Atlassian

User Documentation for Crucible 4.2

Contents

Getting Started

Atlassian Crucible is the on-premises code review solution for enterprise teams. It allows your development teams to catch major defects, improve code architecture, and discuss desired improvements, without the need for meetings.

This section describes how to install, set up and get started with Crucible.

System requirements

Crucible is a Java web application, that works with all modern browsers. See our Supported platforms page to find out about system requirements.

Download and install Crucible

- Windows
- Mac
- Linux

Start using Crucible

For a short introduction see Starting to use Crucible. You'll learn how to:

- Add a repository
- Create a project
- Create and perform reviews

Integrate Crucible with other Atlassian applications

As a first step, see JIRA integration in Crucible.

Read more about using Crucible

To find out more about using Crucible with your team, see Using Crucible.

To find out how to manage the Crucible server, see Administering Crucible.

Supported platforms

This page lists the supported platforms for **Crucible 4.2.x** and its minor releases.

Key: ✓ = Supported ⚠ = Deprecated ✗ = Not Supported

Java		
Oracle JRE / JDK	✓ 1.8 ✗ 1.7	Crucible requires the **Java Runtime** (JDK or JRE), version as noted. Pre-release/Early access versions of Java are *not supported*. We highly recommend that you use the Oracle JVM (or OpenJDK for Linux only). Other Java implementations have not been tested. You can download an Oracle Java Runtime. For the OpenJDK, download and install instructions for Linux flavors are at http://openjdk.java.net/install/. Please note:

OpenJDK	✓ 1.8 (Linux only)	• Once you have installed the Java, you need to set the JAVA_HOME environment variable. See Installing Crucible on Windows or Installing Crucible on Linux and Mac for details. • If you are using a 64-bit JVM, please ensure that you've set your max heap size (`--Xmx`) to a reasonable value, considering the RAM requirements of your system. ✗ Support for Java 7 was removed in Crucible 3.9, as previously announced.
Operating Systems		
Microsoft Windows	✓	• Crucible is a pure Java application and should run on any platform provided the requirements for the JRE or JDK are satisfied. Although Crucible can be run in virtualized environments, Atlassian is not yet able to provide technical support for performance-related problems in a virtualized environment. If you do chose to run Crucible in a VM, please ensure that you choose a VM with good IO throughput.
Linux	✓	
Apple Mac OS X	✓	
Databases		
HSQLDB	✓ Bundled; for evaluation use only	The Crucible built-in database, running HSQLDB, is somewhat susceptible to data loss during system crashes. We recommend that you do not use HSQLDB for production systems. External databases are generally more resistant to data loss during a system crash and are more suited for production use. See the Crucible Database documentation for further details.
MySQL	✓ MySQL Enterprise Server 5.5+ ✓ MySQL Community Server 5.5+ ✗ Maria, Percona	For MySQL: • ✗ For 5.6, versions earlier than 5.6.11 are not supported • ✗ For 5.7, versions earlier than 5.7.5 are not supported • ✗ Support for MySQL 5.1 was removed in Crucible 4.2. See End of Support Announcements for Crucible. • ✗ MariaDB and Percona variants of MySQL are not supported, and are known to cause issues when used with Crucible. ✗ Support for PostgreSQL 8.3, 8.4, 9.0 and 9.1 was removed in Crucible 4.2. See End of Support Announcements for Crucible. ✗ Support for SQL Server 2008, 2008 R2 was removed in Crucible 4.2. See End of Support Announcements for Crucible.
PostgreSQL	✓ 9.2, 9.3, 9.4	
Oracle	✓ 12c ✓ 11g	
SQL Server	✓ 2014 ✓ 2012	
Web browsers		

Microsoft Internet Explorer	✓ 10.0, 11.0 ✗ 9.0	Minimum screen resolution of 1024x768. Recommended screen resolution of 1280x768 or above. ✗ Support for Internet Explorer 9 was removed in Crucible 3.9 . See End of Support Announcements for Crucible.
Mozilla Firefox	✓ Latest stable version supported ✗ 3.6, 4.0	✗ Support for Firefox 3.6 and 4.0 was removed in Cruciblee 3.7.
Safari	✓ Latest stable version supported ✗ 4, 5	✗ Support for Safari 4 and 5 was removed in FishEye 3.7.
Chrome	✓ Latest stable version supported	
Version Control Systems		
Subversion (SVN)	Server: ✓ 1.5, 1.6, 1.7, 1.8, 1.9 Client: ✓ SVNKit (bundled & the default) ✓ Native JavaHL 1.9 ✓ Native JavaHL 1.8 ✓ Native JavaHL 1.7 ✗ Native JavaHL 1.6	✗ Crucible 3.1, and later, **do not** support the native JavaHL 1.6 client. See Native support for SVN for discussion.
CVS (and CVSNT)	✓ All versions	
Perforce	✓ Client version 2007.3 or later ✓ Server version 2005.1 or later	The Server must support the ztag tagged protocol. ✗ Perforce Streams, introduced in 2011.1, is not currently supported. See **FE-3886** - support for Streams in p4 OPEN

Git	✓ 2.9.0, 2.9.3 ✓ 2.8.4 ✓ 2.7.4 ✓ 2.6.6 ✓ 2.5.5 ✓ 2.4.11 ✓ 2.3.10 ✓ 2.2.3 ✓ 2.1.4 ✓ 2.0.5 ✓ 1.9.5 ✓ 1.8.0.3, 1.8.1.5, 1.8.2.3, 1.8.3.4, 1.8.4.5, 1.8.5.6 ✓ 1.7.1.1, 1.7.2.5, 1.7.3.5, 1.7.4.5, 1.7.5.4, 1.7.6.6, 1.7.7.7, 1.7.8.6, 1.7.9.7, 1.7.10.5, 1.7.11.7, 1.7.12.4	These are the versions of Git that we currently test Crucible against. ✗ Git 1.8.4.3 is not supported. See **BSERV-4101** - Clone and fetch fail with "protocol error: impossibly long line" CLOSED [Security vulnerability CVE-2014-9390] affects multiple Git versions. Crucible itself is not affected, however you should update your *clients* to a patched maintenance version: v1.8.5.6, v1.9.5, v2.0.5, v2.1.4 and v2.2.1 or newer. Git for Windows is the *only supported distribution* when running Crucible on Windows. Cygwin Git is *not supported* and has known issues.
Mercurial	✓ 3.0.2, 3.1.2, 3.2.4, 3.3.3, 3.4.2, 3.5.2, 3.6.3, 3.7.3, 3.8.3, 3.8.4, 3.9.1 ✓ 2.0.2, 2.1.2, 2.2.3, 2.3.2, 2.4, 2.5.2, 2.6.3, 2.7.2, 2.8.2, 2.9.1 ✓ 1.5.1, 1.5.4, 1.6.4, 1.7.5, 1.8.4, 1.9.3	These are the versions of Mecurial that we currently test Crucible against. As of version 3.6.3, Crucible supports Mercurial 3. ✗ Mercurial 2.1 has a bug that makes it incompatible with Crucible. Please use Mercurial 2.1.1 or later. You should restart Crucible after upgrading Mercurial.
Atlassian applications		
Crowd	✓ Crowd 2.4.x+ ✓ Crowd client library: 2.4.1	From version 2.8.x, FishEye bundles the Crowd 2.4.1 client library, and supports the Crowd 2.4.x server, and later versions.
JIRA	✓ JIRA FishEye Plugin 6.1.0+ ✓ JIRA 6.4+	The JIRA FishEye Plugin is bundled with JIRA. If you are using earlier versions of JIRA you can upgrade the plugin in JIRA to get support for Crucible.

Deployment Notes for Source Code Repositories

Crucible can also store uploaded files in its own database, removing the need for any kind of repository. A number of external databases are supported when Crucible is used with FishEye. See the FishEye Supported Platforms.

WAR deployment

FishEye/Crucible is a standalone Java program. It cannot be deployed to web application servers such as WebSphere, Weblogic or Tomcat.

Single sign on with Atlassian Crowd

From version 2.8.x, FishEye bundles the Crowd 2.4.1 client library, and supports the Crowd 2.4.x server, and later versions.

Installing Crucible on Windows

This page...	Upgrading?	Upgrading from FishEye?
... describes how to perform a clean install of Crucible on Windows.	If you're upgrading your Crucible installation, read the Crucible upgrade guide first.	If you already have FishEye installed, you should read Upgrading from FishEye to Crucible instead.

1. Check supported platforms

Better check the Supported platforms page first; it lists the application servers, databases, operating systems, web browsers and JDKs that we have tested Crucible with, and that we recommend.

Atlassian only officially supports Crucible running on x86 hardware and 64-bit derivatives of x86 hardware.

2. Create a dedicated Crucible user (recommended)

For production installations, we recommend that you create a new dedicated Windows user that will run Crucible on your system. This user:

- Should not have admin privileges.
- Should be a non-privileged user with read, write and execute access on the Crucible home (install) directory and instance (data) directory. These directories are described below.
- Should only have read access to your repositories.

If you created a dedicated Crucible user, ensure you are logged in as this user to complete the remaining instructions.

3. Check your version of Java

In a command prompt, run this:

```
java -version
```

The version of Java should be **1.8.x**.

The recommended way to install Crucible is to use the installer, which installs Crucible as a Windows service – see step 5 below.

If you don't see a supported version of Java, then get Java...

> Download and install the Java Platform JDK from Oracle's website.
>
> ⚠ *We recommend that the Java install path should not contain spaces, so don't install into C:\ Program Files\Java\. Instead, use a path like C:\Java.*
>
> Now try running '`java -version`' again to check the installation. The version of Java should be **1.8.x**.

4. Check that Windows can find Java

Windows uses the JAVA_HOME environment variable to find Java. To check that, in a new command prompt, run:

```
echo %JAVA_HOME%
```

You should see a path to the Java install location. We recommend that this path does *not* contain spaces, and that JAVA_HOME should point to the JDK home path.

If you don't see a path without spaces...

- If you see a path with spaces, like `C:\Program Files\Java\`, then sorry, but go back to 3. and reinstall Java to a location that doesn't have spaces.
- If you don't see a path at all, or if you just see %JAVA_HOME%, then set JAVA_HOME as follows:

For Windows 7:

1. Go to **Start**, search for "sys env" and choose **Edit the system environment variables**.
2. Click **Environment Variables**, and then **New** under 'System variables'.
3. Enter "JAVA_HOME" as the **Variable name**, and the absolute path to where you installed Java JDK as the **Variable value**, that is, something like `C:\Java\jdk1.7.0_51`. Don't use a trailing backslash. We recommend that JAVA_HOME should point to the JDK home path.
4. Now, in a new command prompt, try running '`java -version`'. You should see the same version of Java as you saw above.

5. Now it's time to get Crucible

Download the Crucible installer from the Atlassian download site.

There are 32-bit and 64-bit installers for Crucible on Windows. Each installer adds Crucible as a Windows service, and starts the service, automatically. The express install creates, by default, a `Data` directory and a separate install directory in `C:\Atlassian`. The custom install mode allows you to choose different locations for the install and `Data` directories, with the restriction that the `Data` directory must not be contained in the install directory.

- The installer creates the `FISHEYE_INST` system environment variable.
- The path to the installation directory is referred to as the `<Crucible home directory>` in these instructions.
- You need separate Crucible data directories if you want to run multiple copies of Crucible.
- If you expect to have a large number of users for this Crucible installation, and Crucible will be connected to an external database, consider installing Crucible on a different server from the one running the external database, for improved performance.
- If you have a large number of repositories, we recommend you increase the default number of files that Crucible is allowed to open. See the following knowledge base article for more info: Subversion Indexer Paused with "Too many open files" Error.
- For Crucible 3.4.4 and later, you can edit JVM parameters for the Windows service by going to **Start** > **All Programs** > **Crucible** > **Configure Crucible**. Ensure that you restart the Crucible service when finished. Do *not* reference any environment variables in the settings (e.g. %FISHEYE_INST%). Instead, set the actual path.

6. Visit Crucible!

Give the Crucible service a minute to launch. Then, in a web browser on the same machine, go to `http://localhost:8060/` (or, from another machine, type `http://hostname:8060/`, where `hostname` is the name of the machine where you installed Crucible).

Enter your license, then an admin password, to finish the setup. Note that this password is for the 'built-in' Crucible admin user. You can log in as this user, if necessary, by clicking the **Administration** link in the page footer. See also How to reset the Administration Page password in Fisheye or Crucible.

You can postpone setting up JIRA integration until later if you wish; see Configuring JIRA integration in the Setup Wizard.

7. Connect to an external database (recommended)

If you intend to use this Crucible installation in a production environment, it is highly recommended that you use one of the supported external databases. See Migrating to an external database.

If you are evaluating Crucible, or don't wish to do this now, Crucible will happily use its embedded database, and you can easily migrate later.

8. Set up your mail server

Configure the Crucible email server so that users can get notifications from Crucible. See Configuring SMTP.

9. Add users and repositories

Now is the time to set up your users in Crucible, and to tell Crucible about any existing repositories you have. Please read Starting to use Crucible for the details.

Crucible will perform an initial index of your repositories, during which it accesses, indexes and organizes a view of your repositories (including all historical items) back to the earliest commits. If you are evaluating Crucible, we suggest that you index a single project, so you can use Crucible as soon as possible. If you choose to index your entire repository, be aware that this can take a long time (possibly days) for massive or complex repositories and can be more complex to set up (especially for Subversion). The basic process is slightly different for each SCM type.

10. Stop Crucible (optional)

Control the Crucible service from the Windows administration console. Alternatively, in a command prompt, change directory to `<Crucible home directory>` and run this:

```
bin\stop.bat
```

Installing Crucible on Linux and Mac

Hey! We're going to install Crucible on a Linux box, or a Mac. There are a few steps involved, but we think you'll find it easy to follow along. If you already have FishEye installed, you should read Upgrading from FishEye to Crucible instead.

1. Check supported platforms

Better check the Supported platforms page first; it lists the application servers, databases, operating systems, web browsers and JDKs that we have tested Crucible with, and that we recommend.

Atlassian only officially supports Crucible running on x86 hardware and 64-bit derivatives of x86 hardware.

Related pages:

- Installing Crucible on Windows
- Starting to use Crucible
- Supported platforms

2. Create a dedicated Crucible user (recommended)

For production installations, we recommend that you create a new user account on your operating system that is dedicated to running Crucible. This user:

- Should not have admin privileges.
- Should be a non-privileged user with read, write and execute access on the Crucible home (install) directory and instance (data) directory. These directories are described below.
- Should only have read access to your repositories.

If you created a dedicated Crucible user, ensure you are logged in as this user to complete the remaining instructions.

3. Check your version of Java

In a terminal, run this:

```
java -version
```

The version of Java should be **1.8.x**.

If you don't see a supported version of Java, then get Java...

Download and install the Oracle Java Platform JDK, or OpenJDK.

Now try running '`java -version`' again to check the installation. The version of Java should be **1.8.x**.

4. Check that the system can find Java

In a terminal, run this:

```
echo $JAVA_HOME
```

You should see a path something like:

OSX	`/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/`
Linux	`/usr/lib/jvm/default-java`

If you don't see a path to the Java location, then set JAVA_HOME...

Linux	Mac

Do either of the following:
• If `JAVA_HOME` is not set, log in with 'root' level permissions and run:
`echo JAVA_HOME="path/to/JAVA_HOME" >> /etc/environment`
where `path/to/JAVA_HOME` may be like: `/usr/lib/jvm/default-java`
• If `JAVA_HOME` needs to be changed, open the `/etc/environment` file in a text editor and modify the value for `JAVA_HOME` to:
`JAVA_HOME="path/to/JAVA_HOME"`
It should look like:
`JAVA_HOME=/usr/lib/jvm/default-java`
	Insert the following in your `~/.profile`
	`JAVA_HOME="path/to/JAVA_HOME"`
	`export JAVA_HOME`
	where `path/to/JAVA_HOME` may be like
	`/System/Library/Frameworks/Java`
	Refresh your ~/.profile in the terminal and
	`source ~/.profile`
	`$JAVA_HOME/bin/java -vers`
	You should see a version of Java that is 1
	`java version "1.8.0_05"`

5. Now it's time to get Crucible

1. Download Crucible from the Atlassian download site.
2. Please check your unzip program before extracting the downloaded zip file. Some archive-extract programs cause errors when unzipping the Crucible zip file:
   - Windows users must avoid the Windows built-in unzip utility, as it doesn't extract all the files. Use a third-party unzip program like 7-Zip or Winzip.
   - Solaris users will need to use GNU tar to handle the long file names.
3. Extract the downloaded file to an install location:
   - Folder names in the path to your Crucible executable should not have spaces in them. The path to the extracted directory is referred to as the `<Crucible home directory>` in these instructions. If you use FishEye and Crucible together, they run as one instance, and use the same home directory – see Crucible and FishEye.
   - If you expect to have a large number of users for this Crucible installation, and Crucible will be connected to an external database, consider installing Crucible on a different server from the one running the external database, for improved performance.

6. Tell Crucible where to store your data

The Crucible instance directory is where your Crucible data is stored.

1. Create your Crucible instance directory.
2. Tell Crucible where you created the instance directory by adding a FISHEYE_INST environment variable as follows:

Linux	Mac

Open the `/etc/environment` file in a text editor and insert :
`FISHEYE_INST="path/to/<Crucible instance directory>"`
	Open the `~/.profile` file for the current user in a text editor and insert:
	`FISHEYE_INST="path/to/<Crucible instance directory>"`
	`export FISHEYE_INST`

3. Now copy the newly extracted `<Crucible home directory>\config.xml` file to the root of your new Crucible instance directory.

⚠ You *should not* locate your Crucible instance directory inside the `<Crucible home directory>` — they should be entirely separate locations. If you do put the instance directory in the `<Crucible home directory>` it will be overwritten, and lost, when Crucible gets upgraded. And by the way, you'll need separate Crucible instance directories if you want to run multiple copies of Crucible.

If you have a large number of repositories, we recommend you increase the default number of files that FishEye is allowed to open. See the following knowledge base article for more info: Subversion Indexer Paused with "Too many open files" Error.

7. Start Crucible!

In a terminal, change directory to `<Crucible home directory>` and run this:

```
bin/start.sh
```

After a few moments, in a web browser on the same machine, go to `http://localhost:8060/` (or, from another machine, type `http://hostname:8060/`, where `hostname` is the name of the machine where you extracted Crucible).

Enter your license, then an admin password, to finish the setup.

You can postpone setting up JIRA integration until later if you wish; see Configuring JIRA integration in the Setup Wizard.

8. Connect to an external database (recommended)

If you intend to use this Crucible installation in a production environment, it is highly recommended that you use one of the supported external databases. See Migrating to an external database.

If you are evaluating Crucible, or don't wish to do this now, Crucible will happily use its embedded database, and you can easily migrate later.

9. Set up your mail server

Configure the Crucible email server so that users can get notifications from Crucible. See Configuring SMTP.

10. Add users and repositories

Now is the time to set up your users in Crucible, and to tell Crucible about any existing repositories you have. Please read Starting to use Crucible for the details.

Crucible will perform an initial index of your repositories, during which it accesses, indexes and organizes a view of your repositories (including all historical items) back to the earliest commits. If you are evaluating Crucible, we suggest that you index a single project, so you can use Crucible as soon as possible. If you choose to index your entire repository, be aware that this can take a long time (possibly days) for massive or complex repositories and can be more complex to set up (especially for Subversion). The basic process is slightly different for each SCM type.

