a review</a>	↓		Closed
<a href="#">CRUC-538</a>	<a href="#">Comments weirdly duplicated on occasion</a>	↓		Resolved
<a href="#">CRUC-417</a>	<a href="#">large var/data/crddb/crucible.log</a>	↓		Closed

### From 1.5.2 to 1.5.3

23 June 2008

































This release contains bug fixes.

Atlassian JIRA (1 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-462</a>	<a href="#">"File closed: var\data\data0.bin" after backups</a>	↑		Resolved

## From 1.5.1 to 1.5.2

27 May 2008



















This release contains bug fixes.

Atlassian JIRA (16 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-262</a>	<a href="#">Urgent : SVNRepository methods are not reenterable</a>			Closed
<a href="#">CRUC-405</a>	<a href="#">[Performance] AnnotatorTag can hang in sort loop for (perhaps) large diffs</a>			Resolved
<a href="#">CRUC-370</a>	<a href="#">create action to remove all revisions from a review</a>			Closed
<a href="#">CRUC-363</a>	<a href="#">Remove all Revisions in manager files-&gt; Changesets tab doesn't work</a>			Resolved
<a href="#">CRUC-355</a>	<a href="#">Review Stats Wrong</a>			Resolved
<a href="#">CRUC-354</a>	<a href="#">Personal dashboard stats don't match</a>			Resolved
<a href="#">CRUC-336</a>	<a href="#">can't search for review keys in quicksearch</a>			Resolved
<a href="#">CRUC-185</a>	<a href="#">Abandoned Reviews Still Show in FishEye</a>			Resolved
<a href="#">CRUC-434</a>	<a href="#">'Expand Commented' appears when there are general comments but no revision comments</a>			Resolved
<a href="#">CRUC-413</a>	<a href="#">Crucible always creates a new default project after it is deleted</a>			Resolved
<a href="#">CRUC-406</a>	<a href="#">Remove 'All Projects' from projects dropdown</a>			Resolved
<a href="#">CRUC-376</a>	<a href="#">Prevent deletion of default project</a>			Resolved
<a href="#">CRUC-371</a>	<a href="#">Comment volume chart reports confusion</a>			Resolved
<a href="#">CRUC-358</a>	<a href="#">Height not respected if less than 144 in defect chart</a>			Resolved
<a href="#">CRUC-335</a>	<a href="#">[CSS] project list a little wonky in IE</a>			Resolved
<a href="#">CRUC-233</a>	<a href="#">Changing or expanding the term - 'Approve'</a>			Resolved

## From 1.5.0 to 1.5.1

24 April 2008

This release contains bug fixes.

Atlassian JIRA (9 issues)				
Key	Summary	Pr		Status
<a href="#">CRUC-364</a>	<a href="#">Starting Crucible 1.5 Spits Lots of Errors Out</a>			Closed
<a href="#">CRUC-367</a>	<a href="#">LOC exceptions filling Debug log to the tune of 3-6GB per day</a>			Closed
<a href="#">CRUC-353</a>	<a href="#">Crucible seems not to be able to handle uploaded git patches</a>			Resolved
<a href="#">CRUC-350</a>	<a href="#">javax.servlet.ServletException</a>			Resolved
<a href="#">CRUC-304</a>	<a href="#">Add REST method to retrieve review items for a given review</a>			Closed
<a href="#">CRUC-291</a>	<a href="#">NullPointerException thrown when invoking the getAllRevisionComments web service method</a>			Closed
<a href="#">CRUC-161</a>	<a href="#">Create document stating when a customer purchases Crucible that they do not need to have the latest version of FishEye</a>			Closed
<a href="#">CRUC-357</a>	<a href="#">Sparklines have wrong content type</a>			Resolved
<a href="#">CRUC-333</a>	<a href="#">In IE, "view diff to latest" partially obscured</a>			Closed



## Crucible 1.6 Release Notes

This page last changed on Sep 23, 2008 by [edawson](#).  
23 September 2008

# Atlassian presents Crucible 1.6

Crucible release 1.6 makes it easier to review content that is not in [FishEye](#). Furthermore, Crucible 1.6 can be deployed without FishEye for the first time. Through Crucible's new 'Light SCM' plugins, you can include content in reviews that are not associated with FishEye or even a source control repository. For example, you can review pages directly from [Confluence](#), files on any file system connected to the machine FishEye is running on, and Subversion repositories not connected to FishEye. Crucible now has better support for uploading files for pre-commit review, in addition to the existing support for patches.

### Highlights of this release:

- [Support for Non-FishEye Repositories](#)
- [Confluence Page Reviews](#)
- [Shared File System Repositories](#)
- [Enhanced Pre-commit Reviews & Image Support](#)
- [Multiple Admin Users](#)
- [Expanded API](#)
- [Plus numerous improvements and bug fixes](#)



### Upgrading to Crucible 1.6

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

## Highlights of Crucible 1.6

1

### Support for Non-FishEye Repositories

Crucible can now be deployed as a stand-alone application for the first time. With Crucible 1.6, you no longer need a FishEye license or even a source-control repository. Crucible's new Light SCM plugin infrastructure already supports [Confluence](#), [server file systems](#), and Subversion repository types. We will be adding GIT and ClearCase in the near future. The Light SCM interface is public and the shipped plugins are open source. As a result, you can extend these plugins or even write your own — great news for plugin developers.

2

### Confluence Page Reviews

Crucible 1.6 allows you to select Atlassian Confluence as a source of material for reviews. In this way, you can use Crucible to easily review the

Wiki Markup of pages created in Confluence. [Read more](#).

Screenshot: Confluence Page Reviews in Crucible




The screenshot shows a Confluence page with a review comment by Edwin Dawson. The comment asks if the terminology is correct in the context of a repository. The page content includes a panel with a title 'Repository List' and a list of repository types. The review comment is highlighted in blue.

3

## Shared File System Repositories

You can create a 'repository' for a local or remote directory on the server file system. Teams that are managing documents through a shared file system instead of a source control system can still benefit from peer reviews. [Read more](#).

Screenshot: Crucible Reviews on the File System



The screenshot shows a Crucible review window for a file named 'fisheye/config.xml'. The review comment by Matt Quail asks if the number of threads to allocate is wise. The code snippet shows a configuration for a web server with a max-threads value of 20.

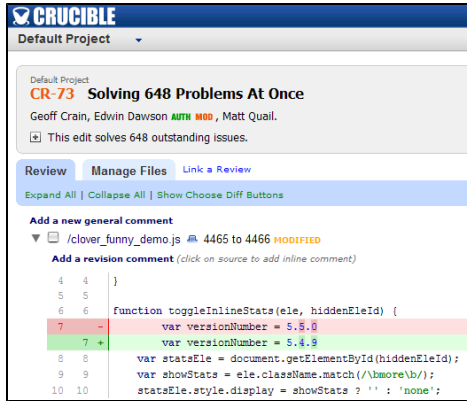
4

## Enhanced Pre-commit Reviews & Image Support

In addition to Crucible's patch support, 1.6 enables any file to be uploaded for review. The new upload functionality enables two files to be uploaded and compared in the review window, with diff highlighting. [Read more](#).

Crucible now supports before and after inline image previews.

Screenshot: Crucible Review of Uploaded Files



The screenshot shows a Crucible review window for a file named 'clover\_funny\_demo.js'. The review comment by Geoff Crain, Edwin Dawson, and Matt Quail asks if the edit solves 648 outstanding issues. The code snippet shows a function toggleInlineStats that updates the version number and shows the stats.

5

## Multiple Admin Users

Crucible now allows the Administrator to grant other users administration privileges. Admin Users can be individually assigned or given privileges through local or remote directory groups. [Read more.](#)











6

























## Expanded API





















The Crucible API now allows programmable review creation, along with a host of other additions. [Read more.](#)












7

## Plus numerous improvements and bug fixes























<a href="#">Atlassian JIRA</a> (104 issues)			
Key	Summary	Pr	Status
<a href="#">CRUC-661</a>	<a href="#">On the Crucible/Fisheye dashboard Projects are displayed even if the user has no access to enter those projects (e.g. anonymous users).</a>		 Resolved
<a href="#">CRUC-646</a>	<a href="#">Disabling and reenabling a plugin gives 'javax.el.ELException: org.springframework.osgi.service.importer.ServiceP service proxy has been destroyed'</a>		 Resolved
<a href="#">CRUC-608</a>	<a href="#">Show a list of LightSCM repositories in admin</a>		 Closed
<a href="#">CRUC-539</a>	<a href="#">Serious: "Next comment" does not always work right</a>		 Closed
<a href="#">CRUC-227</a>	<a href="#">Cannot delete projects</a>		 Resolved

<a href="#">CRUC-663</a>	<a href="#">blank page when a non creator/author/moderator tries to preview a draft review</a>			Resolved
<a href="#">CRUC-659</a>	<a href="#">Email notifications should be more easily threaded by mail clients</a>			Resolved
<a href="#">CRUC-654</a>	<a href="#">improve webwork/jsp error handling</a>			Resolved
<a href="#">CRUC-647</a>	<a href="#">improve ReviewManager.countStatesOn query</a>			Resolved
<a href="#">CRUC-645</a>	<a href="#">Optimise DB queries</a>			Resolved
<a href="#">CRUC-635</a>	<a href="#">Can't save changes to statement of objectives when at .../{id}/confirmApprove</a>			Resolved
<a href="#">CRUC-634</a>	<a href="#">ReviewData should not include repoName</a>			Resolved
<a href="#">CRUC-632</a>	<a href="#">Help text not updated after saving a review...</a>			Resolved
<a href="#">CRUC-623</a>	<a href="#">SvnChangeLog.Browser.listChanges() in LiteSVN fails when processing revisions that are above the repo's configured base path</a>			Closed
<a href="#">CRUC-622</a>	<a href="#">Improve performance of querying for review details</a>			Resolved
<a href="#">CRUC-615</a>	<a href="#">Make sure everything has a help link</a>			Resolved
<a href="#">CRUC-610</a>	<a href="#">Removing revisions from review that have draft comments</a>			Resolved



