11. Stop Crucible (optional)

In a terminal, change directory to `<Crucible home directory>` and run this:

```
bin/stop.sh
```

Configuring JIRA Integration in the Setup Wizard

This page describes the 'Connect to JIRA' screen of the Crucible setup wizard.
You can connect your application to a JIRA server, to manage your users via JIRA and share information with JIRA. When you are installing the application, the setup wizard gives you the opportunity to configure the JIRA connection automatically. This is a quick way of setting up your JIRA integration with the most common options.

You can also configure the JIRA connections via the application administration screens. In that case, you will need to set up connections individually. There are two parts to the integration process:

- A peer-to-peer link between JIRA and the application for sharing information and facilitating integration features. This link is set up via Application Links.
- A client-server link between the application and JIRA for delegating user and group management to your JIRA server.

Requirements: You need JIRA 4.3 or later.

On this page:

Related pages:

- Linking Crucible to JIRA
- User management limitations and recommendations
- JIRA integration in Crucible

Connecting to JIRA in the Setup Wizard

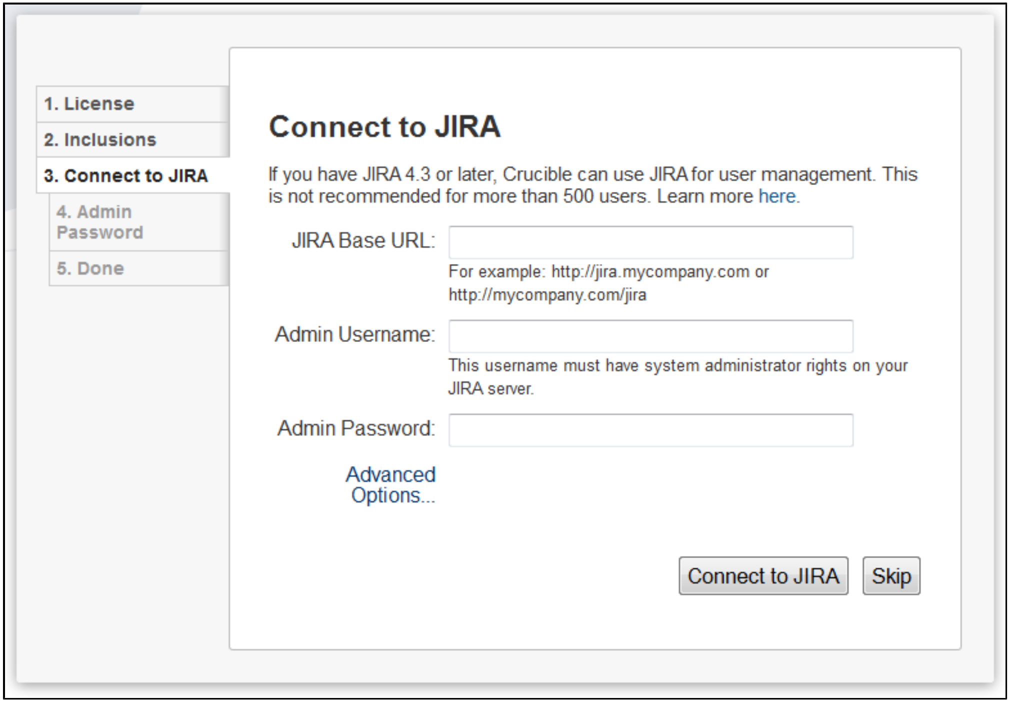
To configure JIRA integration while running the Crucible setup wizard:

1. Enter the following information on the '**Connect to JIRA**' step of the setup wizard:
    - **JIRA Base URL**– The web address of your JIRA server. Examples:

    ```
    http://www.example.com:8080/jira/
    http://jira.example.com
    ```

    - **Admin Username** and **Admin Password** – The credentials of a user with the 'JIRA System Administrators' global permission in JIRA.
    - **FishEye/Crucible Base URL** – Click 'Advanced Options' to see this field. JIRA will use this URL to access your FishEye/Crucible server. The URL you give here will override the base URL specified in your FishEye/Crucible administration console, for the purposes of the JIRA connection.
    - **Groups to synchronize** – Click 'Advanced Options' to see this field. Select at least one JIRA group to synchronize. The default group is `jira-users`. JIRA will synchronize all changes in the user information on a regular basis. The default synchronization interval is 1 hour.
    - **Admin Groups** – Click 'Advanced Options' to see this field. Specify a JIRA group whose members should have administrative access to FishEye/Crucible. The default group is `jira-administrators`.
2. Click the '**Connect to JIRA**' button.
3. Finish the setup process.
4. Configure the following setting in JIRA: Allow remote API access.

Screenshot: Connecting to JIRA in the FishEye/Crucible setup wizard



Connect to JIRA

If you have JIRA 4.3 or later, Crucible can use JIRA for user management. This is not recommended for more than 500 users. Learn more here.

JIRA Base URL:

For example: http://jira.mycompany.com or http://mycompany.com/jira

Admin Username:

This username must have system administrator rights on your JIRA server.

Admin Password:

Troubleshooting

Click to see troubleshooting information...

This section describes the possible problems that may occur when integrating your application with JIRA via the setup wizard, and the solutions for each problem.

Symptom	Cause	Solution
The setup wizard displays one of the following error messages: • Failed to create application link from JIRA server at <URL> to this <application> server at <URL>. • Failed to create application link from this <application> server at <URL> to JIRA server at <URL>. • Failed to authenticate application link from JIRA server at <URL> to this <application> server at <URL>. • Failed to authenticate application link from <application> server at <URL> to this JIRA server at <URL>.	The setup wizard failed to complete registration of the peer-to-peer application link with JIRA. JIRA integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays one of the following error messages: • Failed to register <application> configuration in JIRA for shared user management. Received invalid response from JIRA: <response> • Failed to register <application> configuration in JIRA for shared user management. Received: <response>	The setup wizard failed to complete registration of the client-server link with JIRA for user management. The peer-to-peer link was successfully created, but integration is only partially configured.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays the following error message: • Error setting Crowd authentication	The setup wizard successfully established the peer-to-peer link with JIRA, but could not persist the client-server link for user management in your `config.xml` file. This may be caused by a problem in your environment, such as a full disk.	Please investigate and fix the problem that prevented the application from saving the configuration file to disk. Then remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
The setup wizard displays the following error message: • Error reloading Crowd authentication	The setup wizard has completed the integration of your application with JIRA, but is unable to start synchronizing the JIRA users with your application.	Restart your application. You should then be able to continue with the setup wizard. If this solution does not work, please contact Atlassian Support.

The setup wizard displays the following error message: - An error occurred: java.lang.IllegalStateException: Could not create the application in JIRA/Crowd (code: 500). Please refer to the logs for details.	The setup wizard has not completed the integration of your application with JIRA. The links are only partially configured. The problem occurred because there is already a user management configuration in JIRA for this <application> URL.	Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup. Detailed instructions are below.
No users can log in after you have set up the application with JIRA integration.	Possible causes: - There are no users in the group that you specified on the 'Connect to JIRA' screen. - For FishEye: There are no groups specified in the 'groups to synchronize' section of your administration console. - For Stash: You may not have granted any JIRA groups or users permissions to log in to Stash.	Go to JIRA and add some usernames to the group. - For FishEye: Go to the FishEye administration screens and specify at least one group to synchronize. The default is '**jira-users**'. - For Stash: Grant the **Stash User** permission to the relevant JIRA groups on the Stash Global permissions page. If this solution does not work, please contact Atlassian Support.

Solution 1: Removing a Partial Configuration – The Easiest Way

If the application's setup wizard fails part-way through setting up the JIRA integration, you may need to remove the partial configuration from JIRA before continuing with your application setup. Please follow the steps below.

Remove the partial configuration if it exists, try the 'Connect to JIRA' step again, and then continue with the setup wizard:

1. Log in to JIRA as a user with the '**JIRA System Administrators**' global permission.
2. Click the '**Administration**' link on the JIRA top navigation bar.
3. Remove the application link from JIRA, if it exists:
    a. Click **Application Links** in the JIRA administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
    b. Look for a link to your application. It will have a base URL of the application linked to JIRA. For example:
        - If you want to remove a link between JIRA and FishEye, look for the one where the **Application URL** matches the base URL of your FishEye server.
        - If you want to remove a link between JIRA and Confluence, look for the one where the **Application URL** matches the base URL of your Confluence server.
        - If you want to remove a link between JIRA and Stash, look for the one where the **Application URL** matches the base URL of your Stash server.
    c. Click **Delete** next to the application link that you want to delete.
    d. A confirmation screen will appear. Click **Confirm** to delete the application link.
4. Remove the user management configuration from JIRA, if it exists:
    a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
        - In JIRA 4.3: Click '**Other Applications**' in the '**Users, Groups & Roles**' section of the JIRA administration screen.
        - In JIRA 4.4: Select **'Administration' > 'Users' > 'JIRA User Server'**.
    b. Look for a link to your application. It will have a name matching this format:

```
<Type> - <HostName> - <Application ID>
```

For example:

```
FishEye / Crucible - localhost -
92004b08-5657-3048-b5dc-f886e662ba15
```

Or:

```
Confluence - localhost -
92004b08-5657-3048-b5dc-f886e662ba15
```

If you have multiple servers of the same type running on the same host, you will need to match the application ID of your application with the one shown in JIRA. To find the application ID:

- Go to the following URL in your browser:

  ```
  <baseUrl>/rest/applinks/1.0/manifest
  ```

  Replace `<baseUrl>` with the base URL of your application.
  For example:

  ```
  http://localhost:8060/rest/applinks/1.0/manifest
  ```

- The application links manifest will appear. Check the application ID in the `<id>` element.

c. In JIRA, click '**Delete**' next to the application that you want to remove.

5. Go back to the setup wizard and try the 'Connect to JIRA' step again.

Solution 2: Removing a Partial Configuration – The Longer Way

If solution 1 above does not work, you may need to remove the partial configruration and then add the full integration manually. Please follow these steps:

1. Skip the 'Connect to JIRA' step and continue with the setup wizard, to complete the initial configuration of the application.
2. Log in to JIRA as a user with the '**JIRA System Administrators**' global permission.
3. Click the '**Administration**' link on the JIRA top navigation bar.
4. Remove the application link from JIRA, if it exists:
   a. Click **Application Links** in the JIRA administration menu. The 'Configure Application Links' page will appear, showing the application links that have been set up.
   b. Look for a link to your application. It will have a base URL of the application linked to JIRA. For example:
      - If you want to remove a link between JIRA and FishEye, look for the one where the **Application URL** matches the base URL of your FishEye server.
      - If you want to remove a link between JIRA and Confluence, look for the one where the **Application URL** matches the base URL of your Confluence server.
      - If you want to remove a link between JIRA and Stash, look for the one where the **Application URL** matches the base URL of your Stash server.

   c. Click **Delete** next to the application link that you want to delete.
   d. A confirmation screen will appear. Click **Confirm** to delete the application link.
5. Remove the user management configuration from JIRA, if it exists:
   a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
      - In JIRA 4.3: Click '**Other Applications**' in the '**Users, Groups & Roles**' section of the

        JIRA administration screen.
    - In JIRA 4.4: Select **'Administration' > 'Users' > 'JIRA User Server'**.

   b. Look for a link to your application. It will have a name matching this format:

      ```
      <Type> - <HostName> - <Application ID>
      ```

      For example:

      ```
      FishEye / Crucible - localhost -
      92004b08-5657-3048-b5dc-f886e662ba15
      ```

      Or:

      ```
      Confluence - localhost -
      92004b08-5657-3048-b5dc-f886e662ba15
      ```

      If you have multiple servers of the same type running on the same host, you will need to match the application ID of your application with the one shown in JIRA. To find the application ID:

      - Go to the following URL in your browser:

        ```
        <baseUrl>/rest/applinks/1.0/manifest
        ```

        Replace `<baseUrl>` with the base URL of your application.
        For example:

        ```
        http://localhost:8060/rest/applinks/1.0/manifest
        ```

      - The application links manifest will appear. Check the application ID in the `<id>` element.

   c. In JIRA, click '**Delete**' next to the application that you want to remove.

6. Add the application link in JIRA again, so that you now have a two-way trusted link between JIRA and your application:

   a. Click **Add Application Link**. Step 1 of the link wizard will appear.

   b. Enter the **server URL** of the application that you want to link to (the 'remote application').

   c. Click **Next**.

   d. Enter the following information:

      - **Create a link back to this server** – Check to add a two-way link between the two applications.
      - **Username** and **Password** – Enter the credentials for a username that has administrator access to the remote application.
        Note: These credentials are only used to authenticate you to the remote application, so that Application Links can make the changes required for the new link. The credentials are not saved.
      - **Reciprocal Link URL** – The URL you give here will override the base URL specified in your remote application's administration console, for the purposes of the application links connection. Application Links will use this URL to access the remote application.

   e. Click **Next**.

   f. Enter the information required to configure authentication for your application link:

      - **The servers have the same set of users** – Check this box, because the users are the same in both applications.
      - **These servers fully trust each other** – Check this box, because you trust the code in both applications and are sure both applications will maintain the security of their private keys.
        For more information about configuring authentication, see Configuring authentication for an application link.

   g. Click **Create**.

7. Configure a new connection for user management in JIRA:
    a. Go to the JIRA administration screen for configuring the applications that have been set up to use JIRA for user management:
        - In JIRA 4.3: Click '**Other Applications**' in the '**Users, Groups & Roles**' section of the JIRA administration screen.
        - In JIRA 4.4: Select **'Administration' > 'Users' > 'JIRA User Server'**.

    b. **Add** an application.
    c. Enter the **application name** and **password** that your application will use when accessing JIRA.
    d. Enter the **IP address** or addresses of your application. Valid values are:
        - A full IP address, e.g. `192.168.10.12`.
        - A wildcard IP range, using CIDR notation, e.g. `192.168.10.1/16`. For more information, see the introduction to CIDR notation on Wikipedia and RFC 4632.
        - **Save** the new application.
8. Set up the JIRA user directory in the application.
    - For Confluence:
        a. Go to the **Confluence Administration Console**.
        b. Click '**User Directories**' in the left-hand panel.
        c. **Add** a directory and select type '**Atlassian JIRA**'.
        d. Enter the following information:
            - **Name** – Enter the name of your JIRA server.
            - **Server URL** – Enter web address of your JIRA server. Examples:

```
http://www.example.com:8080/jira/
http://jira.example.com
```

            - **Application name** and **Application password** – Enter the values that you defined for Confluence in the settings on JIRA.

        e. Save the directory settings.
        f. Define the **directory order** by clicking the blue up- and down-arrows next to each directory on the '**User Directories**' screen.
        For details see Connecting to Crowd or JIRA for User Management.
    - For FishEye/Crucible:
        a. Click **Authentication** (under 'Security Settings').
        b. Click **Setup JIRA/Crowd authentication**. Note, if LDAP authentication has already been set up, you will need to remove that before connecting to JIRA for user management.
        c. Make the following settings:

Authenticate against	Select **a JIRA instance**
Application name and **password**	Enter the values that you defined for your application in the settings on JIRA.

JIRA URL	The web address of your JIRA server. Examples:
	`http://www.example.com:8080/jira/`
	`http://jira.example.com`

Auto-add	Select **Create a FishEye user on successful login** so that your JIRA users will be automatically added as a FishEye user when they first log in.
Periodically synchronise users with JIRA	Select **Yes** to ensure that JIRA will synchronize all changes in the user information on a regular basis. Change the value for **Synchronise Period** if required.

When Synchronisation Happens	Select an option depending on whether you want to allow changes to user attributes from within FishEye.
Single Sign On	Select **Disabled**. SSO is not available when using JIRA for user management and if enabled will make the integration fail.

   d. Click **Next** and select at least one user group to be synchronised from JIRA. If necessary, you could create a new group in JIRA, such as 'fisheye-users', and select this group here.
   e. Click **Save**.

- For Stash:
   a. Go to the Stash administration area.
   b. Click **User Directories** in the left-hand panel.
   c. **Add** a directory and select type **Atlassian JIRA**.
   d. Enter the following information:
      - **Name** – Enter the name of your JIRA server.
      - **Server URL**– Enter web address of your JIRA server. Examples:

```
http://www.example.com:8080/jira/
http://jira.example.com
```

      - **Application name** and **Application password** – Enter the values that you defined for Stash in the settings on JIRA.

   e. Save the directory settings.
   f. Define the directory order by clicking the blue up- and down-arrows next to each directory on the 'User Directories' screen.
   For details see Connecting Stash to JIRA for user management.

Having trouble integrating your Atlassian products with application links?
We've developed a guide to troubleshooting application links, to help you out. Take a look at it if you need a hand getting around any errors or roadblocks with setting up application links.

Notes

When you connect to JIRA in the setup wizard, the setup procedure will configure **Trusted Applications authentication** for your application. Please be aware of the following security implications:

- Trusted applications are a **potential security risk**. When you configure Trusted Applications authentication, you are allowing one application to access another as any user. This allows all of the built-in security measures to be bypassed. Do not configure a trusted application unless you know that all code in the application you are trusting will behave itself at all times, and you are sure that the application will maintain the security of its private key.

Starting to use Crucible

This page introduces the basics of using Crucible. By the end, you'll know how to:

- Add a repository
- Create a project
- Create and perform reviews

For more information, see the Crucible user's guide.

Assumptions

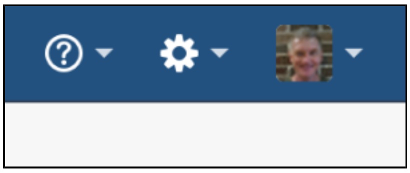
We're assuming that:

- You have installed and started the latest version of Crucible. See the details at Installing Crucible on Windows or Installing Crucible on Linux and Mac.
- You are using a supported browser.
- You have admin permission in Crucible.

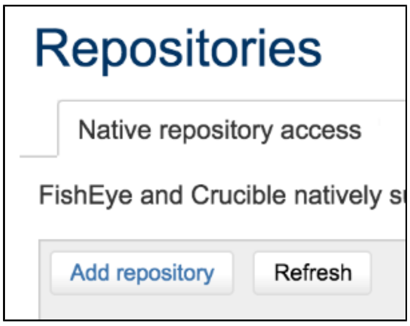
Add a repository

First up we're going to add a repository to Crucible.

Go to the Admin area by clicking on the 'cog' at the top right and choosing **Administration**:



Click **Add repository** in the **Repositories** listing of the Administration area:

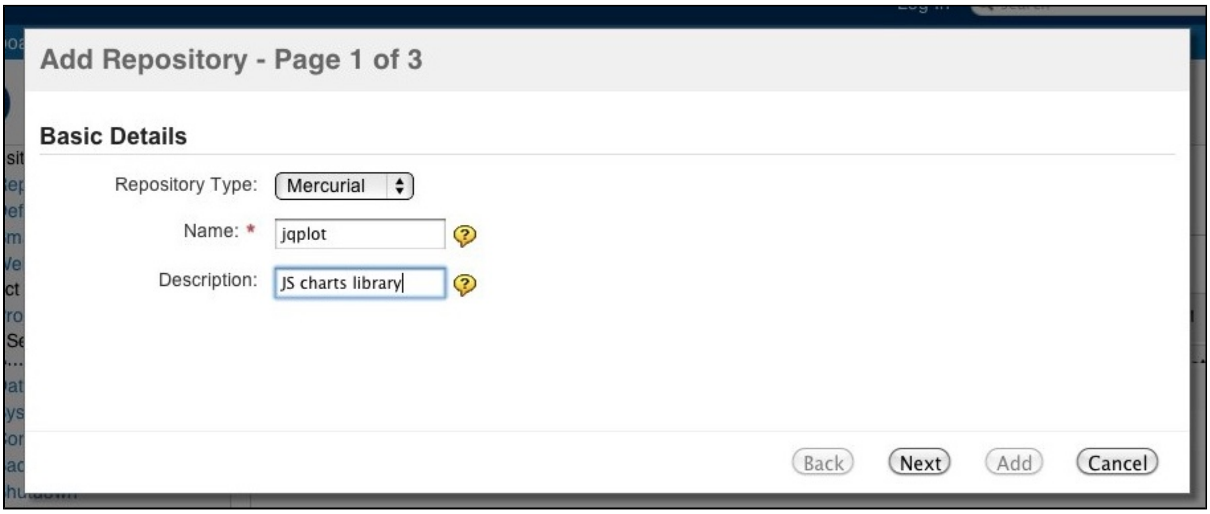


Repositories

Native repository access

FishEye and Crucible natively s

Choose the repository type and fill in the name and description.



Add Repository - Page 1 of 3

Basic Details

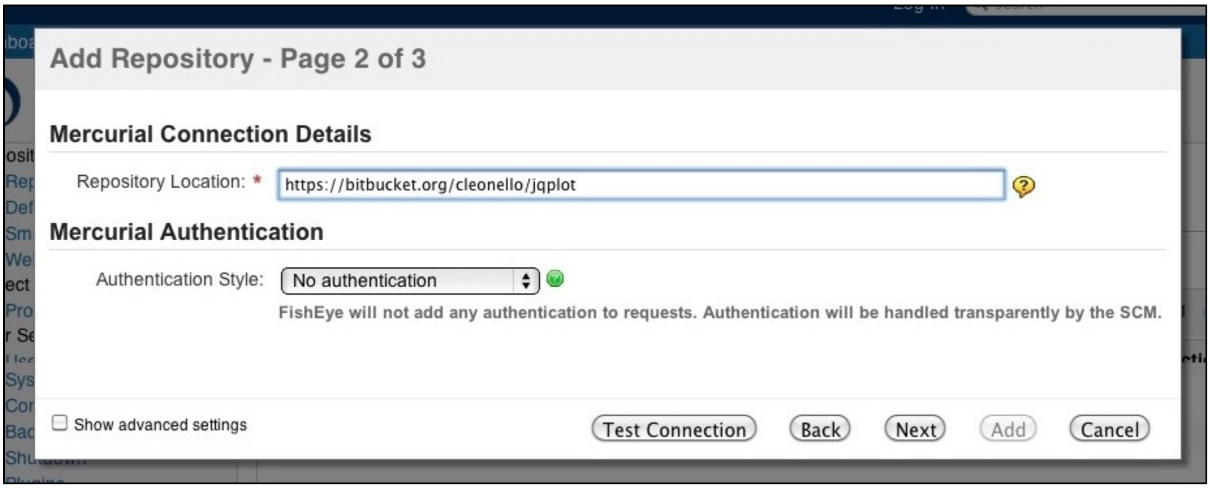
Repository Type: Mercurial

Name: * jqplot

Description: JS charts library

In the repository configuration put the location of your repository. Fill in the authentication details if necessary.

Finally, indicate whether or not you would like diff indexing turned on and if the repository should be indexed right away, then click **Add** to finish the process.



Add Repository - Page 2 of 3

Mercurial Connection Details

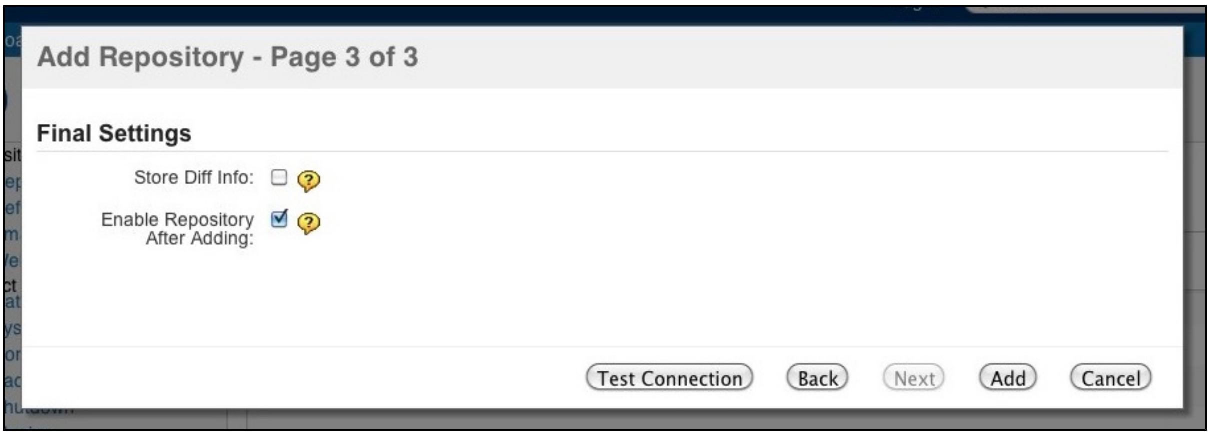
Repository Location: * https://bitbucket.org/cleonello/jqplot

Mercurial Authentication

Authentication Style: No authentication

FishEye will not add any authentication to requests. Authentication will be handled transparently by the SCM.

□ Show advanced settings



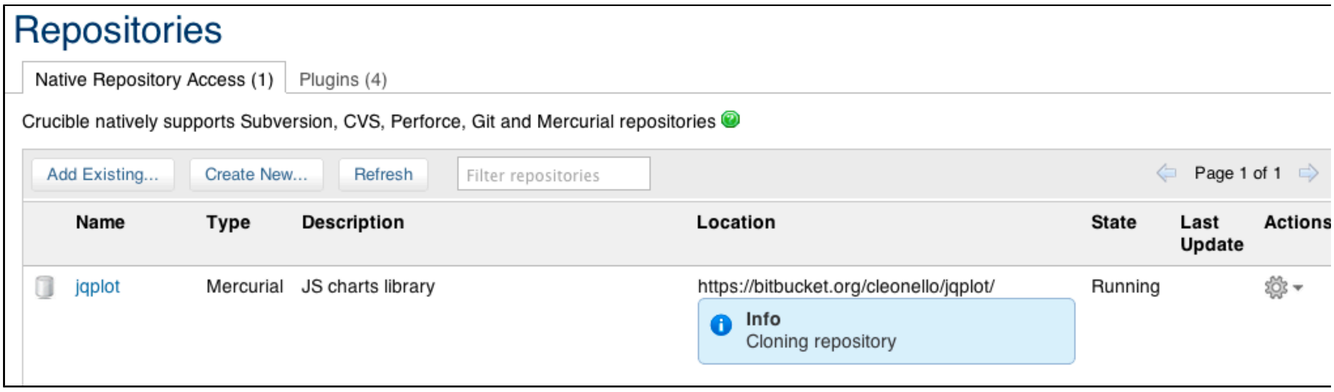
Add Repository - Page 3 of 3

Final Settings

Store Diff Info: □

Enable Repository After Adding: ☑

Your repository is now created in **Crucible** and the indexing should have started.



Repositories

Native Repository Access (1) Plugins (4)

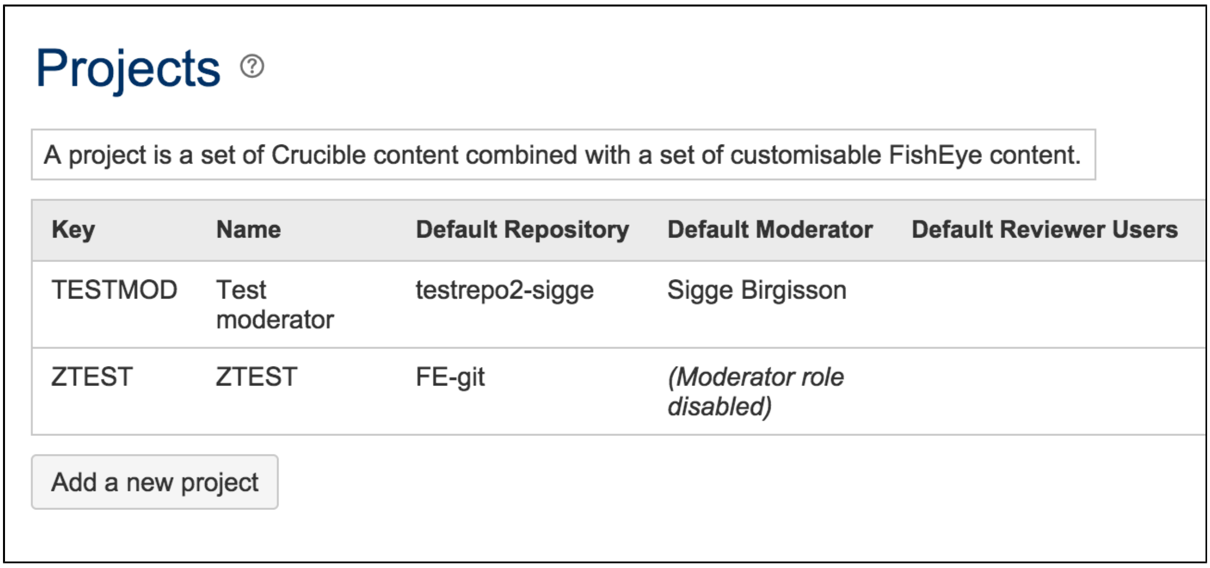
Crucible natively supports Subversion, CVS, Perforce, Git and Mercurial repositories

Name	Type	Description	Location	State	Last Update	Actions
jqplot	Mercurial	JS charts library	https://bitbucket.org/cleonello/jqplot/ Info Cloning repository	Running		

Create a project in Crucible

Crucible comes with a default project, with the key CR, but you will will probably want to create your own projects to contain your reviews. This is achieved in a couple of steps.

Click **Add a new project** in the **Projects** listing of the Administration area.

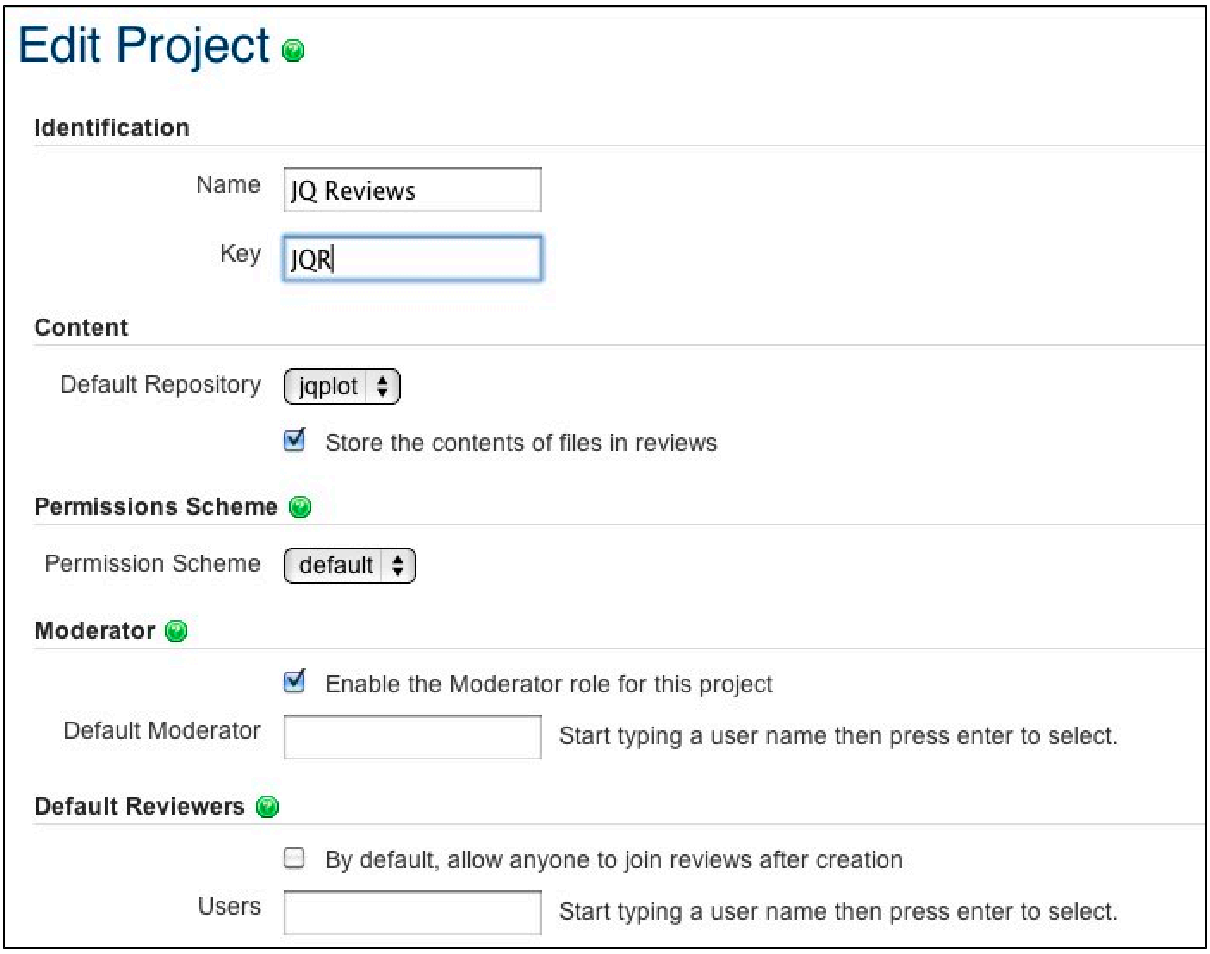


Projects

A project is a set of Crucible content combined with a set of customisable FishEye content.

Key	Name	Default Repository	Default Moderator	Default Reviewer Users
TESTMOD	Test moderator	testrepo2-sigge	Sigge Birgisson	
ZTEST	ZTEST	FE-git	*(Moderator role disabled)*	

Fill in the form with the default settings for the project and hit **Save**.



Edit Project

Identification

Name: JQ Reviews

Key: JQR

Content

Default Repository: jqplot

☑ Store the contents of files in reviews

Permissions Scheme

Permission Scheme: default

Moderator

☑ Enable the Moderator role for this project

Default Moderator: Start typing a user name then press enter to select.

Default Reviewers

☐ By default, allow anyone to join reviews after creation

Users: Start typing a user name then press enter to select.

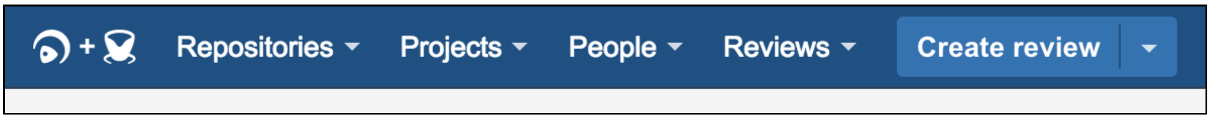
You'll see your new project in the **Projects** listing.

Create a review

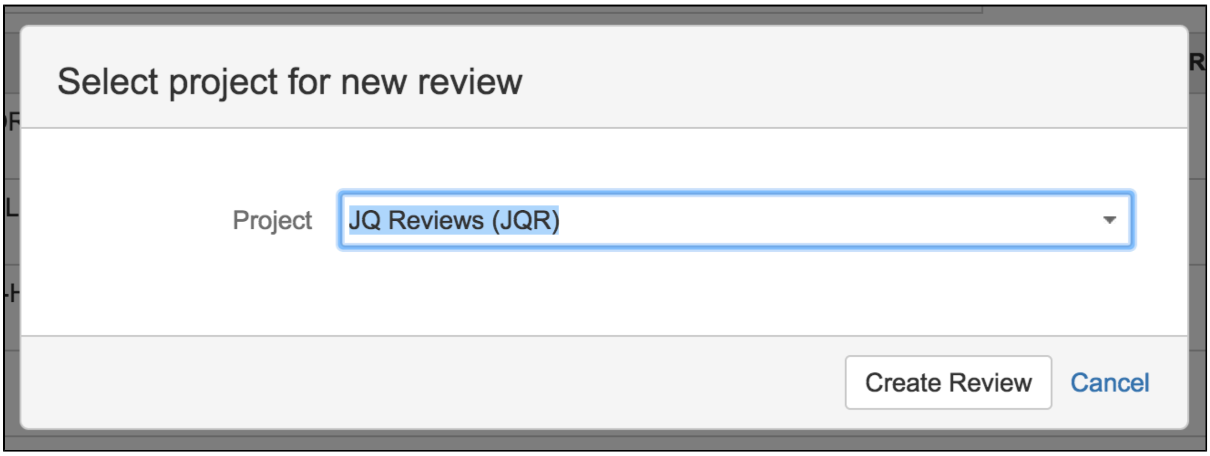
Now that you have your own project you can create reviews in it.

Of course, you'll need to be logged in to create a review.

From the header, click **Create review** to open the review creation form:



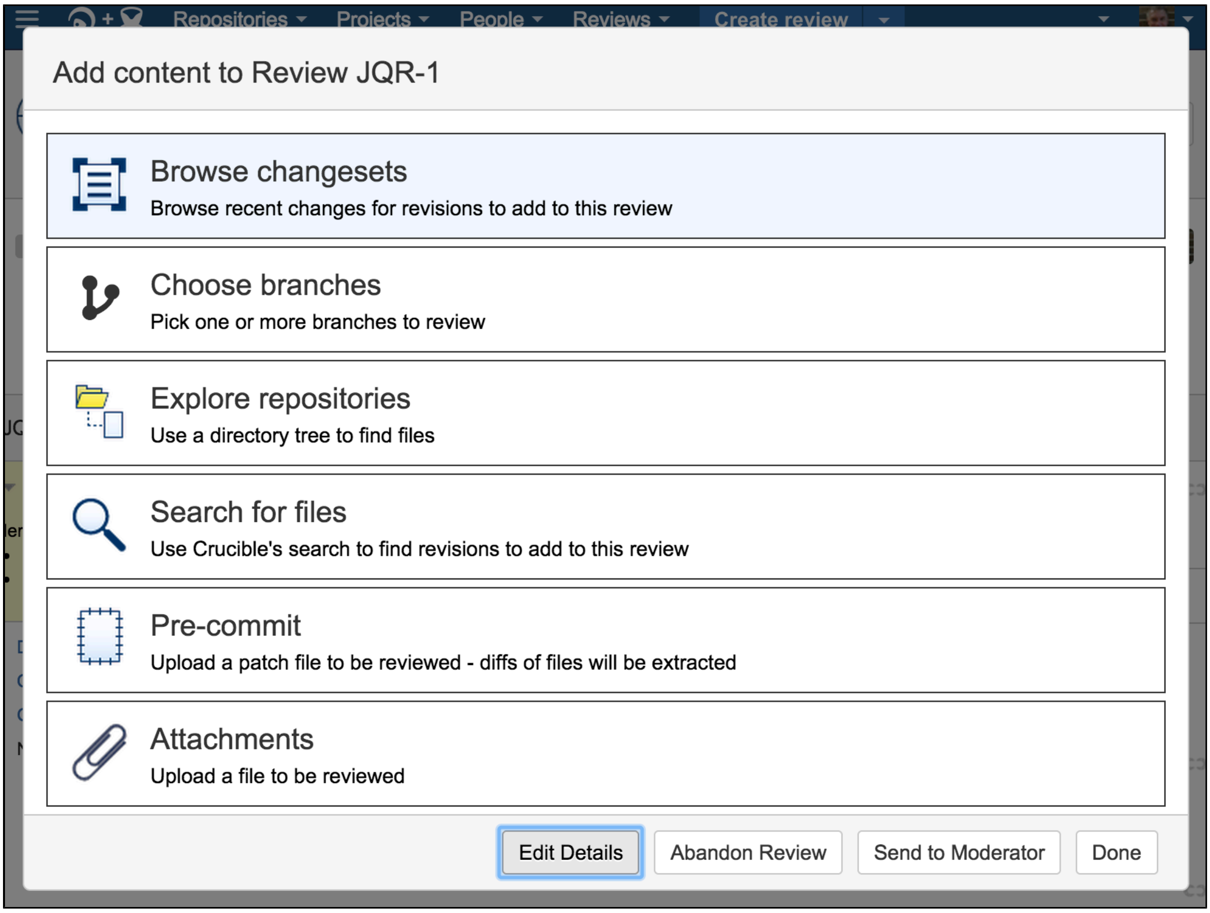
Choose the project in which you want to create the review:



Select project for new review

Project: JQ Reviews (JQR)

In the next screen, click **Browse changesets** to see the list of changesets available for the review.