	<a href="#">fails without an explanation</a>			
<a href="#">CRUC-602</a>	<a href="#">Improve the efficiency of checking whether a review has been completed by all reviewers</a>			Resolved
<a href="#">CRUC-600</a>	<a href="#">REST API should return error when non-existent filter used</a>			Resolved
<a href="#">CRUC-598</a>	<a href="#">Emails always include a "Summary" section</a>			Closed
<a href="#">CRUC-597</a>	<a href="#">Extend the Remote API to allows reviews to be deleted</a>			Resolved
<a href="#">CRUC-594</a>	<a href="#">Review description preview on dashboard swallows new lines from the review description</a>			Resolved
<a href="#">CRUC-591</a>	<a href="#">renaming a confluence repository adds a new one instead of editing</a>			Resolved
<a href="#">CRUC-589</a>	<a href="#">Change revisionData.getAdded/Removed lines an Integer</a>			Resolved
<a href="#">CRUC-588</a>	<a href="#">Search Subtab in Review-&gt;Manage Files throws PropertyNotFoundException for all searches</a>			Resolved
<a href="#">CRUC-586</a>	<a href="#">Remove Delete option from project drop down</a>			Closed
<a href="#">CRUC-579</a>	<a href="#">Changing repository</a>			Resolved

	when certain review tabs are selected causes NPE/ bad redirect			
<a href="#">CRUC-566</a>	lightscm package should not be "fisheye"	↑		Resolved
<a href="#">CRUC-565</a>	need get bulk version of getRevisionData() in SCMRepository	↑		Resolved
<a href="#">CRUC-563</a>	Toggle side-by-side and show per frx diff options on review page	↑		Resolved
<a href="#">CRUC-562</a>	REST API: createReview does not set Moderator and author correctly	↑		Resolved
<a href="#">CRUC-561</a>	anchors to some comment types don't work	↑		Closed
<a href="#">CRUC-560</a>	REST-API: provide comit type information - esp. deleted file status	↑		Resolved
<a href="#">CRUC-556</a>	Next/ previous comment arrows do not jump between file and general comments	↑		Closed
<a href="#">CRUC-554</a>	Upgrade to non-beta atlassian-plugins	↑		Resolved
<a href="#">CRUC-551</a>	Confluence Light SCM plugin dependencies belong in plugin	↑		Resolved
<a href="#">CRUC-549</a>	Adjust IFRAME size when the admin page is resized	↑		Closed
<a href="#">CRUC-547</a>		↑		Closed

	<a href="#">Make Repository source check if the engine is available.</a>			
<a href="#">CRUC-545</a>	<a href="#">NPE Closing Review</a>	↑	→	Resolved
<a href="#">CRUC-542</a>	<a href="#">Use content manager to retrieve detailed file revisions</a>	↑	→	Resolved
<a href="#">CRUC-526</a>	<a href="#">Space character in installation path to Crucible disables remote API</a>	↑	→	Resolved
<a href="#">CRUC-519</a>	<a href="#">Added files display "no change"</a>	↑	→	Closed
<a href="#">CRUC-518</a>	<a href="#">Crucible does not seem to be incrementally updating the FishEye cache</a>	↑	→	Closed
<a href="#">CRUC-516</a>	<a href="#">REST API: Return detailed information on review in one call - items, comments, reviewers, available actions</a>	↑	→	Resolved
<a href="#">CRUC-511</a>	<a href="#">Added files when stored show "No Change"</a>	↑	→	Closed
<a href="#">CRUC-507</a>	<a href="#">Confluence: implement change set paging</a>	↑	→	Resolved
<a href="#">CRUC-505</a>	<a href="#">Add "Search for Review" functionality to Crucible remote API</a>	↑	→	Resolved
<a href="#">CRUC-504</a>	<a href="#">Confluence: get revision comments</a>	↑	→	Closed
<a href="#">CRUC-499</a>	<a href="#">Create separate API jar</a>	↑	→	Resolved
<a href="#">CRUC-496</a>	<a href="#">Document store revisions with review</a>	↑	→	Closed






	<a href="#">CRUC-494</a>	<a href="#">Don't make source type visible in URLs or elsewhere</a>			Resolved
	<a href="#">CRUC-488</a>	<a href="#">clicking summarise without clicking close makes the review 'disappear' from the workflow</a>			Closed
	<a href="#">CRUC-484</a>	<a href="#">you can link a review to itself</a>			Closed
	<a href="#">CRUC-478</a>	<a href="#">Ability to specify that Branches are not supported</a>			Resolved
	<a href="#">CRUC-472</a>	<a href="#">Changing tab while editing review deletes details</a>			Resolved
	<a href="#">CRUC-461</a>	<a href="#">Line comments are sometimes rendered twice</a>			Closed
	<a href="#">CRUC-456</a>	<a href="#">crucible doesn't create default permission scheme when creating a blank db</a>			Closed
	<a href="#">CRUC-455</a>	<a href="#">creating a review from a changeset with a non-existent repo keeps redirecting to the login screen</a>			Closed
	<a href="#">CRUC-453</a>	<a href="#">Allow adding a screenshot as an attachment to a review</a>			Closed
	<a href="#">CRUC-439</a>	<a href="#">Invite to review</a>			Closed
	<a href="#">CRUC-438</a>	<a href="#">Not sending email notifications after</a>			Closed



	<a href="#">Crucible and Fisheye upgrade</a>			
<a href="#">CRUC-436</a>	<a href="#">create 'remove all revision' link for all tabs under 'manage files'</a>			Closed
<a href="#">CRUC-433</a>	<a href="#">REST API: Get review list based on predefined and custom filters</a>			Resolved
<a href="#">CRUC-422</a>	<a href="#">Create Schema Upgrade to remove foreign key constraint as reqd. by delete project work</a>			Closed
<a href="#">CRUC-407</a>	<a href="#">Minimal OSGi Infrastructure</a>			Resolved
<a href="#">CRUC-404</a>	<a href="#">Turn FE Off When only Crucible licence is present</a>			Resolved
<a href="#">CRUC-402</a>	<a href="#">Provide Plugin Authors with somewhere to store properties</a>			Resolved
<a href="#">CRUC-400</a>	<a href="#">Add Configuration UI to SVN Plugin, and Polish</a>			Resolved
<a href="#">CRUC-397</a>	<a href="#">Stored FRX Create UI</a>			Resolved
<a href="#">CRUC-396</a>	<a href="#">Stored FRX Data Impl</a>			Closed
<a href="#">CRUC-394</a>	<a href="#">Add Light SCM Revisions to Reviews</a>			Resolved
<a href="#">CRUC-392</a>	<a href="#">Filesystem Light SCM Plugin</a>			Resolved
<a href="#">CRUC-391</a>	<a href="#">Default Repository for a Project can be a Light SCM repo</a>			Resolved
<a href="#">CRUC-389</a>	<a href="#">Light SCM Plugin</a>			Resolved

	<a href="#">instances appear in source/repository dropdowns</a>			
<a href="#">CRUC-388</a>	<a href="#">Subversion LSCM plugin</a>			Resolved
<a href="#">CRUC-387</a>	<a href="#">Confluence LSCM Plugin</a>			Resolved
<a href="#">CRUC-386</a>	<a href="#">Modify File Browser to Use Light SCM Plugins</a>			Resolved
<a href="#">CRUC-385</a>	<a href="#">Modify Changeset Browser to use Light SCM Plugins</a>			Resolved
<a href="#">CRUC-384</a>	<a href="#">Create Light SCM Module Type</a>			Resolved
<a href="#">CRUC-381</a>	<a href="#">Admin Pages for Plugins</a>			Closed
<a href="#">CRUC-373</a>	<a href="#">Add General Comment via REST API</a>			Resolved
<a href="#">CRUC-362</a>	<a href="#">Return replies to comments via API</a>			Resolved
<a href="#">CRUC-361</a>	<a href="#">Project RSS feed</a>			Closed
<a href="#">CRUC-349</a>	<a href="#">REST method / rest-service/reviews-v1/CR-1/comments returns HTTP 500 error</a>			Resolved
<a href="#">CRUC-303</a>	<a href="#">Add REST method to retrieve all reviews which involve a given file</a>			Resolved
<a href="#">CRUC-302</a>	<a href="#">Add REST methods to allow review creation</a>			Resolved
<a href="#">CRUC-297</a>	<a href="#">Upload files for review (not patches)</a>			Closed
<a href="#">CRUC-292</a>	<a href="#">javax.xml.bind.JAXBException generated when invoking the getGeneralComments</a>			Resolved

	<a href="#">web service method</a>			
<a href="#">CRUC-244</a>	Add revisions not diffs to a review	↑	✓	Closed
<a href="#">CRUC-186</a>	Allow an abandoned review to be deleted	↑	➡	Resolved
<a href="#">CRUC-150</a>	Allow the email address in the from section of the notification emails to be the user's actual email address	↑	➡	Resolved
<a href="#">CRUC-18</a>	Load All Users from Crowd/ LDAP/etc	↑	✓	Closed
<a href="#">CRUC-662</a>	Updating general comments in the REST api has a permission flaw	↓	➡	Resolved
<a href="#">CRUC-574</a>	Allowed Review Participants left blank means what?	↓	✓	Closed
<a href="#">CRUC-548</a>	REST API: Handle allowReviewersToJoin flag on review	↓	➡	Resolved
<a href="#">CRUC-540</a>	make metrics-config.xsd available online	↓	✓	Closed
<a href="#">CRUC-487</a>	Page title says "dead" for a review that has been abandoned	↓	➡	Resolved
<a href="#">CRUC-479</a>	A comment which was a defect, but is no longer, still shows the defect attributes	↓	✓	Closed
<a href="#">CRUC-457</a>	show authorname next to	↓	➡	Resolved

	revision in revision dropdowns			
<a href="#">CRUC-401</a>	Real Admin Users	↓		Resolved
<a href="#">CRUC-399</a>	Optionally Store Files for all FileRevisions	↓		Closed
<a href="#">CRUC-592</a>	Filter names used in Cru do not match the menus	↓		Resolved
<a href="#">CRUC-469</a>	Sort list of repositories alphabetically on "Add project" page	↓		Closed
<a href="#">CRUC-419</a>	Typo in email "reviewers are now complete"	↓		Resolved

## Crucible Release Summary

---

This page last changed on Jun 09, 2008 by [jmcmillan](#).

### 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 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](mailto: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 Aug 07, 2008 by [edawson](#).

### Crucible Knowledge Base

Answers to commonly raised questions about configuring and using Crucible.

- [Do I need a FishEye licence to run Crucible?](#)
- [Troubleshooting](#)
- [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 knowledge base](#)

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

## Do I need a FishEye licence to run Crucible?


---

This page last changed on May 19, 2008 by [jmcmillan](#).

You need a FishEye licence to run Crucible. Crucible includes FishEye, because FishEye comprises important parts of Crucible's functionality.