Add content to Review JQR-1

Browse changesets
Browse recent changes for revisions to add to this review

Choose branches
Pick one or more branches to review

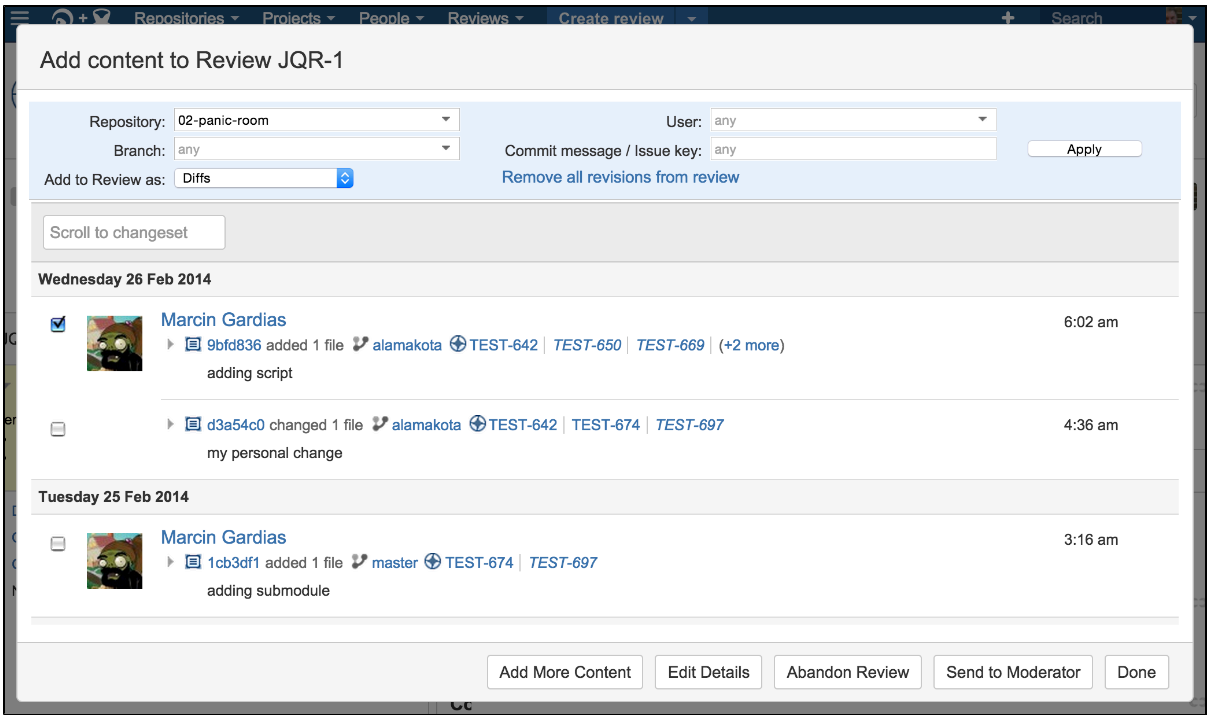
Explore repositories
Use a directory tree to find files

Search for files
Use Crucible's search to find revisions to add to this review

Pre-commit
Upload a patch file to be reviewed - diffs of files will be extracted

Attachments
Upload a file to be reviewed

Select the changesets that you want to be reviewed, then click **Edit Details**:



Add content to Review JQR-1

Repository: 02-panic-room
Branch: any
Add to Review as: Diffs
User: any
Commit message / Issue key: any

Wednesday 26 Feb 2014

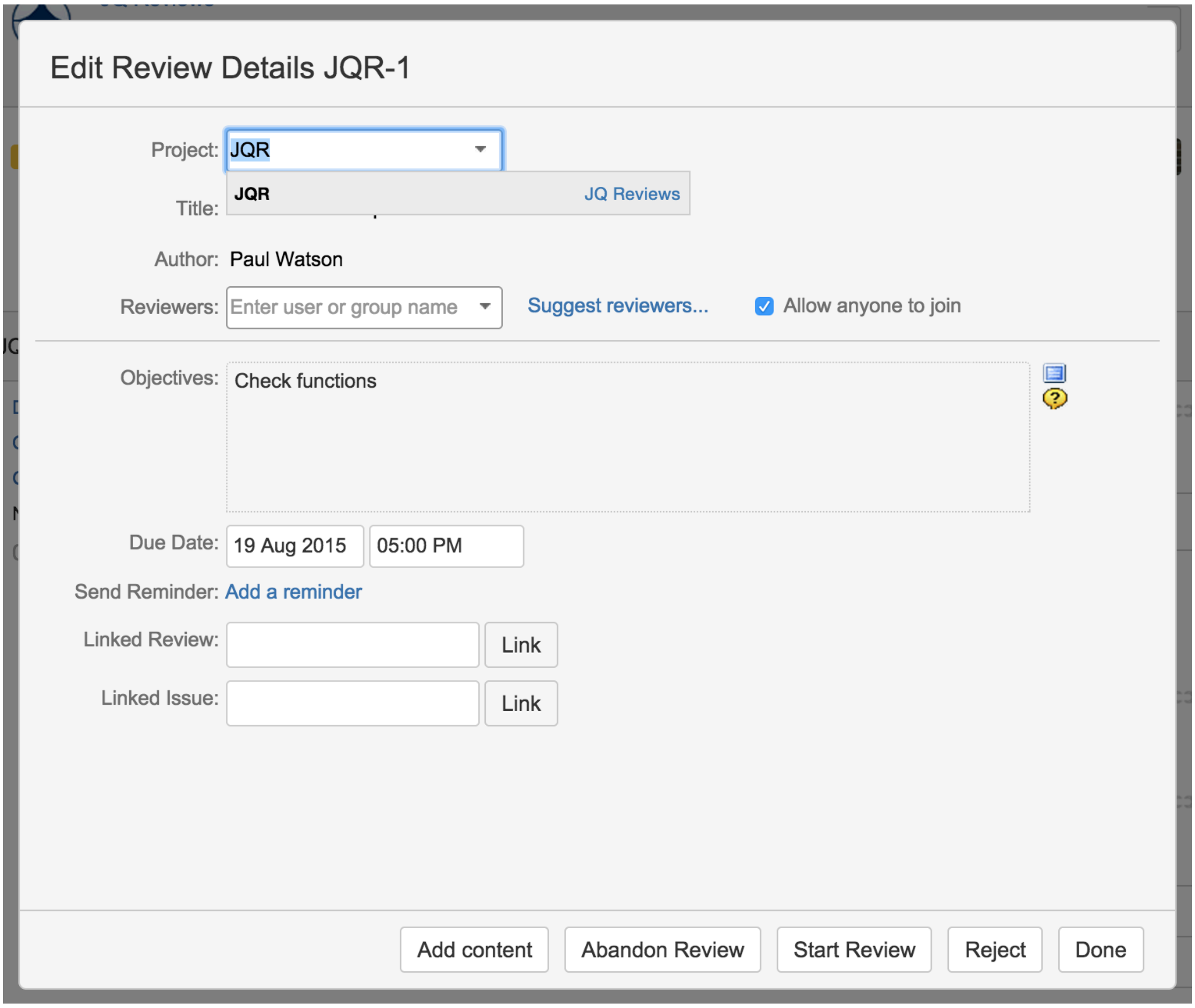
☑ Marcin Gardias 6:02 am
9bfd836 added 1 file alamakota TEST-642 | TEST-650 | TEST-669 | (+2 more)
adding script

□ d3a54c0 changed 1 file alamakota TEST-642 | TEST-674 | TEST-697 4:36 am
my personal change

Tuesday 25 Feb 2014

□ Marcin Gardias 3:16 am
1cb3df1 added 1 file master TEST-674 | TEST-697
adding submodule

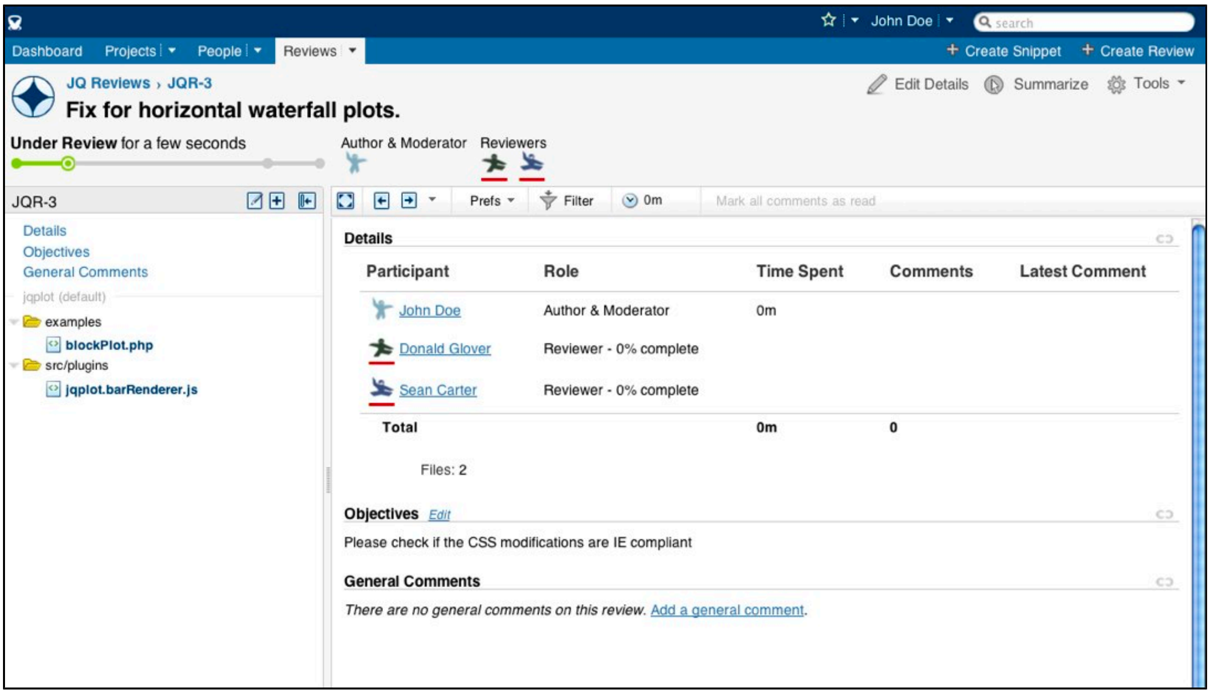
Now add reviewers and update the review information, then click **Start Review**:



Edit Review Details JQR-1

Project: JQR
JQR JQ Reviews
Title:
Author: Paul Watson
Reviewers: Enter user or group name
☑ Allow anyone to join
Objectives: Check functions
Due Date: 19 Aug 2015 05:00 PM
Send Reminder: Add a reminder
Linked Review:
Linked Issue:

The review is now created and the reviewers will have been notified that a review is pending.



Fix for horizontal waterfall plots.

Under Review for a few seconds

Details

Participant	Role	Time Spent	Comments	Latest Comment
John Doe	Author & Moderator	0m		
Donald Glover	Reviewer - 0% complete			
Sean Carter	Reviewer - 0% complete			
Total		0m	0	

Files: 2

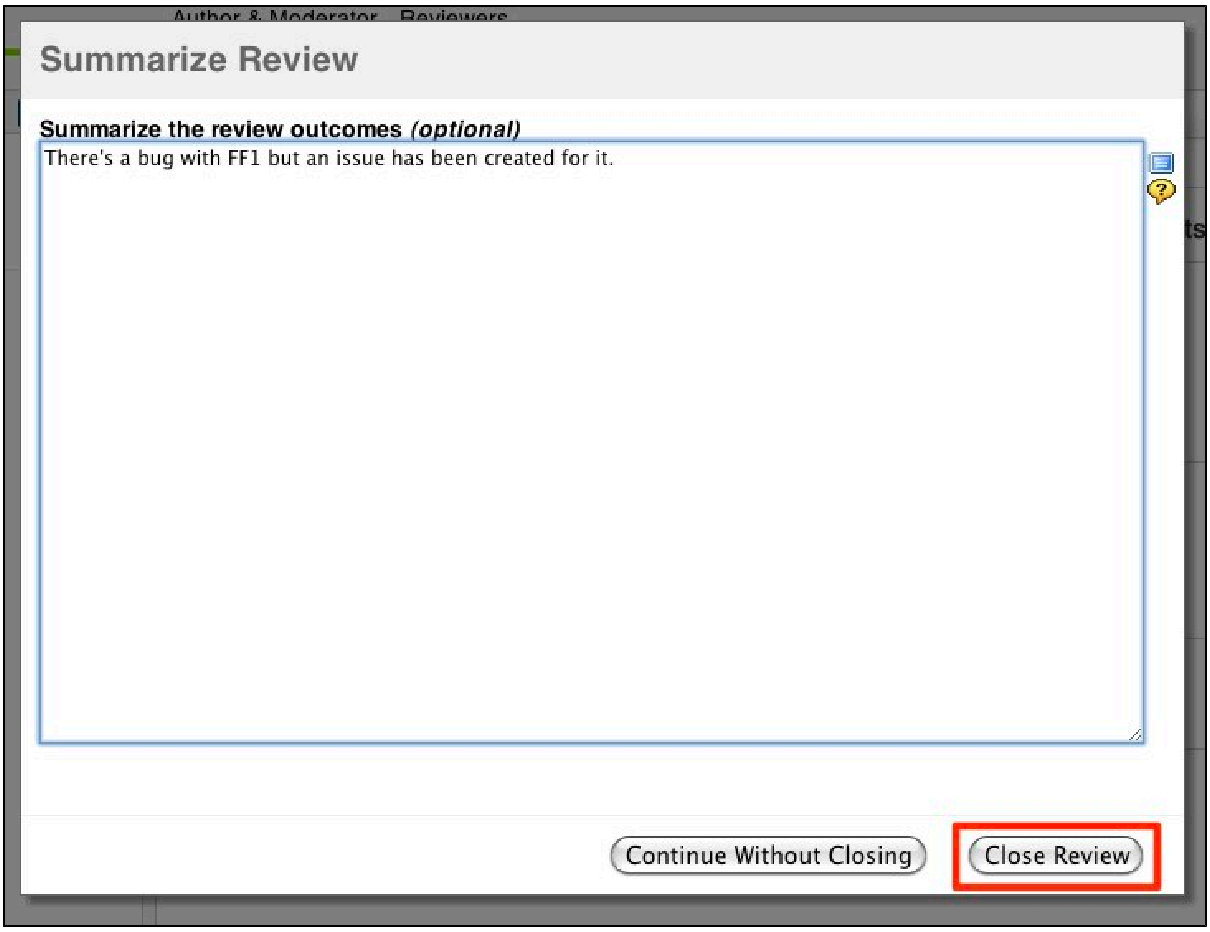
Objectives *Edit*

Please check if the CSS modifications are IE compliant

General Comments

There are no general comments on this review. Add a general comment.

In order to close a review, when you are the moderator, you need to click on **Summarize** at the top right and then close the review from the dialog:



Summarize Review

Summarize the review outcomes ***(optional)***

There's a bug with FF1 but an issue has been created for it.

Installing and upgrading Git

This page describes how to install or upgrade Git on the Crucible server:

- Check your version of Git
- Install or upgrade Git on Linux
- Install or upgrade Git on Mac OS X
- Install or upgrade Git on Windows

- Restart Crucible

Check your version of Git

The versions of Git supported by Crucible are listed on Supported platforms.

You can check your current version of Git by running the `git --version` command in a terminal (Linux, Mac OS X) or command prompt (Windows).

For example:

```
git --version
git version 1.7.7.3
```

If you don't see a supported version of Git, you'll need to either upgrade Git or perform a fresh install, as described below.

Install or upgrade Git on Linux

Use your package manager to install Git. For example, on Ubuntu 13.10, use:

```
sudo apt-get install git
```

If you are using a different Linux distribution, you may need to use a different package repository to get the latest stable version of Git.

Now check the Git version – you should see the new version of Git.

If you still can't see the Git version, you may need to add the Git install location to your path. Open your `~/.profile` file in a text editor and add this line, where `<path/to/git>` is the install location for Git:

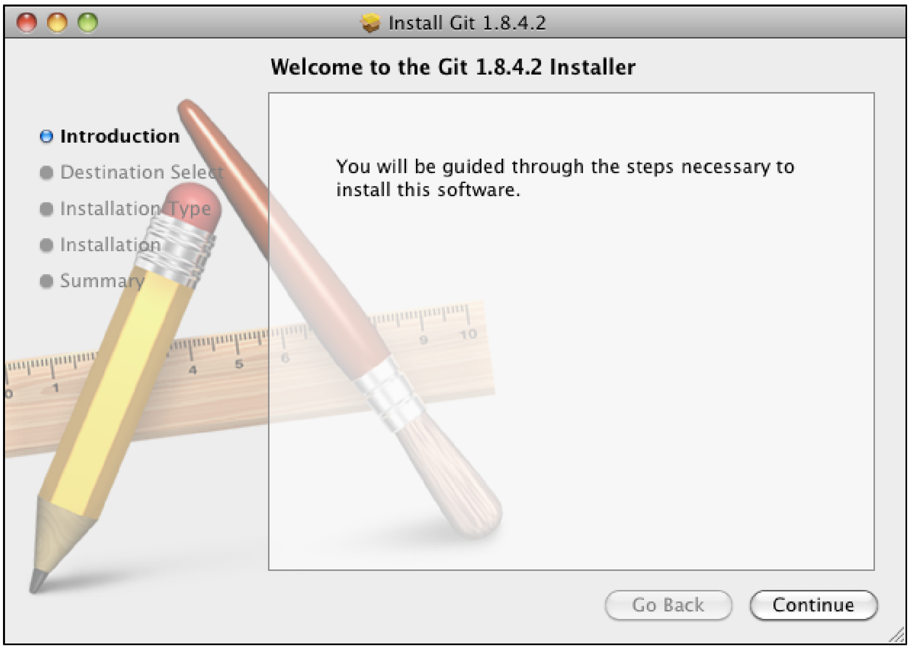
```
export PATH=$PATH:<path/to/git>
```

You can use the `which git` command to find the install location for Git.

Install or upgrade Git on Mac OS X

This section describes how to install the latest stable Git release on your Mac. It does not describe how to update the version of Git that is bundled with Apple's Xcode.

Download the latest stable Git release from the Git website. Click on the downloaded .dmg file, then double-click the .pkg icon to run the installer. This will install the new version of Git over the existing version:



You will be guided through the steps necessary to install this software.

Now check the Git version – you should see the new version of Git.

If you still can't see the Git version, you may need to add the Git install location to your path. Open your `~/.profile` file in a text editor and add this line, where `<path/to/git>` is the install location for Git:

```
export PATH=$PATH:<path/to/git>
```

You can use the `which git` command to find the install location for Git.