FishEye and Crucible are developed concurrently and have dependencies on a particular version number. For this reason, all purchasers of Crucible need a working FishEye license whose maintenance covers the version of Crucible they will be installing. You can use an existing FishEye license you already have, or you can buy a current FishEye license (upgrade/renew) at the same time. If you have an old version of FishEye, and you purchase a Crucible license, then you need to turn off your old FishEye server and start the new Crucible server, putting both licenses into this new Crucible server.

- With one FishEye license and one Crucible license, you cannot run Crucible plus your old FishEye server.
- If your old FishEye license has expired maintenance, then it probably won't work in the new Crucible.

 You can use an old FishEye license and your new Crucible licence so long as the maintenance period on your old FishEye license overlaps with the maintenance period of the version of Crucible you are trying to run.

For more information on acquiring a current licence, email [sales@atlassian.com](mailto:sales@atlassian.com).

## 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](mailto:rosie@atlassian.com).

- [0. Getting Started with Crucible](#)
- [1. Using the Dashboard](#)
  - [Using the Project 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](#)
      - [Removing Reviewers From An Active Review](#)
    - [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](#)
  - [2.5 Deleting an Abandoned Review](#)
- [3. Changing your User Profile](#)

## 0. Getting Started with Crucible

This page last changed on Jan 29, 2008 by [edawson](#).

Crucible is a flexible application that caters for a wide range of work styles and team sizes. There are a number of different ways in which you can use Crucible for code reviews.

This page contains a simplified set of instructions for executing a one-to-one review with just 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 acts as the reviewer.

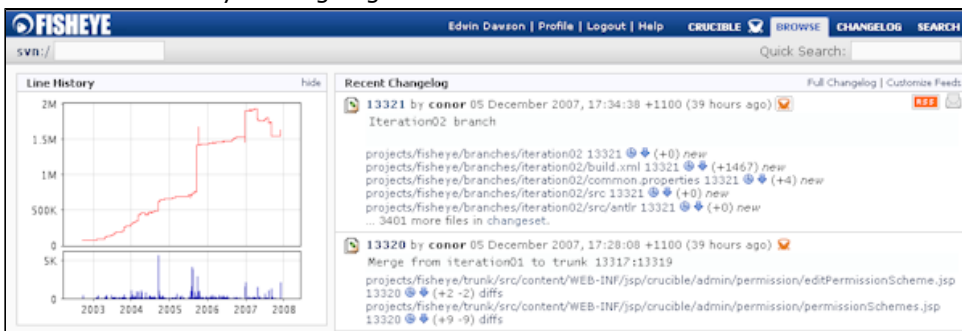
- [1. The Author Starts the Review](#)
- [2. The Reviewer Comments on the Code](#)
- [3. The Author Responds to the Comments](#)
- [4. The Author Closes the Review](#)

For instructions on workflow with more than two people, see [2. Requesting and Conducting a Review](#).

### 1. The Author Starts the Review

To begin, the code author sets up the review. There are a [number of ways to do this](#). In this example, the author starts from the FishEye Changelog screen:

Screenshot: FishEye changelog

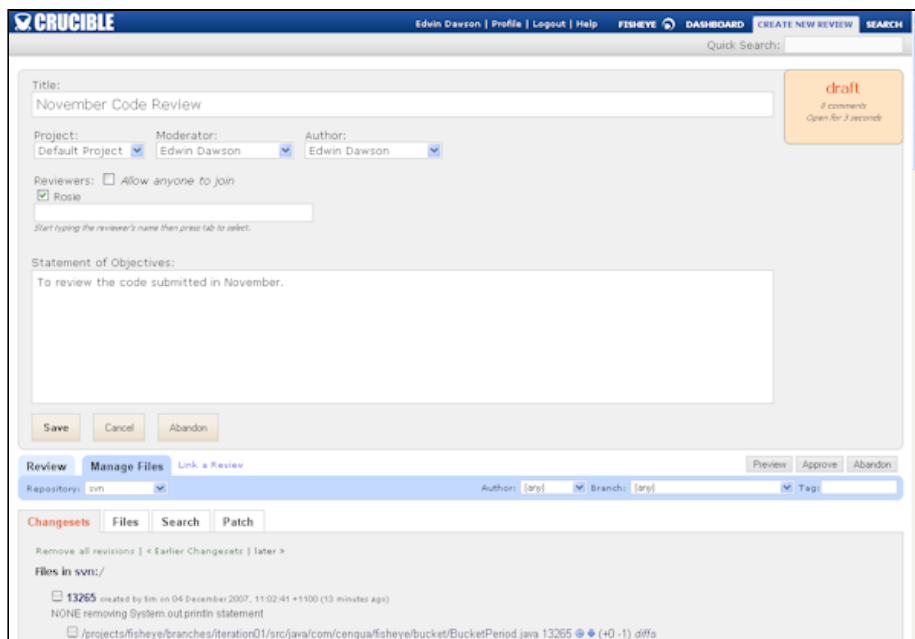




1. From the FishEye Changelog screen, the author clicks the Crucible icon



next to the name of the changeset they would like reviewed, then click their name on the following screen. The Crucible Create New Review screen appears, where the author can create and issue the review.

Screenshot: Crucible create new review



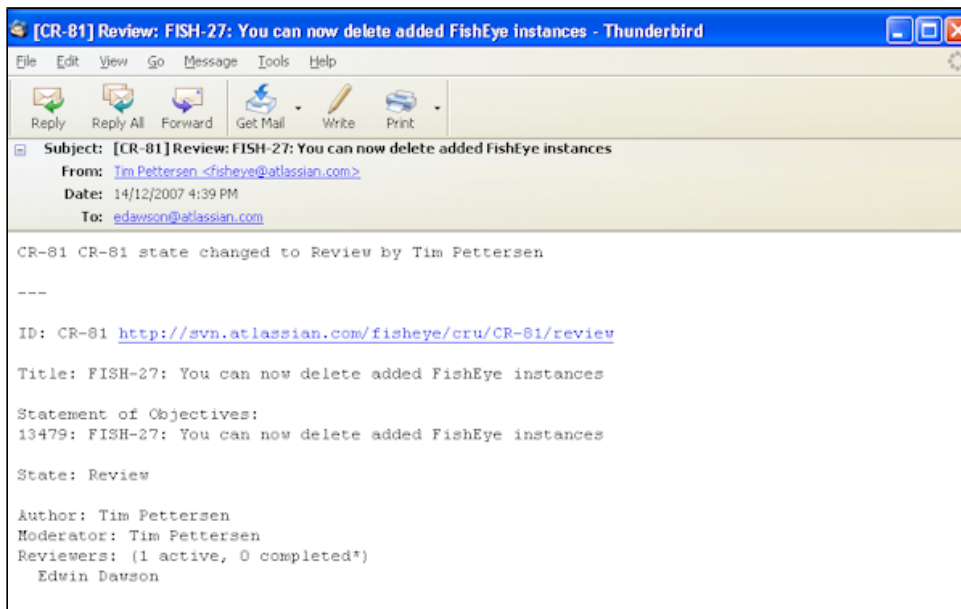
2. On the Crucible Create New Review screen, the author does the following:
  - a. Types a name for the review in the 'Title' field.
  - b. Checks that the author's name shows in both the 'Moderator' and 'Author' fields.
  - c. Chooses the appropriate reviewer.
  - d. In the 'Statement of Objectives' box, types a short description of the review objectives (if required).
  - e. Clicks the  button to finish the review creation (Note: the  button here will make the review a draft review and store it under the drafts folder).

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: Crucible review notification email

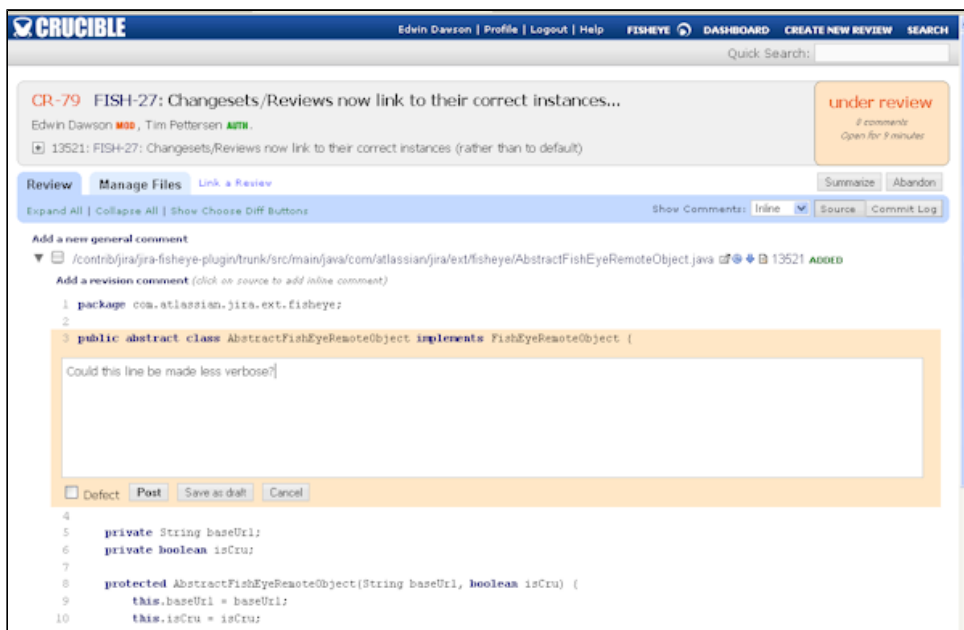


1. When the reviewer clicks the URL in the notification email, the 'Crucible Under Review screen' opens.
- Screenshot: Crucible under review



2. On the Crucible Under Review screen, the code changes under review are displayed, along with a list of participants. The reviewer clicks the arrow next to a filename to expand the code for in-line reviewing, reads the changes and simply clicks on any line to enter a comment on that line (multiple lines can be selected by clicking and dragging).