Install or upgrade Git on Windows

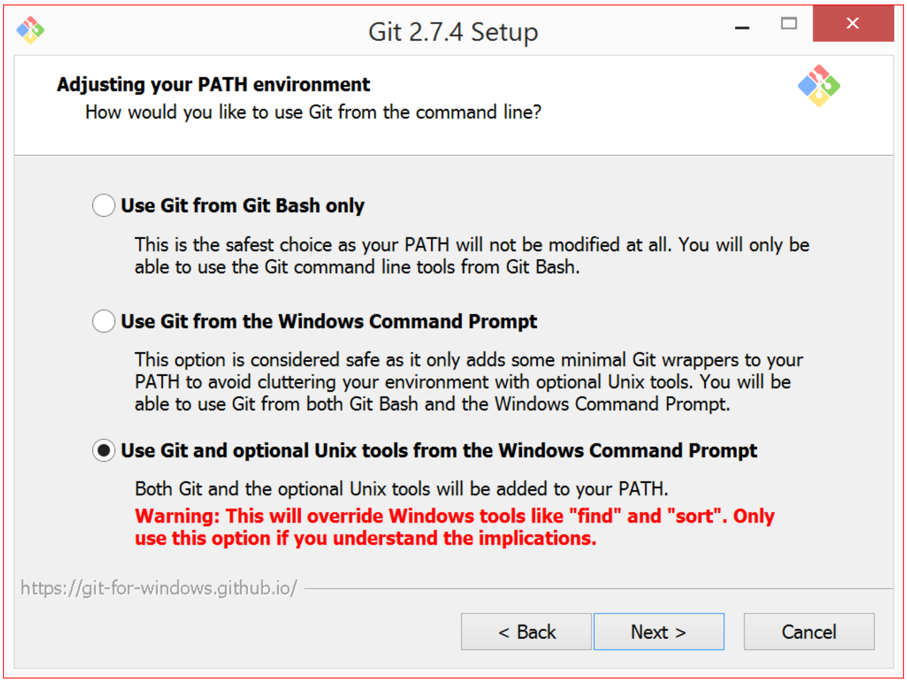
Download the Full installer for official Git for Windows. Installing Git for Windows (previously known as msysGit) also installs a supported version of Perl.

⚠ Git for Windows is the *only supported distribution* when running Crucible on Windows. Cygwin Git is *not supported* and has known issues.

Run the Git installer, ensuring that you install into the same location as any existing Git installation. You can use `where git` to locate existing installations.

Ensure that git.exe is available in the path:

- Choose either Option 2, *Run Git from the Windows Command Prompt*, or Option 3, *Run Git and included Unix tools from the Windows Command Prompt*. Both these options will work with Crucible.
- Do *not* select Option 1, *Use Git Bash only*, when installing or upgrading Git for the Crucible server – *this will not work with Crucible.*



Git 2.7.4 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

○ **Use Git from the Windows Command Prompt**

This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

● **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.

Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://git-for-windows.github.io/

Now, check the Git version – you should see the new version of Git.

Restart Crucible

You'll need to stop and restart Crucible so that it will pick up the upgraded version of Git.

On Windows:

Control the Crucible service from the Windows administration console. Alternatively, in a command prompt, change directory to `<Crucible home directory>` and run:

```
bin\start.bat
```

On Linux and Mac OS X:
In a terminal, change directory to `<Crucible home directory>` and run:

```
bin\start.sh
```

Using Crucible

Atlassian Crucible is the on-premises code review solution for enterprise teams. It allows your development teams to catch major defects, improve code architecture, and discuss desired improvements, without the need for meetings.

This page provides an overview of how to use Crucible.

1. Point Crucible to your repositories

Crucible is all about code reviews. It's no surprise then that Crucible needs access to your source code.

A Crucible administrator can connect a repository managed by any of these tools:

- Bitbucket Server

- Git
- Subversion
- Mercurial
- CVS
- Perforce

2. Set up a Crucible project

A Crucible 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.

A Crucible administrator can create new projects – see Creating a project.

3. Review something!

When you create a review you'll want to:

- Add the files, changesets or other content that you want to be reviewed.
- Choose the people who you want to be reviewers.

We've found that reviews should be created with care to get the best value from them:

- Avoid overloading the review. Reviews should be focused on just a few necessary files.
- Avoid overcrowding the review. Reviewers should be selected with care, and should be guided individually on what to look for.

See Creating a review for more information.

Using the Crucible screens

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

On this page:

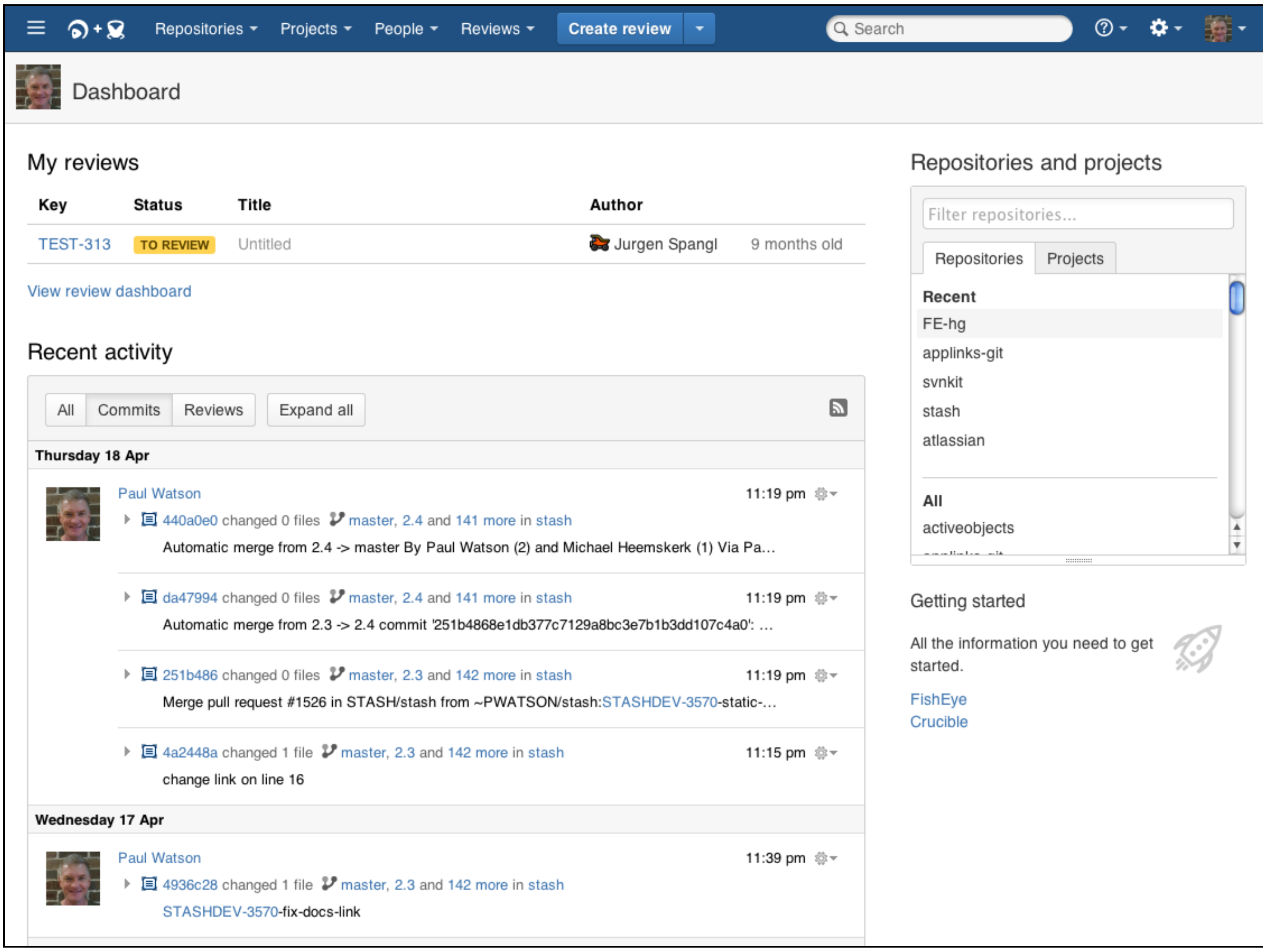
- Dashboard
- Header
- Recent activity
- Related pages

Dashboard

The dashboard is the first screen you see when you log into FishEye/Crucible. The dashboard displays reviews and system activity related to you, and provides filtering for your recent repositories and projects. The dashboard can be accessed from anywhere in the application by clicking the FishEye/Crucible icon in the header.

Click **View review dashboard** to see more information about your reviews.

Screenshot: The Crucible dashboard (with FishEye), showing current reviews and recent activity



My reviews

Key	Status	Title	Author	
TEST-313	TO REVIEW	Untitled	Jurgen Spangl	9 months old

View review dashboard

Recent activity

Thursday 18 Apr

Paul Watson 11:19 pm

440a0e0 changed 0 files master, 2.4 and 141 more in stash
Automatic merge from 2.4 -> master By Paul Watson (2) and Michael Heemskerk (1) Via Pa...

da47994 changed 0 files master, 2.4 and 141 more in stash 11:19 pm
Automatic merge from 2.3 -> 2.4 commit '251b4868e1db377c7129a8bc3e7b1b3dd107c4a0': ...

251b486 changed 0 files master, 2.3 and 142 more in stash 11:19 pm
Merge pull request #1526 in STASH/stash from ~PWATSON/stash:STASHDEV-3570-static-...

4a2448a changed 1 file master, 2.3 and 142 more in stash 11:15 pm
change link on line 16

Wednesday 17 Apr

Paul Watson 11:39 pm

4936c28 changed 1 file master, 2.3 and 142 more in stash
STASHDEV-3570-fix-docs-link

Repositories and projects

Recent
FE-hg
applinks-git
svnkit
stash
atlassian

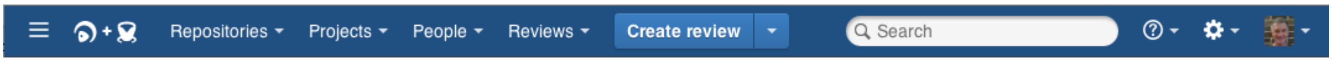
All
activeobjects

Getting started

All the information you need to get started.

FishEye
Crucible

Header



The table below explains the tabs in the Crucible header:

Tab	Function	Appears
Repositories	Displays contents of connected source repositories. The dropdown menu has links to recently visited repositories.	Only when FishEye is used with Crucible. Only when FishEye is used with Crucible for logged-in users.
Projects	Displays reviews and content from specific projects. The dropdown menu has links to recently visited projects.	All screens All screens for logged-in users.
People	Displays metrics on the users of the Crucible instance. The dropdown menu has links to recently visited user pages.	All screens. All screens for logged-in users.
Reviews	Allows you to search and report on reviews. The dropdown menu has links to recently visited reviews, as well as links to the Crucible Inbox and Outbox. Choose **Reviews** > **Review dashboard** to see the Review Dashboard that has more information about your reviews.	All screens All screens for logged-in users. .

Recent activity

The dashboard has an activity stream that displays recent commit activity and reviews activity. The activity stream will display your own activity as well as information from projects, reviews, people, repositories, etc, that you have selected as favorites. For more information on favorites, see Using favorites.

Browsing commit activity

Commit activity includes files commits to repositories that you have selected as favorites.

Click the **Commits** tab to filter the activity stream to display only source activity.

Browsing reviews activity

Reviews activity includes updates to reviews in all projects that you have selected as favorites. See Browsing all reviews for more information about browsing reviews.

Click the **Reviews** tab to filter the activity stream to display only reviews activity (see screenshot below).

Related pages

Browsing source files
Browsing projects
Viewing People's Statistics in Crucible
Viewing reports
Searching Crucible
Using RSS feeds in Crucible
Changing your User Profile

Browsing all reviews

To browse reviews in Crucible, choose **Reviews** > **Review dashboard**.

The dashboard displays reviews according to the filters you click in the sidebar:

- Your reviews
- Other reviews
- Custom filter
- Reports
- Related topics

Your reviews

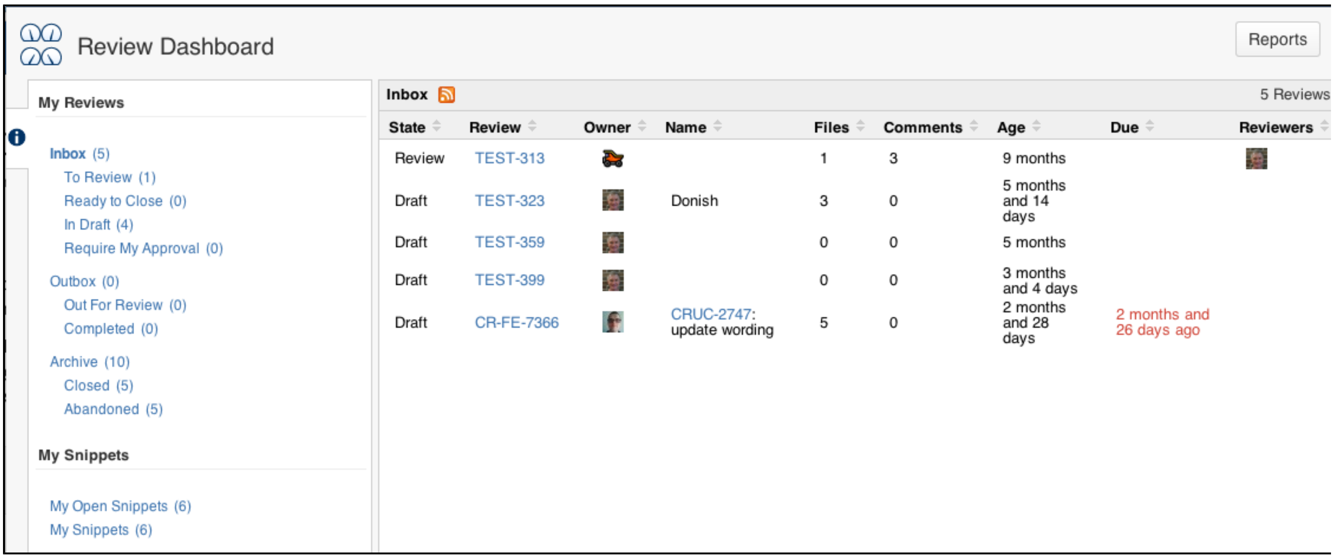
By default, the dashboard shows the reviews you are involved in.

- Browse your reviews by clicking the links under 'My Reviews' and 'My Snippets' in the sidebar.

Inbox	
To Review	Reviews where you are a reviewer and haven't yet completed your review work.
Ready to Close	Reviews where you are a moderator and haven't yet summarized and closed the review.
In Draft	Reviews that you have created but have not yet been moved to the 'Approval' state or the 'Require Approval' state.
Require My Approval	Reviews where you are a moderator and need to approve the review.
Outbox	

Out for Review	Reviews that you are a participant of, that have review work that is yet to be completed by other reviewers.
Completed	Reviews that you are a participant of, and have been completed.
Archive	
Closed	Reviews that you are a participant of, that have been summarized and closed.
Abandoned	Reviews that you are a participant of, that have been abandoned. You may wish to delete these reviews.
My Open Snippets	All open snippets created by you.
My Snippets	All snippets created by you.

Screenshot: Browsing your reviews



Review Dashboard

Reports

My Reviews

- Inbox (5)
  - To Review (1)
  - Ready to Close (0)
  - In Draft (4)
  - Require My Approval (0)
- Outbox (0)
  - Out For Review (0)
  - Completed (0)
- Archive (10)
  - Closed (5)
  - Abandoned (5)

My Snippets

- My Open Snippets (6)
- My Snippets (6)

Inbox 5 Reviews

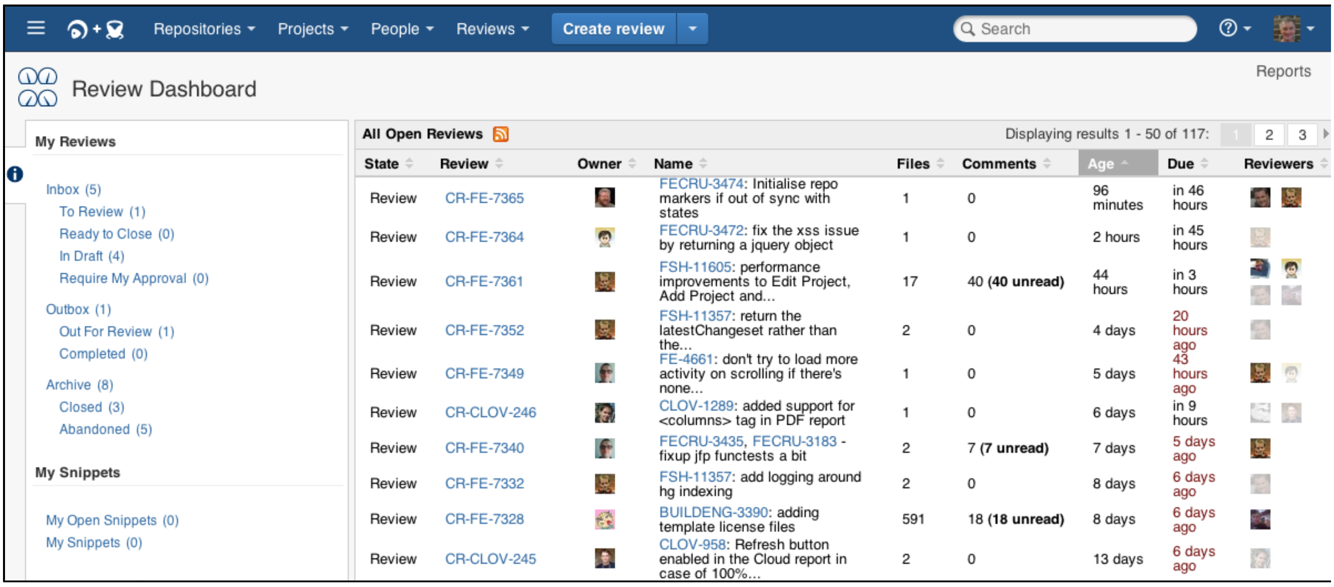
State	Review	Owner	Name	Files	Comments	Age	Due	Reviewers
Review	TEST-313			1	3	9 months		
Draft	TEST-323		Donish	3	0	5 months and 14 days		
Draft	TEST-359			0	0	5 months		
Draft	TEST-399			0	0	3 months and 4 days		
Draft	CR-FE-7366		CRUC-2747: update wording	5	0	2 months and 28 days	2 months and 26 days ago	

Other reviews

- Browse reviews for all people by clicking the links under 'Everyone's Reviews' and 'Everyone's Snippets' in the sidebar:

All Open Reviews	Reviews that have not been summarized and closed yet.
All Closed Reviews	Reviews that have been summarized and closed.
All Reviews	All reviews, including open reviews, closed reviews and draft reviews.
All Open Snippets	All open snippets.
All Snippets	All snippets, i.e. open and closed snippets.