Screenshot: Making a comment in Crucible



The reviewer clicks the 'Post' button when each comment is finished. The reviewer repeats this process for all files in the review. Reviewers can leave the session and resume it later — their work is automatically saved. When the reviewer has finished their code review work, they click the

Complete

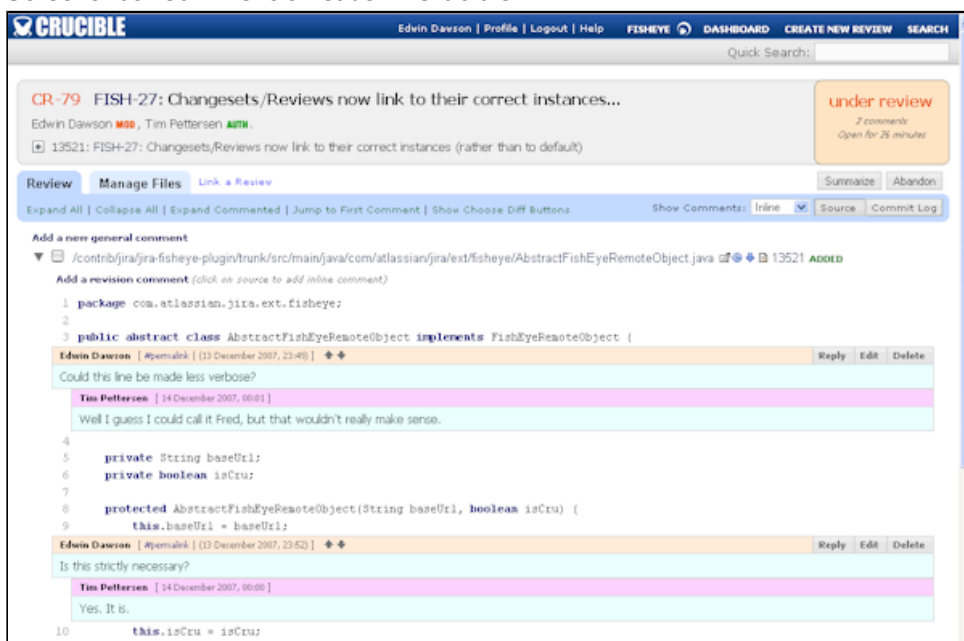
button.

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

### 3. The Author Responds to the Comments

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

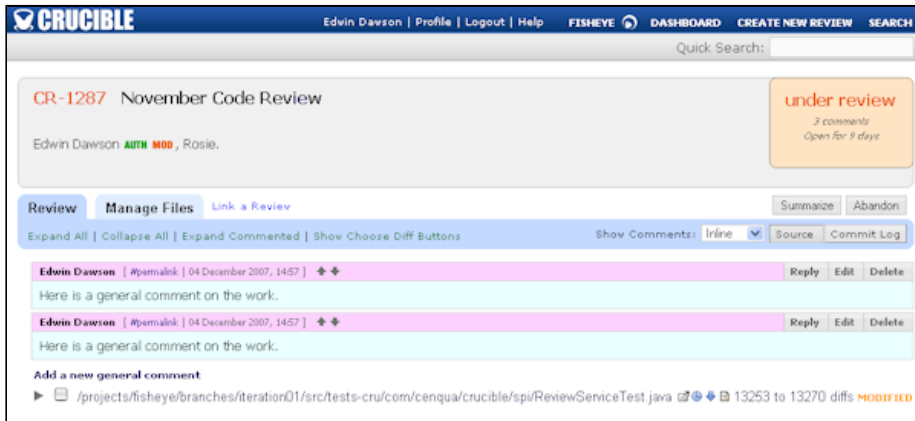
Screenshot: Comment threads in Crucible




#### 4. The Author Closes the Review

1. When all reviewers have Completed their reviews, the author/moderator is notified via email. The author/moderator will then add any final comments, summarize and close the review.
2. The author/moderator clicks the link in the notification email, returning to the Crucible Under Review screen.

Screenshot: Crucible under review




3. On the Crucible Under Review screen, the author/moderator will add final comments, then click

the  button. (if not all reviewers are finished, the author/moderator will be prompted to cancel or 'Summarize anyway'.) The Crucible Summary screen appears.

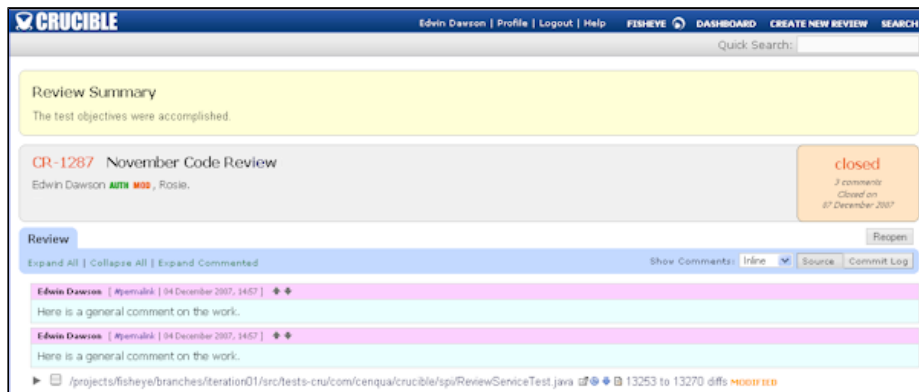
Screenshot: Summarizing a review in Crucible



4. On the Crucible Summary screen, the author/moderator enters a summary of the review's results, then clicks . This closes the review, signalling the end of work. The closed review screen will load, displaying the summary and archiving the completed review as read-only.

Screenshot: 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 [2. Requesting and Conducting a Review](#).

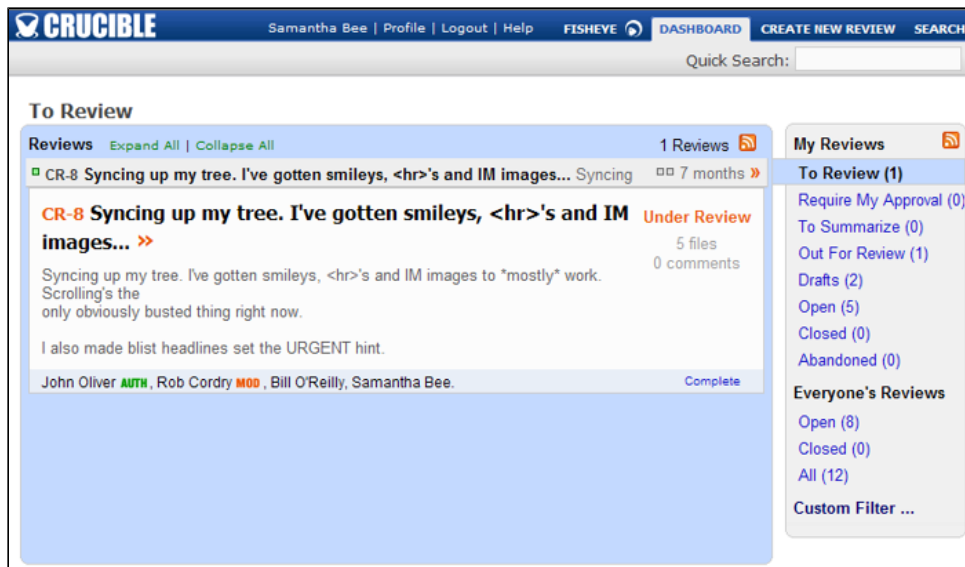
## 1. Using the Dashboard

This page last changed on Aug 25, 2008 by [edawson](#).

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

- [Managing Reviews](#)
  - [Side Panel](#)
  - [Custom Filters](#)
  - [Review State](#)
  - [Search Box \(Quick Search\)](#)
- [Changing your User Profile](#)
- [Subscribing to an RSS Feed](#)

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 <a href="#">moderator</a> for these reviews and needs to <a href="#">approve</a> them.
To Summarize	The user has been assigned the role of <a href="#">moderator</a> for these reviews and needs to <a href="#">summarise and close</a> 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' <a href="#">states</a> .
Open	These reviews have been created by the user and are at a <a href="#">state</a> 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.

#### Custom Filters

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

Title	Find reviews by searching for words within the title.
Project	Find reviews under a particular project.
Author	Find reviews moderated by a particular <a href="#">authors</a> .
Moderator	Find reviews moderated by a particular <a href="#">moderators</a> .
Creator	Find reviews created by a particular <a href="#">creator</a> .
Reviewer	Find reviews that are reviewed by a particular <a href="#">reviewer</a> . 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

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

Draft	Reviews that are still in 'Review' <a href="#">state</a> .
Pending Approval	Reviews that have been moved out of 'Draft' <a href="#">state</a> and are now waiting for the <a href="#">moderator</a> to <a href="#">approve</a> .
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 <a href="#">moderator</a> 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 like this:



Click an icon to set up an RSS feed containing reviews as follows:



panel - reviews that you have had interaction with.

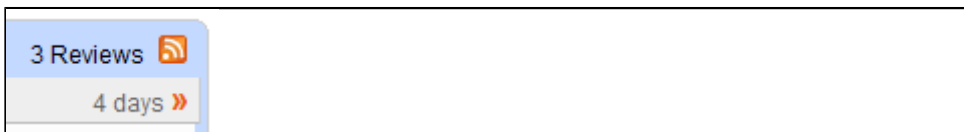


panel - all reviews.

Screenshot: The RSS Icon in Crucible

Icon on the right-hand

Icon on the left-hand



## Using the Project Dashboard

This page last changed on Apr 14, 2008 by [edawson](#).

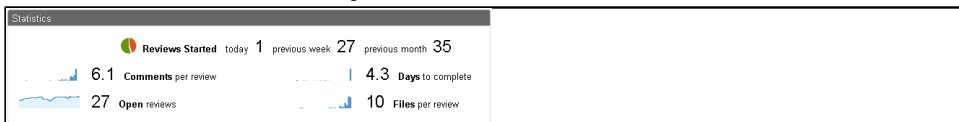
The Project Dashboard in Crucible allows you to view a list of reviews belonging to a project along with relevant statistics and charts, aggregated into one view. It contains the following elements:

- [Project Statistics](#)
- [List of Open Reviews](#)
- [Side Panel](#)
- [Comment Volume chart](#)
- [Defect Metrics chart](#)
- [Open Reviews chart](#)
- [Open Review Age chart](#)
- [Quick Search](#)

### Project Statistics

The Project Statistics chart shows information about the frequency of reviews, the average comments per review, the average time taken to complete reviews, the total number of reviews and reviews that are currently open in this project.

### Screenshot: Crucible Per-Project Statistics



### List of Open Reviews

In the centre of the Project Dashboard, the project's reviews are listed according to the option selected on the left-hand side panel.

### Side Panel

The panel on the left-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 centre panel.

Draft	These are reviews created by the user that have not yet been moved to the 'Approval' or the 'Require Approval' <a href="#">states</a> .
Open	These reviews have been created by the user and are at a <a href="#">state</a> other than 'Closed'.
Closed	These are reviews that the user has been involved in and are now closed.
All	All reviews.

### Comment Volume chart

The Comment Volume chart shows a visual representation of defects and comments on the project, with a bar graph.