Screenshot: Browsing all open reviews



State	Review	Owner	Name	Files	Comments	Age	Due	Reviewers
Review	CR-FE-7365		FECRU-3474: Initialise repo markers if out of sync with states	1	0	96 minutes	in 46 hours	
Review	CR-FE-7364		FECRU-3472: fix the xss issue by returning a jquery object	1	0	2 hours	in 45 hours	
Review	CR-FE-7361		FSH-11605: performance improvements to Edit Project, Add Project and...	17	40 **(40 unread)**	44 hours	in 3 hours	
Review	CR-FE-7352		FSH-11357: return the latestChangeset rather than the...	2	0	4 days	20 hours ago	
Review	CR-FE-7349		FE-4661: don't try to load more activity on scrolling if there's none...	1	0	5 days	43 hours ago	
Review	CR-CLOV-246		CLOV-1289: added support for <columns> tag in PDF report	1	0	6 days	in 9 hours	
Review	CR-FE-7340		FECRU-3435, FECRU-3183 - fixup jfp functests a bit	2	7 **(7 unread)**	7 days	5 days ago	
Review	CR-FE-7332		FSH-11357: add logging around hg indexing	2	0	8 days	6 days ago	
Review	CR-FE-7328		BUILDENG-3390: adding template license files	591	18 **(18 unread)**	8 days	6 days ago	
Review	CR-CLOV-245		CLOV-958: Refresh button enabled in the Cloud report in case of 100%...	2	0	13 days	6 days ago	

Custom filter

You can filter reviews by author or by projects that you have selected as favorites.

Reports

Click **Reports** at the top of the screen to generate reports on review blockers for all people.You can also filter reviews by status, e.g. 'Open', 'Closed'.

Related topics

Viewing reports

Browsing source files

When FishEye is installed with Crucible, the **Repositories** tab is available in the header.

To browse source files:

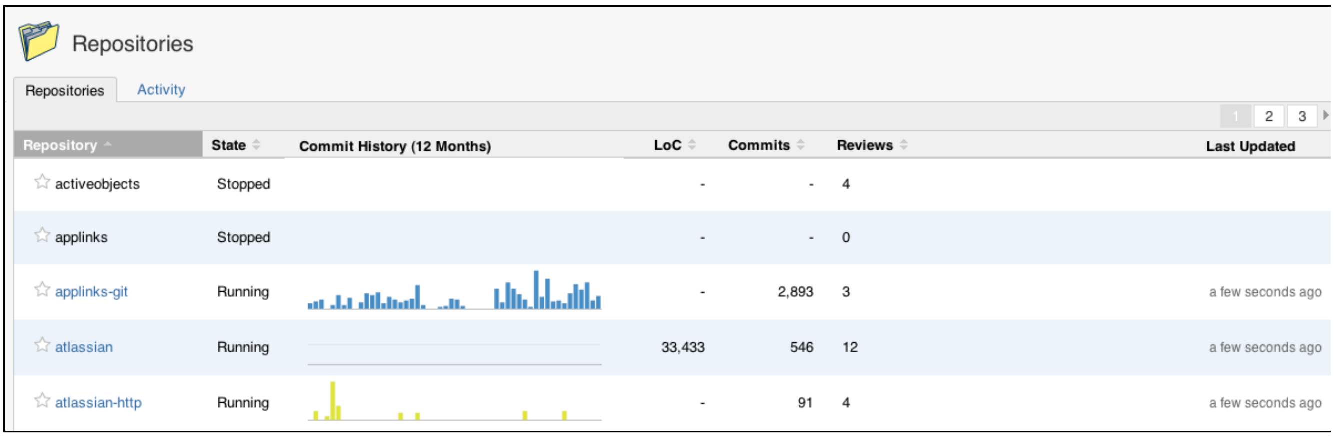
1. Choose **Repositories** > **All repositories** from the header. The 'Repositories' view will be displayed, showing summary information if you have multiple repositories set up. See the 'Viewing all repositories' screenshot below.
2. Click the name for a repository to view its contents. See the 'Viewing a repository' screenshot below.
3. Browse the repository for the desired source file using the directory tree in the left menu. See the 'Viewing a file' screenshot below.
4. You can view various information about the file:

Tab	Description

Activity	Shows recent activity for the item. There are a number of sub-options here: **All** — The default view, showing commits, reviews and JIRA issues. **Commits** — Shows commits in the activity stream. **Reviews** — Shows review activity in the activity stream. **Filter commits** — Applies constraints to the current activity stream. **Expand all** – Shows more detail for all changesets. **Scroll to changeset** — Displays the changeset ID specified
Revisions	When viewing a file, shows the latest revisions of the file.
Users	Shows the commit history of the different users that have committed changes on the item.
Reports	Shows activity charts for the item. Various chart options can be selected in the left navigation bar.
Source	Shows the contents of the file.

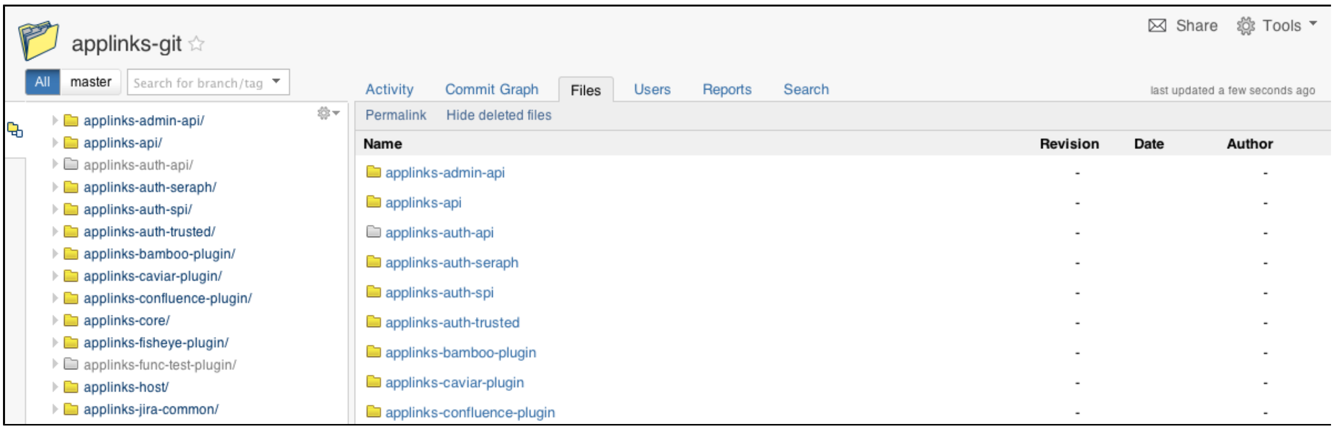
ℹ To download files, click the **Source** tab for the desired file, then right-click **Raw**.

Screenshot: Viewing all repositories



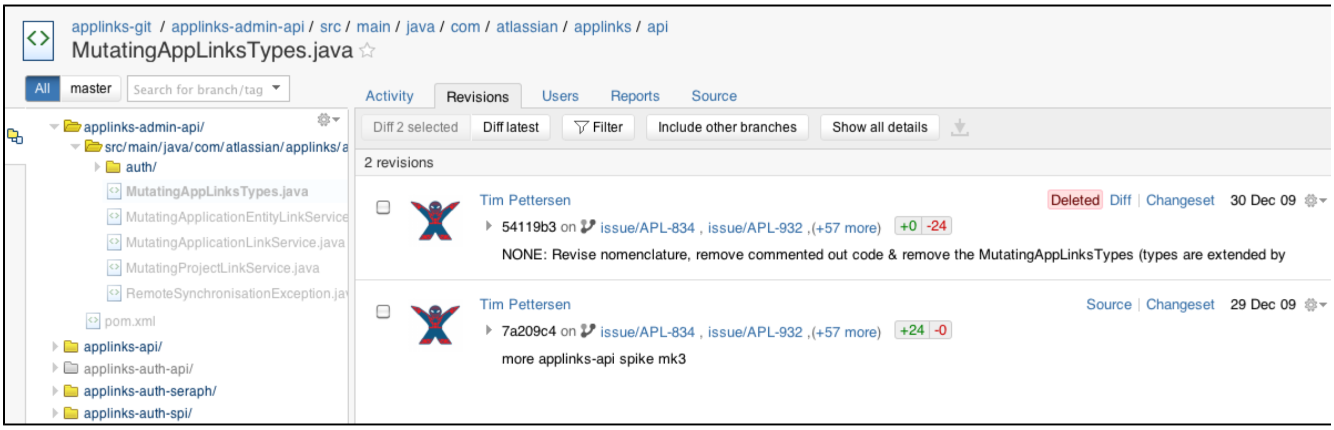
Repository	State	Commit History (12 Months)	LoC	Commits	Reviews	Last Updated
☆ activeobjects	Stopped		-	-	4	
☆ applinks	Stopped		-	-	0	
☆ applinks-git	Running		-	2,893	3	a few seconds ago
☆ atlassian	Running		33,433	546	12	a few seconds ago
☆ atlassian-http	Running		-	91	4	a few seconds ago

Screenshot: Viewing a repository



Name	Revision	Date	Author
applinks-admin-api	-		-
applinks-api	-		-
applinks-auth-api	-		-
applinks-auth-seraph	-		-
applinks-auth-spi	-		-
applinks-auth-trusted	-		-
applinks-bamboo-plugin	-		-
applinks-caviar-plugin	-		-
applinks-confluence-plugin	-		-

Screenshot: Viewing a file


applinks-git / applinks-admin-api / src / main / java / com / atlassian / applinks / api
MutatingAppLinksTypes.java
All
master
Search for branch/tag
Activity
Revisions
Users
Reports
Source
Diff 2 selected
Diff latest
Filter
Include other branches
Show all details
applinks-admin-api/
src/main/java/com/atlassian/applinks/a
auth/
MutatingAppLinksTypes.java
MutatingApplicationEntityLinkService
MutatingApplicationLinkService.java
MutatingProjectLinkService.java
RemoteSynchronisationException.ja
pom.xml
applinks-api/
applinks-auth-api/
applinks-auth-seraph/
applinks-auth-spi/

2 revisions

Tim Pettersen — Deleted Diff | Changeset 30 Dec 09

54119b3 on issue/APL-834 , issue/APL-932 ,(+57 more) +0 -24

NONE: Revise nomenclature, remove commented out code & remove the MutatingAppLinksTypes (types are extended by

Tim Pettersen — Source | Changeset 29 Dec 09

7a209c4 on issue/APL-834 , issue/APL-932 ,(+57 more) +24 -0

more applinks-api spike mk3

Browsing projects

To browse the content in a project, click **Projects** at the top of the page and choose a recent project, or choose **All projects** and click on a project name in the table.

The page for the project has the following sections:

Left navigation panel

Displays an overview of the project's history, statistics and activity.

Activity tab

Lists recent commit and review activity on separate sub-tabs, and all these together on the **All** sub-tab.

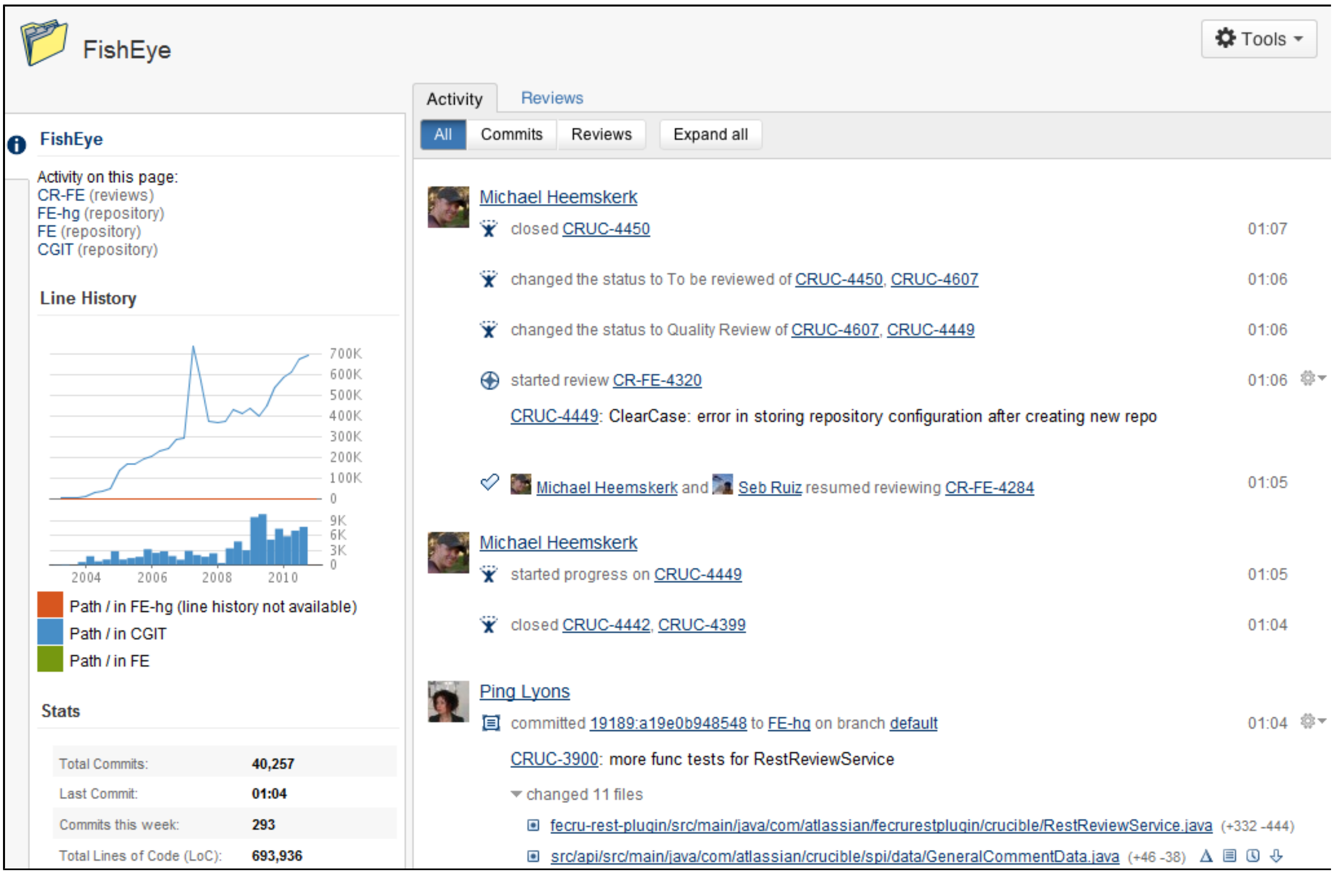
Click **Expand all** to see more detail for every commit or review.

Reviews tab

Lists all the reviews for the project.

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

Screenshot: The Crucible Project View



FishEye

Activity on this page:
CR-FE (reviews)
FE-hg (repository)
FE (repository)
CGIT (repository)

Line History

🟥 Path / in FE-hg (line history not available)
🟦 Path / in CGIT
🟩 Path / in FE

Stats

Total Commits:	40,257
Last Commit:	01:04
Commits this week:	293
Total Lines of Code (LoC):	693,936

Michael Heemskerk

closed CRUC-4450 01:07

changed the status to To be reviewed of CRUC-4450, CRUC-4607 01:06

changed the status to Quality Review of CRUC-4607, CRUC-4449 01:06

started review CR-FE-4320 01:06

CRUC-4449: ClearCase: error in storing repository configuration after creating new repo

Michael Heemskerk and Seb Ruiz resumed reviewing CR-FE-4284 01:05

Michael Heemskerk

started progress on CRUC-4449 01:05

closed CRUC-4442, CRUC-4399 01:04

Ping Lyons

committed 19189:a19e0b948548 to FE-hg on branch default 01:04

CRUC-3900: more func tests for RestReviewService

changed 11 files

fecru-rest-plugin/src/main/java/com/atlassian/fecrurestplugin/crucible/RestReviewService.java (+332 -444)

src/api/src/main/java/com/atlassian/crucible/spi/data/GeneralCommentData.java (+46 -38)

Changing your User Profile

See Changing your User Profile in the FishEye documentation.

Using favorites

This page describes how to use 'favorites' in Crucible.

You can add code reviews, people and repositories to your favorites. This allows you to customize the information that you see in your activity stream. Try favoriting items that you are currently working on, to get greater relevance and context in your activity stream.

You can view all your favorites at once in your profile – choose **Favorites** from your User menu (the one with your avatar).

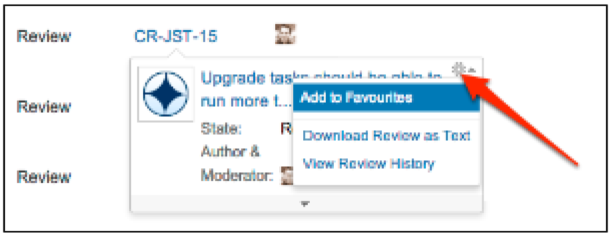
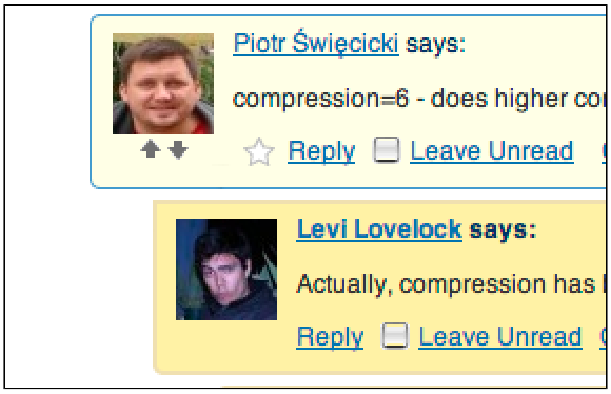
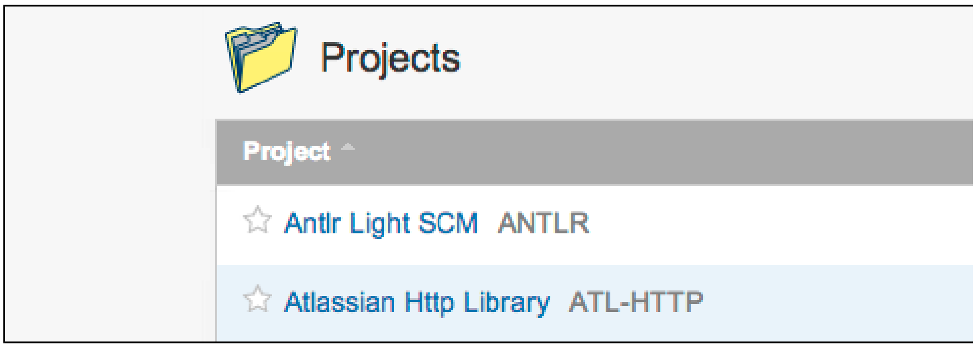
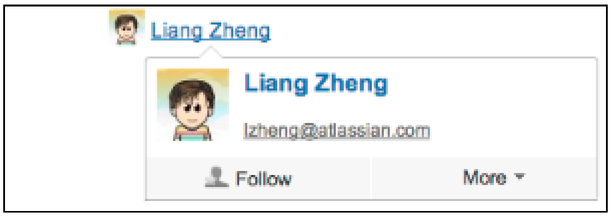
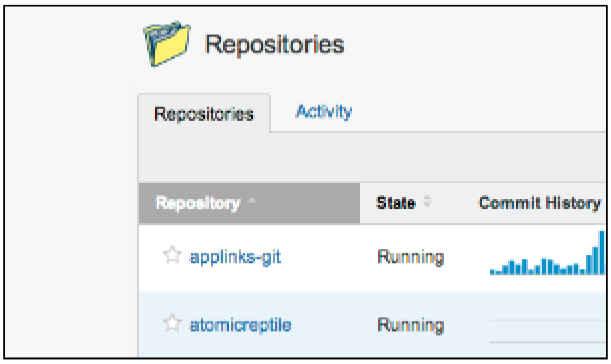
On this page:

- Adding favorites
- Managing favorites

Adding favorites

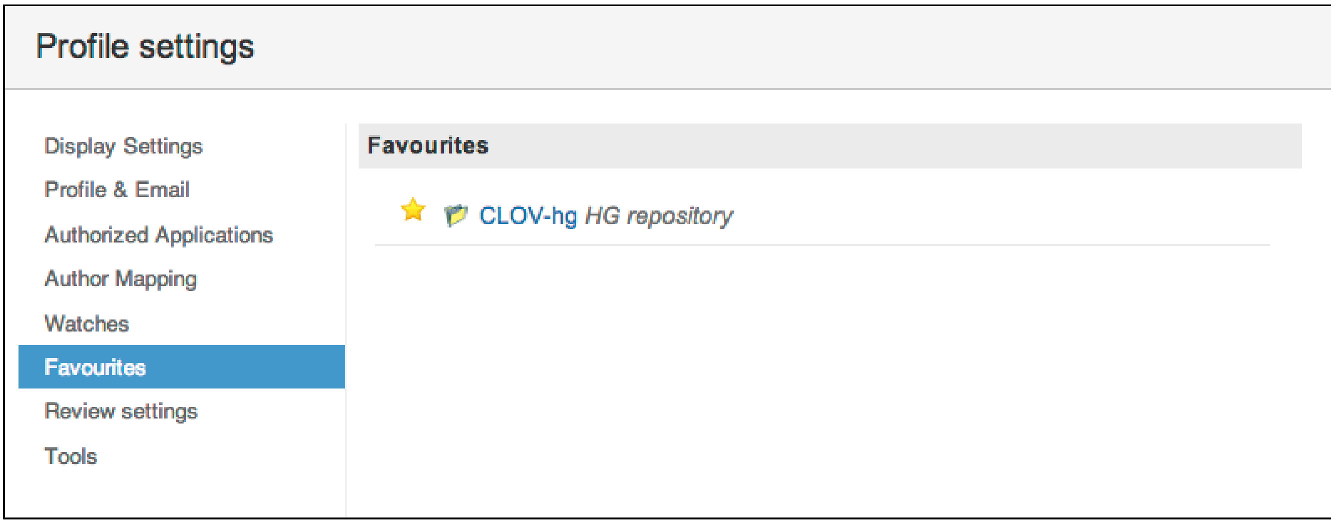
To add an item to your favorites, follow one of these options:

Item	Favorite it by...	Looks like this...