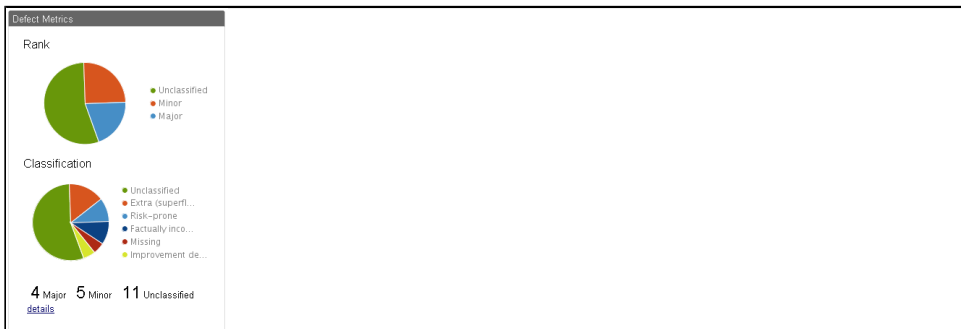
### Screenshot: Crucible Comment Volume chart



### Defect Metrics chart

The Defect Metrics chart shows the breakdown of defect rank and classification in this project, with pie charts. You can click through to more information by clicking the 'details' link.

#### Screenshot: Crucible Defect Metrics chart



### Open Reviews chart

The Open Reviews chart shows the number of open reviews over time in this project, with an area graph.

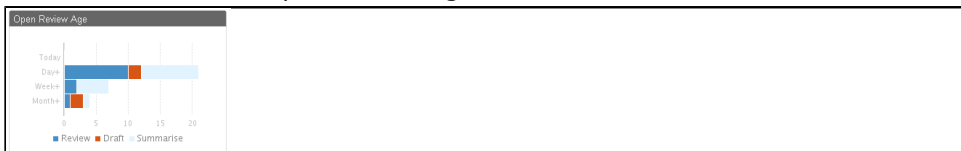
#### Screenshot: Crucible Open Reviews chart



### Open Review Age chart

The Open Review Age shows a breakdown of the age of charts in this project.

#### Screenshot: Crucible Open Review Age chart



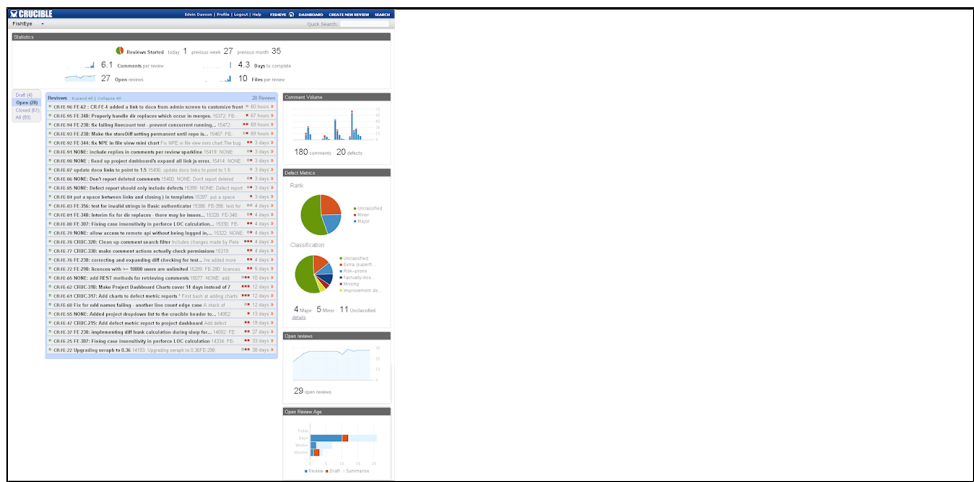
### Quick Search

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-xxx
- Contents of Title
- Contents of Statement of Objective
- Contents of a comment

#### Screenshot: Crucible Project Dashboard



## 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](#)
- [2.5 Deleting an Abandoned Review](#)



## 2.1 Creating a Review

---

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



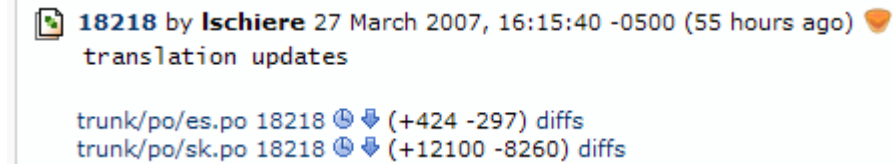
Note that only people with the ['Create' permission](#) can create a review.

### 2.1.1 Creating a Review from FishEye

This page last changed on Jan 02, 2008 by [edawson](#).

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



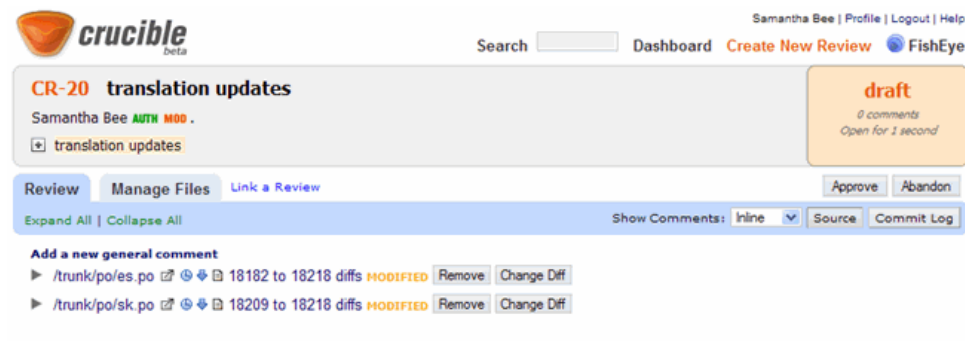
When you click the icon a new draft review will be created, including the following information:

- The option to select a project, if there are multiple.
- 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 option to add reviewers, by typing their names into the 'Reviewers' field selecting their names as they appear in the results. (If there is a small number of reviewers in the system, they'll appear as a list.)
- An option to check the box 'Reviewers: allow anyone to join' if you want that to be the default setting.
- The commit log message is used as both the Title and [Statement of Objective](#). You may also enter your own, instead.

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](mailto: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](mailto:rosie@atlassian.com).

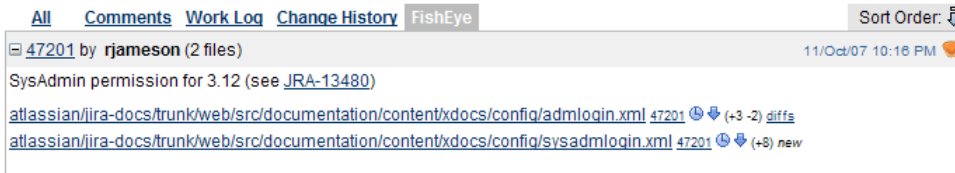
To create a review from within [JIRA](#), click the small Crucible icon



next to the required

changeset on the 'FishEye' tab.

Screenshot: Adding a Review from within JIRA



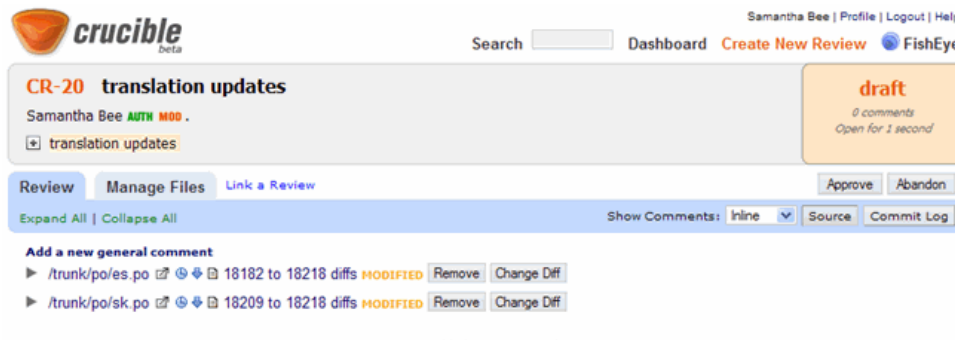
When you click the icon, you will be prompted to select the relevant [project](#) (if more than one project exists) in which to create your review. A new draft review will then be created, including the following information:

- 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](mailto: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%2F&repob%2F7583
```

The parameters are as follows:

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

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

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

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

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

## 2.1.5 Selecting the Files for the Review

This page last changed on Sep 23, 2008 by [edawson](#).

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.

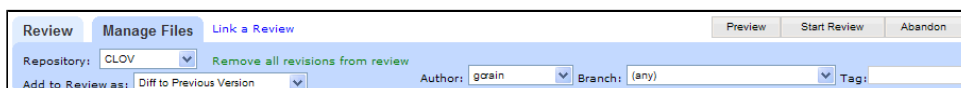
On this page:

- [Constraints on Manage Files Tab](#)
- [Choosing the format of the review](#)
- [Changeset Tab](#)
  - [Understanding Changesets in Detail](#)
- [Files Tab](#)
- [Search Tab](#)
- [Patch Tab](#)
- [Upload Tab](#)

### Constraints on Manage Files Tab

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

Screenshot: Constraint options in the Manage Files tab



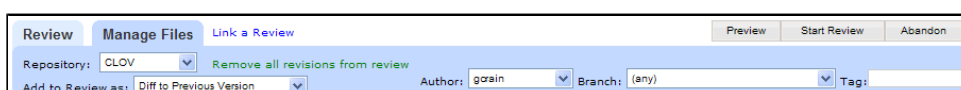
The screenshot shows the 'Manage Files' tab in a web interface. At the top, there are tabs for 'Review', 'Manage Files', and 'Link a Review'. Below these, there are several input fields and buttons. The 'Repository' field is set to 'CLOV'. There is a button 'Remove all revisions from review'. The 'Add to Review as:' dropdown is set to 'Diff to Previous Version'. The 'Author' field is set to 'gorain', the 'Branch' field is set to '(any)', and the 'Tag' field is empty. At the top right, there are buttons for 'Preview', 'Start Review', and 'Abandon'.

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 <a href="#">authors</a> 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.

### Choosing the format of the review

Before using the 'Changesets', 'Files' or 'Search' tabs, you can set the type of version comparison used in the review. Several options are listed along the top of the 'Manage Files' tab, in the Add to Review as: drop-down menu:

Screenshot: Review format options in the Manage Files tab



This screenshot is identical to the one above, showing the 'Manage Files' tab with the 'Add to Review as:' dropdown set to 'Diff to Previous Version'.

Diff to previous version

	This is the default behaviour, and how previous versions of Crucible worked by default.
Diff to last reviewed	This creates a review with a diff to the last reviewed changeset.
Diff to (a particular revision)	This allows you to specify the review to show the differences between two specific versions of a file. This is useful in various situations. For example, you may want to review all changes since you branched your project.
Whole file	The review content is not limited by changesets, showing the entire file contents.

## Changeset 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 from review' to remove them.

### Screenshot: Changesets tab



### Understanding Changesets in Detail

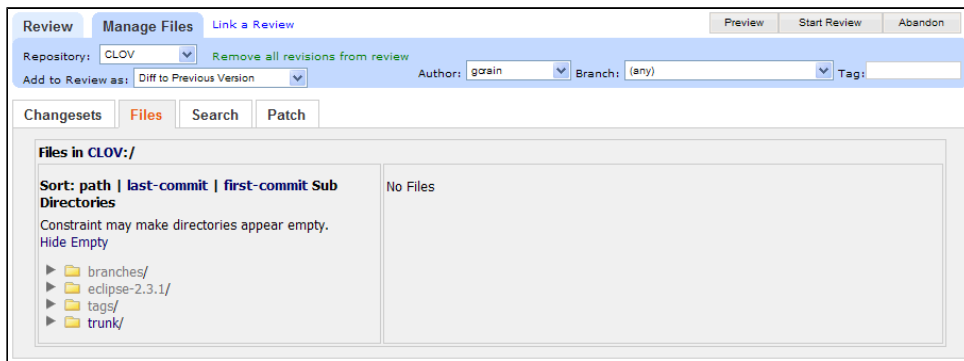
To help decide what files are to be placed under review, the [moderator](#) can click the icons next to the files to gain further information about them before they go out for review:

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



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 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

For a full explanation of the 'Patch Tab' functions, read about [creating a patch review](#).

## Upload Tab

You can upload additional files to be used in the review. The 'Upload' tab contains the following options to help you do this.

File to review	By clicking 'Browse', you can find the file you want to review.
File to diff with	By clicking 'Browse', you can find the file you want to compare against the first file.
Author username	Enter the author username into this field.
Comment	Here, you can put in an identifying comment about this particular piece of work.

Character Set (if any)

In the drop-down list, you can choose the character set being used. '[US-ASCII](#)' is the default.

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

**i** The triangular Delta symbol is used as short-hand for 'The file on the left is diffed with the file on the right'.

Screenshot: Accessing the Upload Tab

Review Manage Files Link a Review

Repository: CLOV

Add to Review as: Diff to Previous Version

Changesets Files Search Patch **Upload**

« Earlier Changesets | later »

**Files in CLOV:/**

☐ **19184** created by mstudman on 24 August 2008, 16:07:58 -0500 (7 hours ago)

CLOV-308: Stopping certain tests from removing the output of previously run tests.

Screenshot: Crucible's Upload Tab

Changesets Files Search Patch **Upload**

**File to review \***

Browse...

**File to diff with**

Browse...

**Author username**

edawson

**Comment**

**Character Set (if text)**

US-ASCII

Upload

No upload files

## 2.1.6 Adding Reviewers

This page last changed on Sep 23, 2008 by [edawson](#).

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

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

When adding reviewers, you can add registered users immediately. In addition, you can easily invite external users who do not yet have accounts in Crucible to take part.

On this page:

- [Adding registered users to a review](#)
- [Inviting non-registered users to a review](#)

### Adding registered users to a review

- 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.
- The list of 'Reviewers' appears next to the Author and Moderator fields on the review details screen, after they are added.
- If there are fewer than seven potential reviewers, they are shown with checkboxes next to each name.

Click the 'Save' button to save as a draft for later issue, or click the 'Start Review' button (bottom right-hand corner) to issue the review to the reviewers.

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'.

### Screenshot: Adding reviewers

### Inviting non-registered users to a review

You can invite users who do not yet have a Crucible account to take part in a review.




There are two prerequisites for inviting external users in Crucible:

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

To invite an external user to a review,

- a. Create a new review.
- b. On the 'Create New Review' screen, simply type the user's email address into the 'Invited Reviewers' field, then click 'Invite'.
- c. Click Save to save the draft review. The users are not sent any information at this time.

- d. When you click 'Start Review', this is when all email invites and notifications are sent out.
- e. The external user will receive an email address from the Crucible server, containing a special URL that they can visit.
- f. 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 log in username and password.)
- g. 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.

Screenshot: The Invite Anyone Field



The screenshot shows a web interface for inviting reviewers. It features a text input field with the placeholder text "test@test.com;" and an "Invite" button next to it. The input field is part of a larger form area.

## Removing Reviewers From An Active Review

This page last changed on Sep 11, 2008 by [edawson](#).

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

To remove a reviewer from an active review,

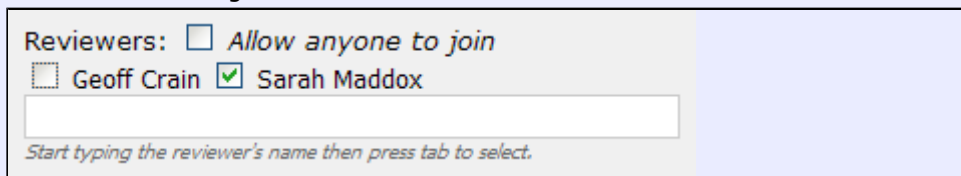
1. Log in to Crucible as the review creator or moderator.
2. Open the review in question.
3. Click on the list of reviewers, to enter 'Edit Mode'.


Screenshot: Clicking to edit review properties




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

Screenshot: Editing the list of reviewers in Crucible



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

 You cannot remove the review creator or moderator from the review.

## 2.1.7 Creating a Patch Review

This page last changed on Sep 23, 2008 by [edawson](#).

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

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.

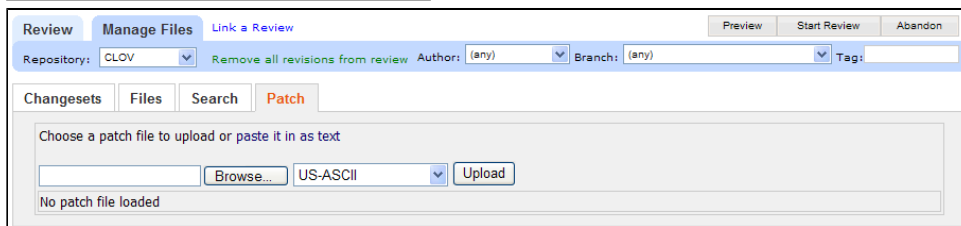
While instructions on this page cover only Subversion, CVS and Perforce repositories, Crucible can accept a patch file created from any version control system.

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](#)
  - [Creating a patch in Subversion with all lines of code](#)
  - [Creating a patch in Perforce with all lines of code](#)

## Loading a patch into Crucible

Screenshot: The Crucible Patch tab



Once you have generated the patch file, simply select the file in Crucible's Patch Tab and click 'Upload' to add it to your review. Of course, once the review is complete, it is up to the author to check in the changes from the patch that were under review.

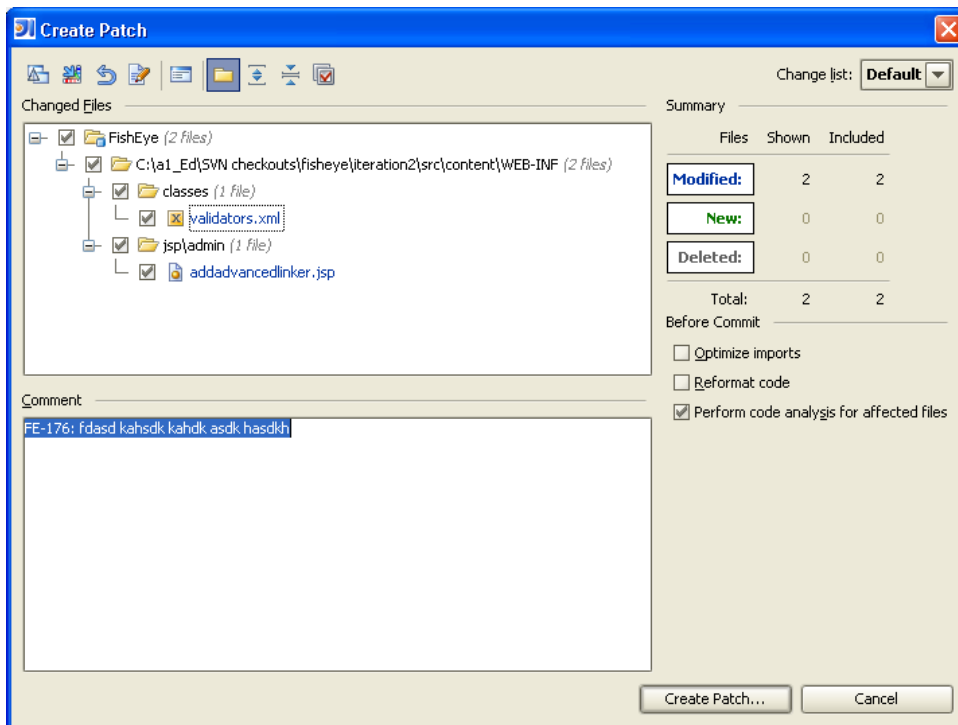
## Creating a patch file from your IDE

### Creating a patch file in IntelliJ IDEA 7.0

To create a patch file under IntelliJ IDEA, do the following:

Select a parent folder, subfolder 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



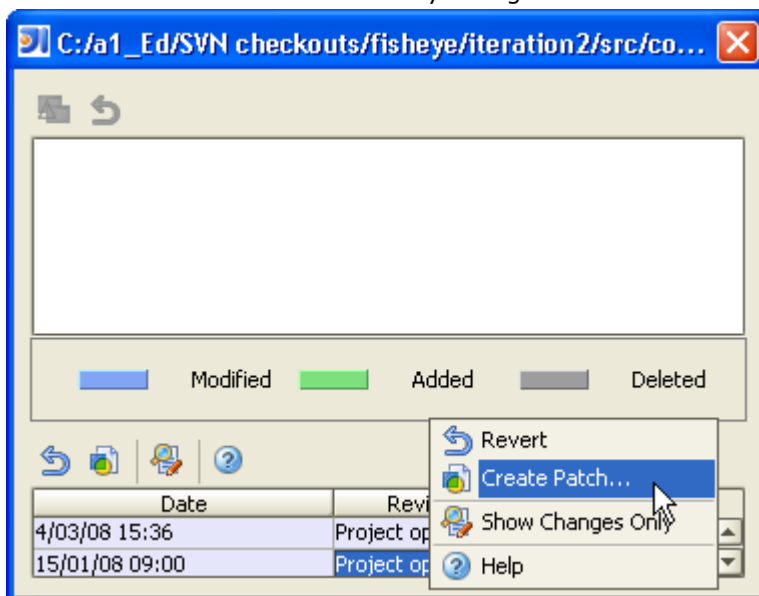
Click 'Create Patch'. Choose a location to save the patch file and click 'Ok'.

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:

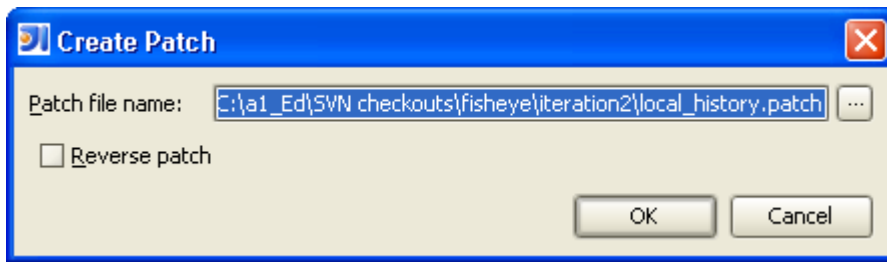
1. Select a parent folder, subfolder or file that you have altered in the Project tool window, right-click it and choose 'Local History' > 'Show History'.

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



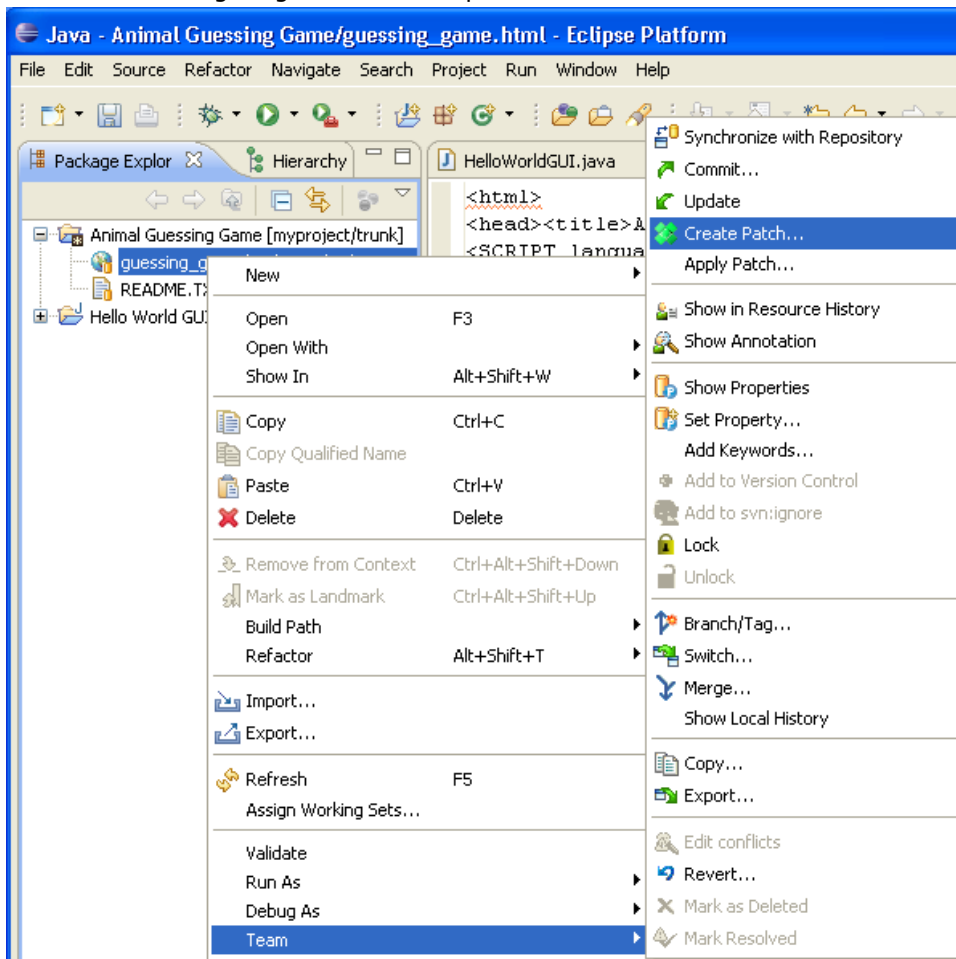
3. In the Create Patch dialog, choose a location for the patch file and a file name, then click 'OK'.

### 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'.

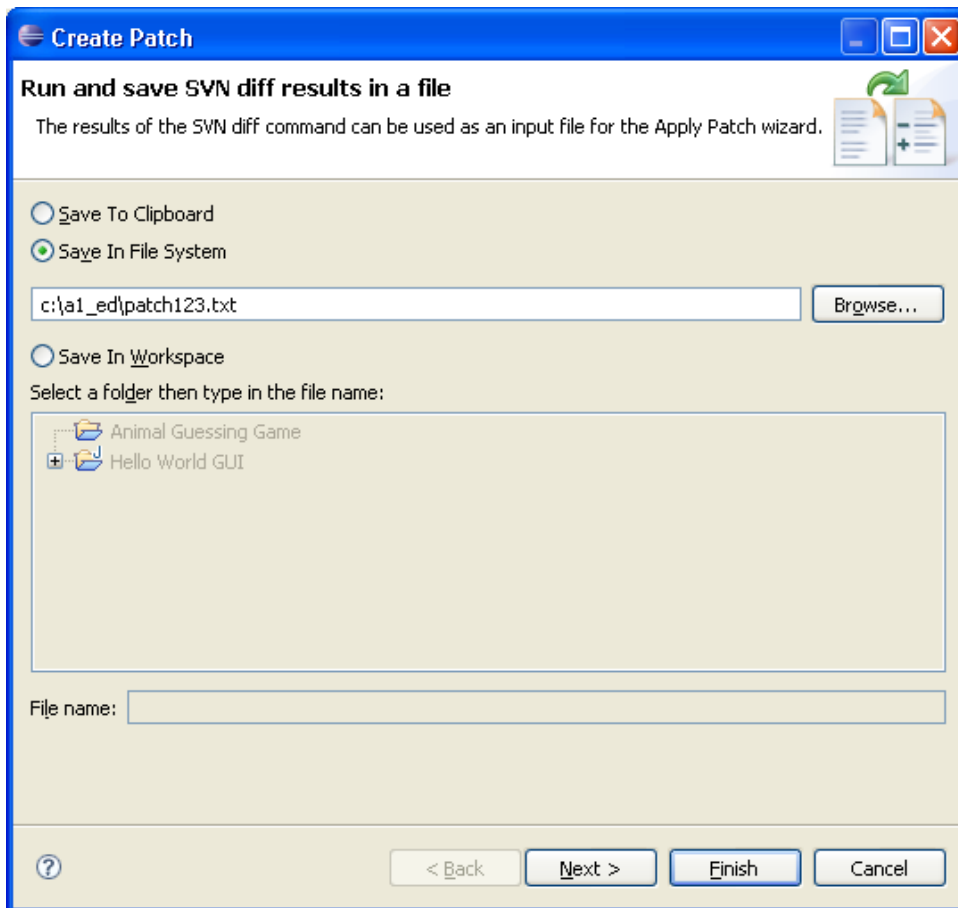
Screenshot: Instigating a Patch in Eclipse



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 `cv diff -Nu` command from your workspace. For example:

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

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

### Subversion patch creation via the command line

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

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



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

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

## Perforce patch creation via the command line

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

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

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

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

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

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

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


## Creating patches that include all lines of code

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

### Creating a patch in CVS with all lines of code

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