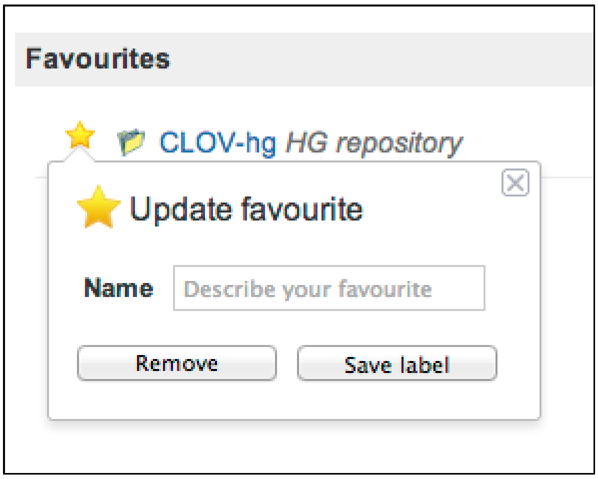
Review	Hover over the review name, for example in the 'Open Reviews' list of the Reviews Dashboard. In the popup, click the cog icon and choose **Add to Favorites**.	 Review CR-JST-15 Review Review Upgrade tas... run more t... State: R Author & Moderator: Add to Favourites Download Review as Text View Review History
Review comment thread	Click the grey star in the first comment of the comment thread.	 Piotr Święcicki says: compression=6 - does higher co ☆ Reply □ Leave Unread Levi Lovelock says: Actually, compression has Reply □ Leave Unread
Project	Click the grey star beside the project's name in the 'All Projects' list. See Browsing projects.	 Projects Project ☆ Antlr Light SCM ANTLR ☆ Atlassian Http Library ATL-HTTP
Person	Hover over a person's username and click **Follow**.	 Liang Zheng Liang Zheng lzheng@atlassian.com Follow More
Repository	(Requires FishEye) In the 'Repositories' list, click the grey star beside the repository's name. See Browsing source files.	 Repositories Repositories Activity Repository State Commit History ☆ applinks-git Running ☆ atomicreptile Running

Managing favorites

You can manage your favorites from your profile in Crucible – choose **Favorites** from your User menu (the one with your avatar):



Click the star beside a favorite to change its label or to delete it:



Using Wiki Markup in Crucible

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

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

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

Headings

Notation	Description
h1.Biggest heading	Turns text into a heading at size 1. **Biggest Text**

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

Text Effects

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

Notation	Description
bold	Makes text appear **bold**.
italic	Makes text appear in *italics*.
+underline+	Makes text appear underlined.
??citation??	Makes text appear in —*citation* form.
-strikethrough-	Makes text appear ~~struck-through~~.
^superscript^	Makes text appear in $^{\text{superscript}}$.
~subscript~	Makes text appear in $_{\text{subscript}}$.
`{{monospaced}}`	Placing double curly-brackets around text makes it appear `monospaced`.
bq. Block Quote	To make an entire paragraph into a block quotation, place "bq. " before it. Example: *Some block quoted text*

{quote} here is quoteable content to be quoted {quote}	Quote a block of text that's longer than one paragraph. Example: *here is quoteable* *content to be quoted*

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

Text Breaks

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

Notation	Description
(empty line)	Produces a new paragraph
`\\`	Creates a line break.
`----`	Creates a horizontal ruler.
`---`	Produces em dash — symbol.
`--`	Produces en dash – symbol.

Links

Creating links is easy with Wiki Markup.

Notation	Description
[Crucible Review CR-FE-100|CR-FE-100]	Creates a link to a Crucible review or FishEye artifact using the internal key reference for the item.
[Atlassian Crucible|http://atlassian.com]	Creates a link to an external resource, special characters that come after the URL and are not part of it must be separated with a space. External links are denoted with an arrow icon. Examples: • http://www.atlassian.com/crucible • Atlassian Crucible Note: The square brackets [,], around external links are optional in the case you do not want to use any alternate text for the link (i.e. just display the raw URL).

[mailto:mail@example.com]	Creates a link to an email address. Example: mail@example.com
[file:///c:/temp/foo.txt] [file:///z:/file/on/network/share.txt]	Creates a download link to a file on your computer or on a network share that you have mapped to a drive. To access the file, you must right click on the link and choose "Save Target As".
{anchor:anchorname}	Creates a bookmark anchor inside the page. You can then create links directly to that anchor. So a link like this: [My Page#here] will link to wherever in "My Page" there is an {anchor:here} macro, and the link [#there] will link to wherever in the current page there is an {anchor:there} macro.

Lists

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

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

Images

Images can be referenced from remote sources only.

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

Tables

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

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

The code above produces a table that looks like this:

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

Advanced Formatting

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

Notation	Description

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

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

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

Miscellaneous Markup Features

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

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

Notation	:)	:(	:P	:D	;)	(y)	(n)	(i)	(/)	(x)	(!)
Image	🙂	☹	😛	😃	😉	👍	👎	ℹ	✓	✖	⚠

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

Using RSS feeds in Crucible

Subscribing to an RSS feed

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

Right-click the RSS icon and choose **Copy Link Address** to get the URL that you can paste into your RSS reader of choice.

Click the RSS icon to see a page with the RSS feed displayed.

Using keyboard shortcuts in Crucible

To see the available shortcuts, navigate to a review in Crucible, then choose **Tools** > **Keyboard Shortcuts**.

General shortcuts

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

Custom navigation

Key	Function
,	(Comma) Go to the previous element (file, comment, defect or diff hunk depending on your current context)
.	(Period) Go to the next element (file, comment, defect or diff hunk depending on your current context)

Comment navigation

Key	Function
n	Go to next review comment
p	Go to previous review comment
shift + p	Go to first review comment
shift + n	Go to last review comment
l	Go to next thread (skips replies)
h	Go to previous thread (skips replies)
]	Go to next unread comment
[	Go to previous unread comment
r	Reply to a comment
m	Toggle comment read/unread status

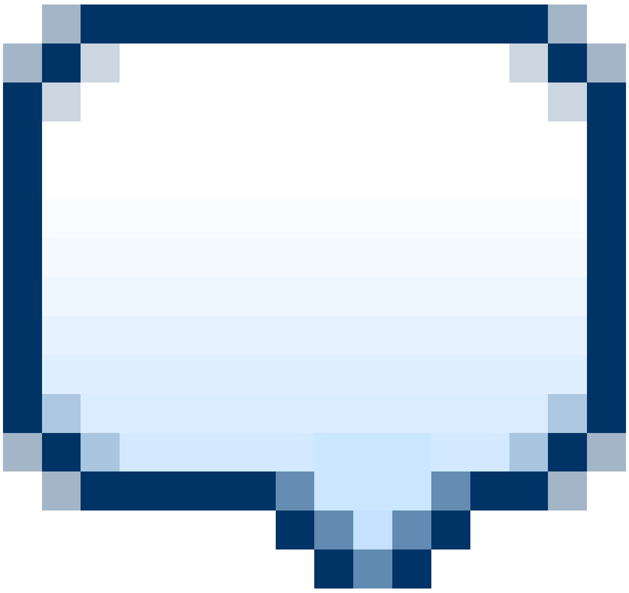
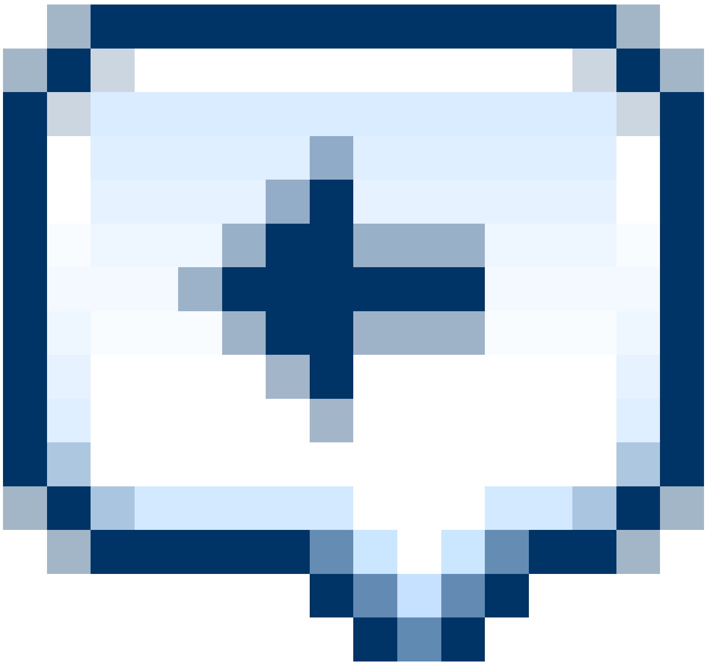
File navigation

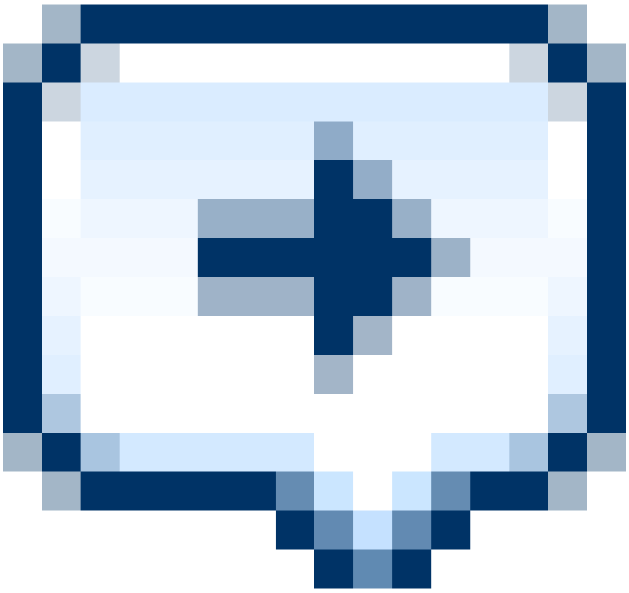
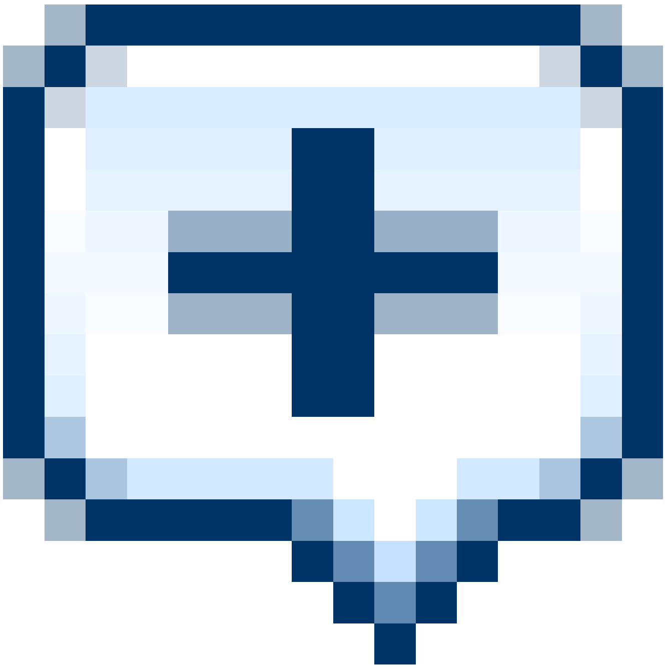
Key	Function
j	Go to the next element (file, comment, defect or diff hunk depending on your current context)
k	Go to the previous element (file, comment, defect or diff hunk depending on your current context)
shift + k	Go to first file
shift + j	Go to last file
u	Go to next unreviewed file
i	Go to previous unreviewed file

y	Set file reviewed and go to next unreviewed file
shift + y	Toggle file reviewed/unreviewed status
e	Expand current file
c	Collapse current file
shift + e	Expand all files
shift + c	Collapse all files

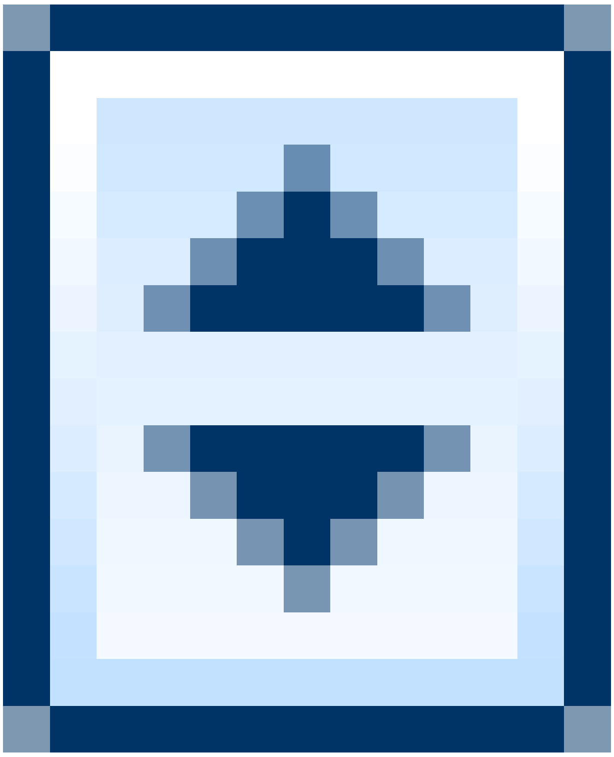
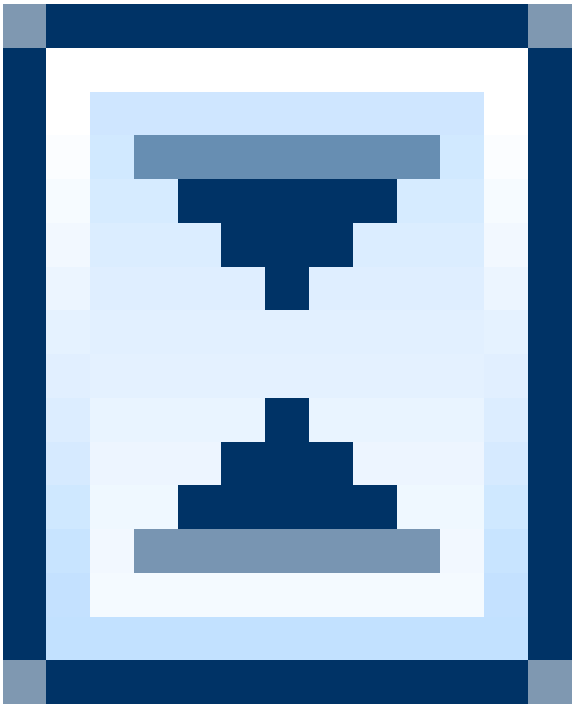
Crucible icons

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

Icon	Description
	View review-level comments
	Go to the previous comment

	Go to the next comment
	Add a comment

←	Go to the previous file in this review
→	Go to the next file in this review

	Expand all files
	Collapse all files

📄	A file included in this review
📁	A directory included in this review

The Crucible workflow

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

On this page:

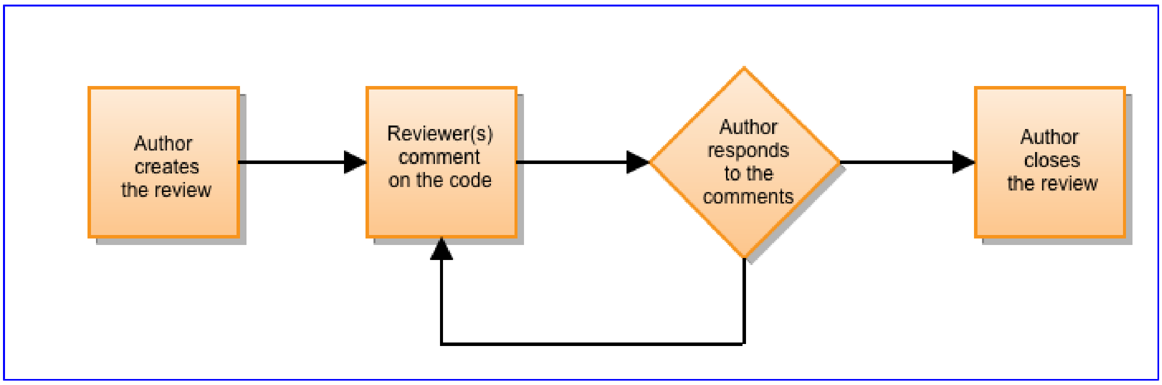
Roles
Crucible is a flexible application that caters for a wide range of team sizes and work styles. You will need to know about the basic roles used in Crucible:

- authors: Usually the creator of the code; the person who will act on the review's outcome.
- reviewer: A participant that will comment on the source files in the review, raising points and discussion on the work that was done.
- moderator: Usually the person who starts the review and is responsible for deciding the outcomes and closing it. The moderator is disabled for the "agile" permission scheme to simplify workflow.