```
cv 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.


### Creating a patch in Subversion with all lines of code


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

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

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

## Creating a patch in Perforce with all lines of code

 Unfortunately, Perforce does not directly support creating patches that include all lines of code. A workaround is to checkout 'before' and 'after' versions of the file, and use GNU Diff to create a patch between the two files. That file could then be loaded into a Crucible review.

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

## 2.2 Issuing a Review

---

This page last changed on Feb 26, 2008 by [edawson](#).

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

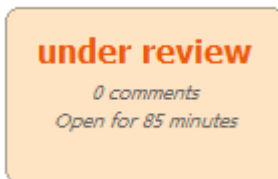
Once you have [selected the reviewers](#), the next stage is to notify the [reviewers](#) and the [author](#) (if different to the [moderator](#)) that they can start reviewing. The review has been in 'Draft' [state](#) until this point. Only the moderator has the permission to approve a review.

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 'Under Review'.

Screenshot: Under Review screen



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

## 2.3 Performing the Review

This page last changed on Apr 14, 2008 by [edawson](#).

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.



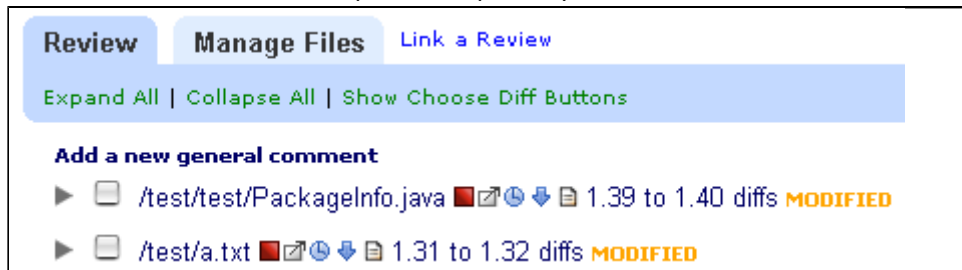
### Deciding what needs to be reviewed

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.

## When a file under review changes in the repository

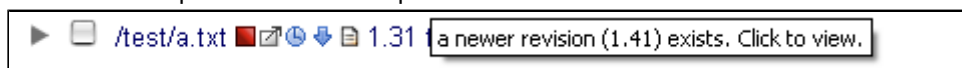
If a file in the repository changes during a review, Crucible will visually alert you by showing a small red square  next to the name of the file under review.

Screenshot: Visual Cue for Updated Repository Files



You can hover your mouse cursor over the file name or the square red icon to quickly see the revision number of the updated file.

Screenshot: Updated File Tool Tip



If you click the name of the file, the details will expand and a new hyperlink will appear below, called 'view diff to latest'. You can click that link to view diffs against the newer file.

Screenshot: Expanded File Details Showing New Revision Link



- [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:

**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 dont think this code is correct.

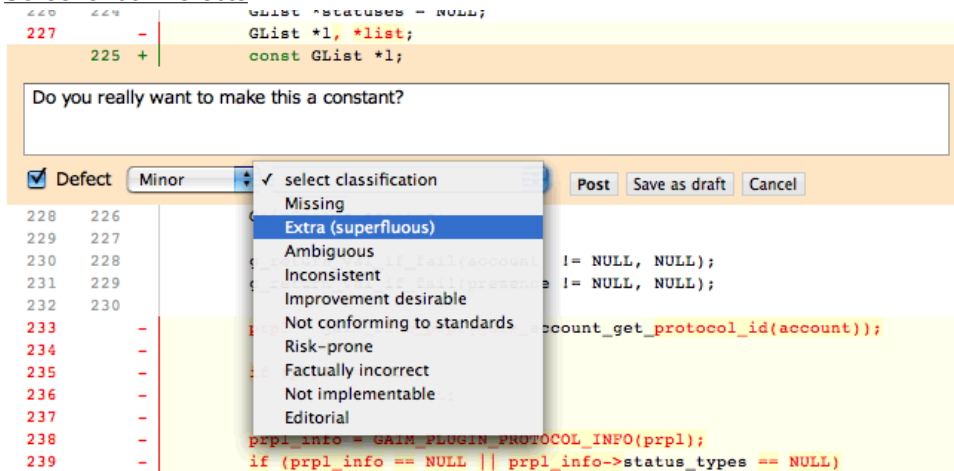
Please post any comments you want to keep, then press **Confirm Complete**

## 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](mailto: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,

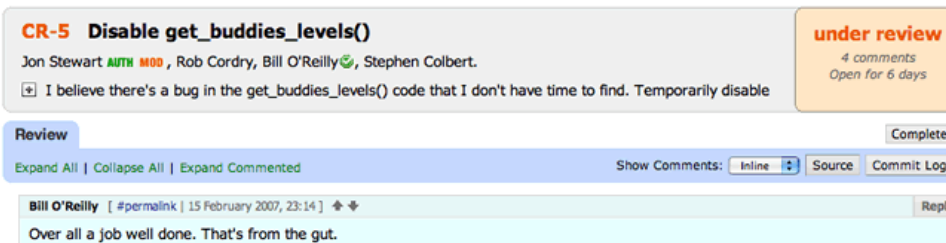
- either click the 'Complete' link at bottom right of the [Dashboard](#),
- or click the 'Complete' button at far right of the 'Review' screen.

**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](mailto:rosie@atlassian.com).

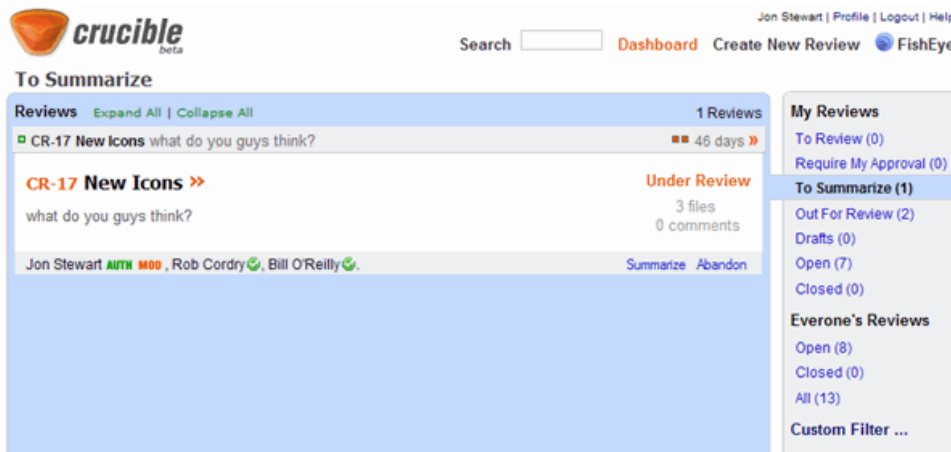
As the [moderator](#), you can choose to summarise a review at any time.

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

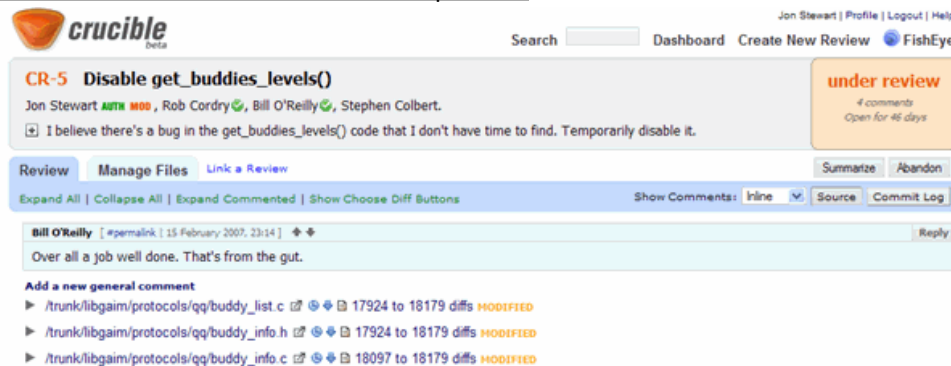
The reviews that the reviewers have completed will be in the 'To Summarize' menu on the [Dashboard](#). There are two ways to summarise a review:


- 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

The screenshot shows the Crucible web application interface. At the top, there's a search bar and user navigation links. The main content area displays a review titled "CR-5 Disable get\_buddies\_levels()". The review is in the "Summarize" state, indicated by a yellow background. The review content area is empty, with a "Close" button at the bottom. Below the review, there are comments from Bill O'Reilly and Jon Stewart. The interface includes a search bar, user profile links, and navigation tabs like "Review", "Manage Files", and "Link a Review".

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.

## 2.5 Deleting an Abandoned Review

This page last changed on Sep 23, 2008 by [edawson](#).

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

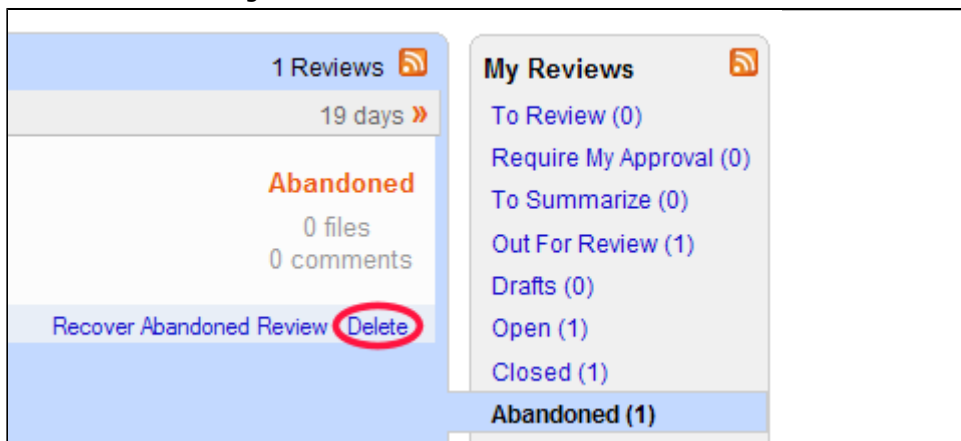
 Deleted reviews cannot be retrieved.

### Deleting Reviews from the 'Abandoned' list

To delete a review from the 'Abandoned' list;

1. From the 'Crucible Dashboard', click 'Abandoned' in the right navigation bar.
2. A list of abandoned reviews appears. With care, click the 'Delete' link situated to the right of the review you wish to remove.
3. The review will be instantly deleted.

Screenshot: Deleting a Review in Crucible



### 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 <a href="#">states</a> e.g: 'Draft', 'Under Review'. An email is sent when the state changes.
Comment added	Default is 'Yes'. An email is sent when a <a href="#">comment is added</a> to a review.
Participant finished	Default is 'Yes'. An email is sent when any reviewer has <a href="#">completed their review</a> .
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' <a href="#">state</a> .
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 Mar 12, 2008 by [edawson](#).

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 [3. Changing your User Profile](#).



## 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 Jul 29, 2008 by [edawson](#).

The creator is the person who [creates the review](#). In most cases this person will also act as [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 Mar 12, 2008 by [edawson](#).

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](mailto: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](mailto: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 <a href="#">set of revisions</a> being reviewed.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'View'	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users ' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> ' ' <a href="#">Author</a> ' ' <a href="#">Reviewer</a> '
'Abandon'	Ability to abandon (i.e. cancel) a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Re-Open'	Ability to re-open a closed or abandoned review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Uncomplete'	Ability of a reviewer to change their individual review status from 'Complete' to 'Uncomplete'.	' <a href="#">Reviewer</a> '
'Reject'	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	' <a href="#">Moderator</a> '
'Complete'	Ability of a reviewer to change their individual review status to 'Complete'.	' <a href="#">Reviewer</a> '
'Comment'	Ability to add or remove a comment to or from a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> ' ' <a href="#">Author</a> ' ' <a href="#">Reviewer</a> '
'Approve'	Ability to approve a review (i.e. issue it to the reviewers).	' <a href="#">Moderator</a> '
'Submit'	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	' <a href="#">Creator</a> ' ' <a href="#">Author</a> '
'Close'	Ability to close a review once it has been summarised.	' <a href="#">Moderator</a> '
'Delete'	Ability to delete a review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '
'Summarise'	Ability to summarise a review. (Normally this would be	' <a href="#">Moderator</a> '

	done after all reviewers have completed their review.)	
'Create'	Ability to create a review.	All logged-in users
'Recover'	Ability to resurrect an abandoned (i.e. cancelled) review.	' <a href="#">Moderator</a> ' ' <a href="#">Creator</a> '



## permission scheme

---

This page last changed on Dec 02, 2007 by [rosie@atlassian.com](mailto: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](mailto: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 Mar 12, 2008 by [edawson](#).

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](mailto: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 <a href="#">Creating a Review</a> .
Require Approval	Relevant only when the <a href="#">moderator</a> is not the <a href="#">creator</a> . See <a href="#">Issuing a Review</a> .
Under Review	See <a href="#">Issuing a Review</a> and <a href="#">Reviewing the Code</a> .
Summarize	See <a href="#">Summarising and Closing the Review</a> .
Closed	See <a href="#">Summarising and Closing the Review</a> .

 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 <a href="#">moderator</a> 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 Mar 12, 2008 by [edawson](#).

A user is a person using Crucible.

## \_\_newreleaseCrucible

---

This page last changed on Sep 23, 2008 by [edawson](#).



Crucible 1.6 has now been released. Read the [Release Notes](#).



## TreeNavigation

---

This page last changed on Oct 04, 2007 by [smaddox](#).

[Index](#)