Crucible Workflow

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

Diagram: Workflow for One-to-One Reviews


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

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

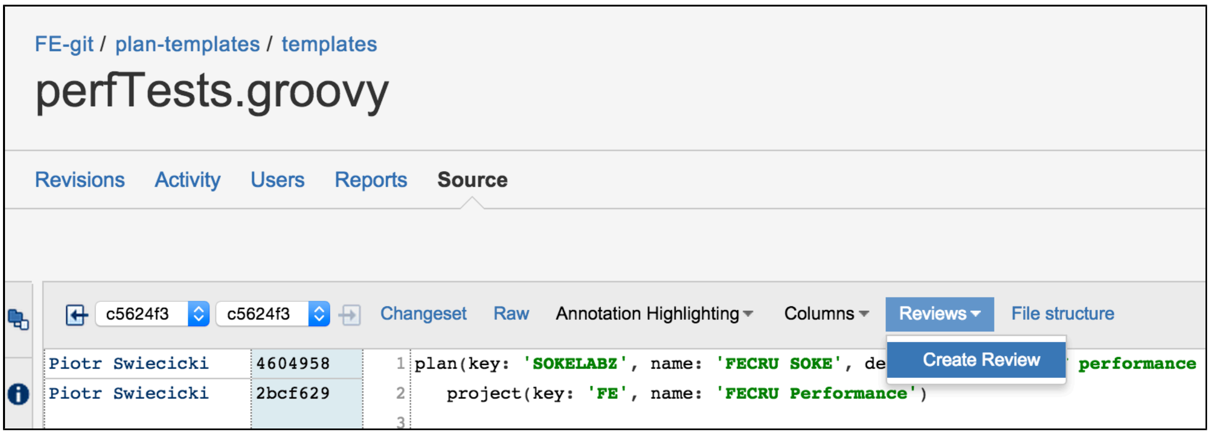
Next, we explore an example workflow for a two-person code review in Crucible.

Example workflow: Two participant code review

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

1. The code author starts the review

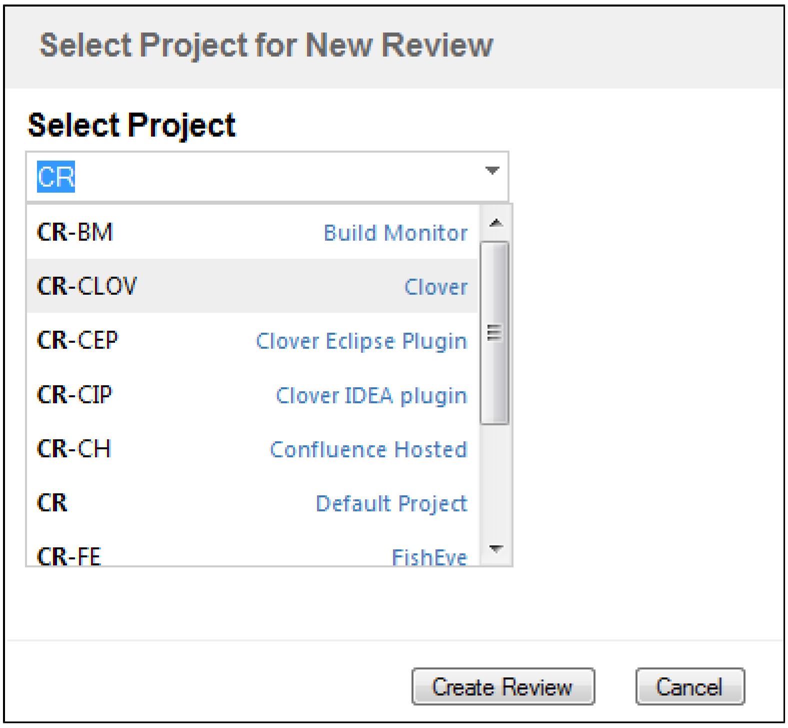
To begin, the code author sets up the review. There are a number of ways to do this, but for this example, you start from the FishEye source view of the file you want to review:



```
Piotr Swiecicki  4604958   plan(key: 'SOKELABZ', name: 'FECRU SOKE', de                  ' performance
Piotr Swiecicki  2bcf629      project(key: 'FE', name: 'FECRU Performance')
```

In the source view, choose **Reviews** > **Create Review**.

If there are multiple projects, the Select Project dialog opens:



Select Project for New Review

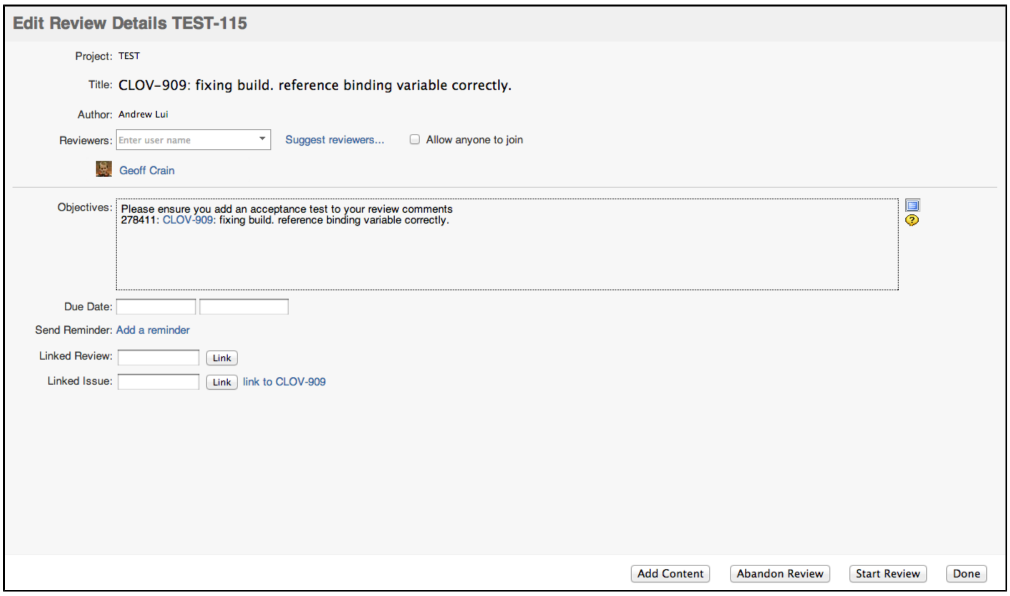
Select Project

CR

CR-BM	Build Monitor
CR-CLOV	Clover
CR-CEP	Clover Eclipse Plugin
CR-CIP	Clover IDEA plugin
CR-CH	Confluence Hosted
CR	Default Project
CR-FE	FishEye

Choose a project for this review from the drop-down list, then click **Create Review**.

The Edit Review Details dialog opens, you can create and issue the review:



Edit Review Details TEST-115

Project: TEST

Title: CLOV-909: fixing build. reference binding variable correctly.

Author: Andrew Lui

Reviewers: Enter user name

Geoff Crain

Objectives: Please ensure you add an acceptance test to your review comments
278411: CLOV-909: fixing build. reference binding variable correctly.

Due Date:

Send Reminder: Add a reminder

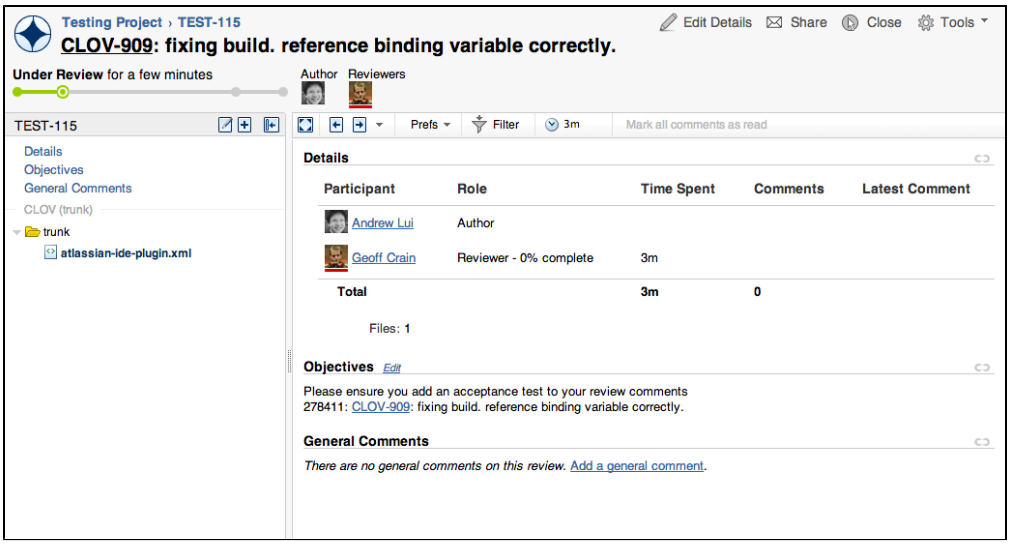
Linked Review:

Linked Issue: link to CLOV-909

In the Edit Review dialog, enter information needed for the review. This includes a title and description for the review, selecting reviewers, a due date and any related JIRA issues. The project and author are pre-selected.

You can also add more content to the review, by clicking **Add Content**. See Adding content to the review.

When ready, click **Done**. The review will now be created as a draft:



Testing Project › TEST-115

CLOV-909: fixing build. reference binding variable correctly.

Under Review for a few minutes

Author Reviewers

Details

Participant	Role	Time Spent	Comments	Latest Comment
Andrew Lui	Author			
Geoff Crain	Reviewer - 0% complete	3m		
Total		3m	0	

Files: 1

Objectives Edit

Please ensure you add an acceptance test to your review comments
278411: CLOV-909: fixing build. reference binding variable correctly.

General Comments

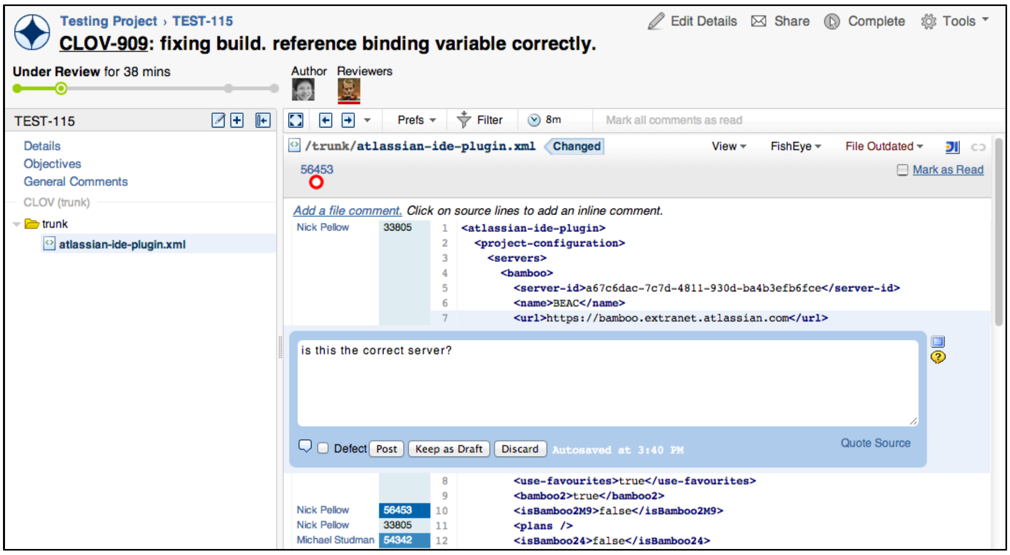
There are no general comments on this review. Add a general comment.

The draft review opens. In the draft stage, you can check the contents of the review files and add notes for reviewers as comments. During the draft phase, no notification emails are sent out to reviewers. Once finished with the draft phase, click **Start Review**.

The review will now be started and a notification email will go out to all 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.

2. The reviewer comments on the code

The reviewer will receive an email from Crucible with a link that they can follow to the review:



Testing Project › TEST-115

CLOV-909: fixing build. reference binding variable correctly.

Under Review for 38 mins

Author Reviewers

Add a file comment. Click on source lines to add an inline comment.

```
Nick Pellow      33805   <atlassian-ide-plugin>
                           <project-configuration>
                             <servers>
                               <bamboo>
                                 <server-id>a67c6dac-7c7d-4811-930d-ba4b3efb6fce</server-id>
                                 <name>BEAC</name>
                                 <url>https://bamboo.extranet.atlassian.com</url>
```

is this the correct server?

Autosaved at 3:40 PM

```
                                 <use-favourites>true</use-favourites>
                                 <bamboo2>true</bamboo2>
Nick Pellow      56453           <isBamboo2M9>false</isBamboo2M9>
Nick Pellow      33805           <plans />
Michael Studman  54342           <isBamboo24>false</isBamboo24>
```

The Review screen displays the source files that are under review. The reviewer clicks file names to see the code to be reviewed. As the reviewer reads the changes, they can add comments:

- Click **Add a general comment** (under 'General Comments' on the Review screen) to comment on the overall review.
- Click **Add a file comment** (just above the source code listing) to add a general comment about a source code file.
- Click on any line in the source file to enter a comment there (multiple lines can be selected by clicking and dragging).

The reviewer clicks **Post** to add the comment to the review.
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.

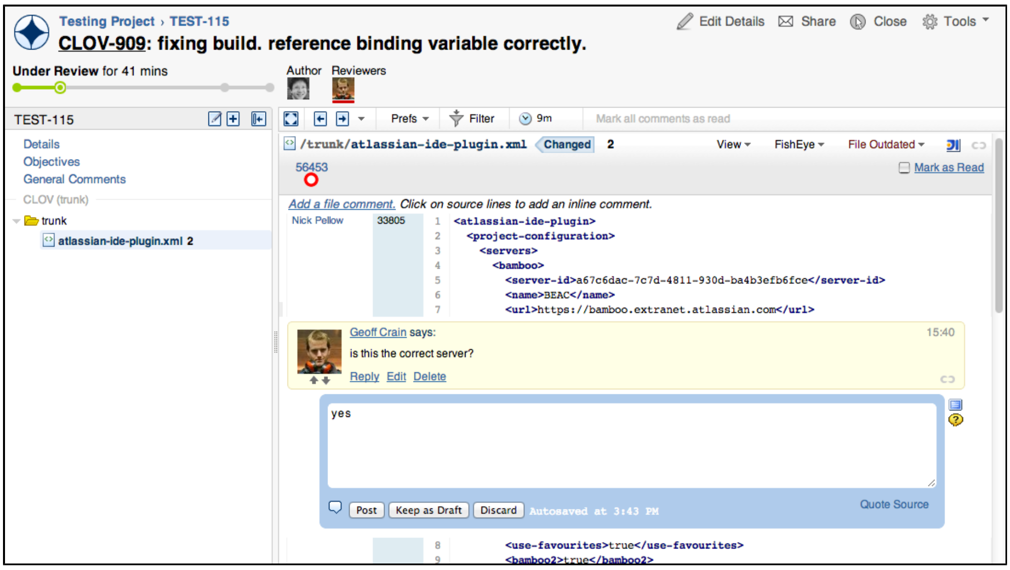
When the reviewer has finished their code review work, they click **Complete**.

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 can also make contributions, responding to reviewer comments and making corrections:



CLOV-909: fixing build. reference binding variable correctly.

Under Review for 41 mins

/trunk/atlassian-ide-plugin.xml

Add a file comment. Click on source lines to add an inline comment.

Nick Pellow 33805

```
<atlassian-ide-plugin>
  <project-configuration>
    <servers>
      <bamboo>
        <server-id>a67c6dac-7c7d-4811-930d-ba4b3efb6fce</server-id>
        <name>BEAC</name>
        <url>https://bamboo.extranet.atlassian.com</url>
```

Geoff Crain says: 15:40

is this the correct server?

yes

Autosaved at 3:43 PM

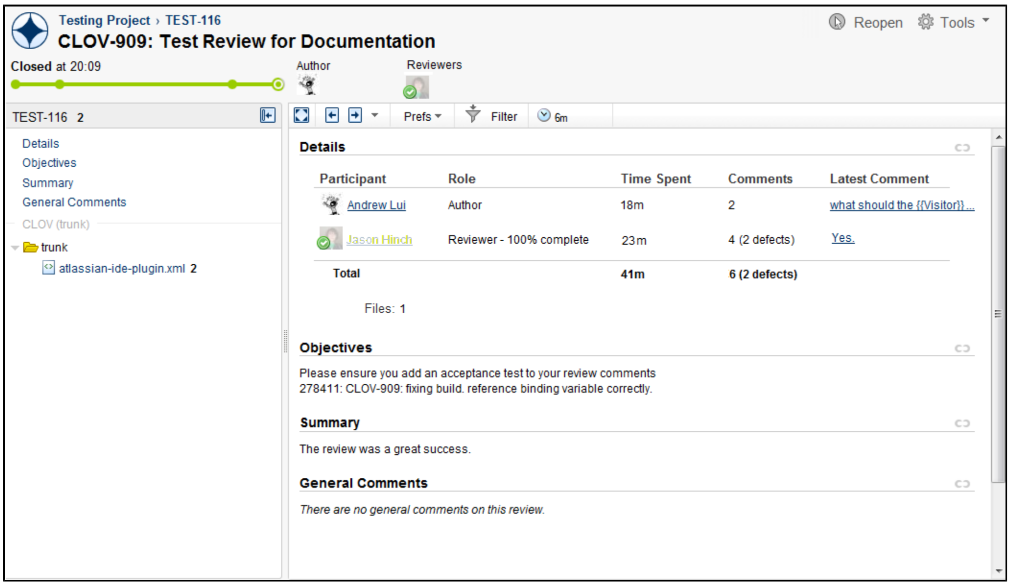
```
        <use-favourites>true</use-favourites>
        <bamboo2>true</bamboo2>
```

4. The author closes the review

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

The author can then add any final comments, and click **Close** when finished.

This closes the review, signaling the end of work. A final email notification will be sent to the review participants, informing them that the review is now closed. The closed review screen will load, archiving the completed review as read-only.



CLOV-909: Test Review for Documentation

Closed at 20:09

Details

Participant	Role	Time Spent	Comments	Latest Comment
Andrew Lui	Author	18m	2	what should the {{Visitor}} ...
Jason Hinch	Reviewer - 100% complete	23m	4 (2 defects)	Yes.
Total		41m	6 (2 defects)	

Files: 1

Objectives

Please ensure you add an acceptance test to your review comments
278411: CLOV-909: fixing build. reference binding variable correctly.

Summary

The review was a great success.

General Comments

There are no general comments on this review.

If the author ever needs to resume work on the closed review, they can simply click **Reopen** when viewing this screen. This returns the status of the review to "Open", without changing the staus of existing reviewers. Click **E dit Details** to add reviewers or to change other details of the review.

Defining your workflow

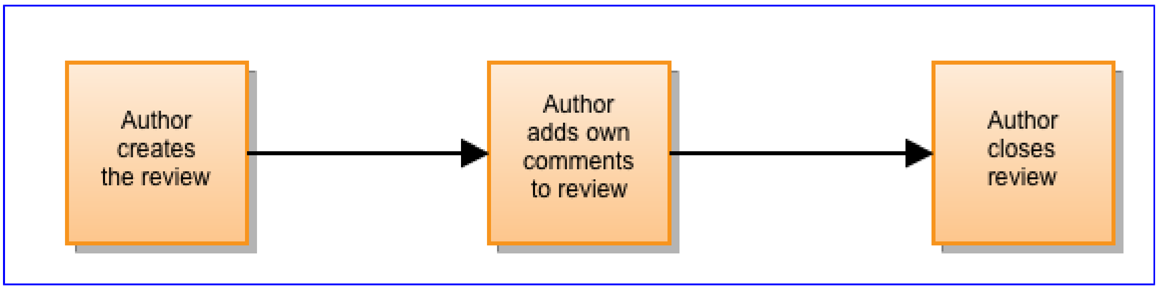
This page provides brief outlines of three different ways that a development team could use Crucible for code reviews:

- Lightweight code commenting with Crucible (individual)
- One-to-many reviews without a moderator (Agile team)
- Formal group reviews (CMM team)

For a detailed example of how to perform Crucible reviews with two participants, see The Crucible workflow.

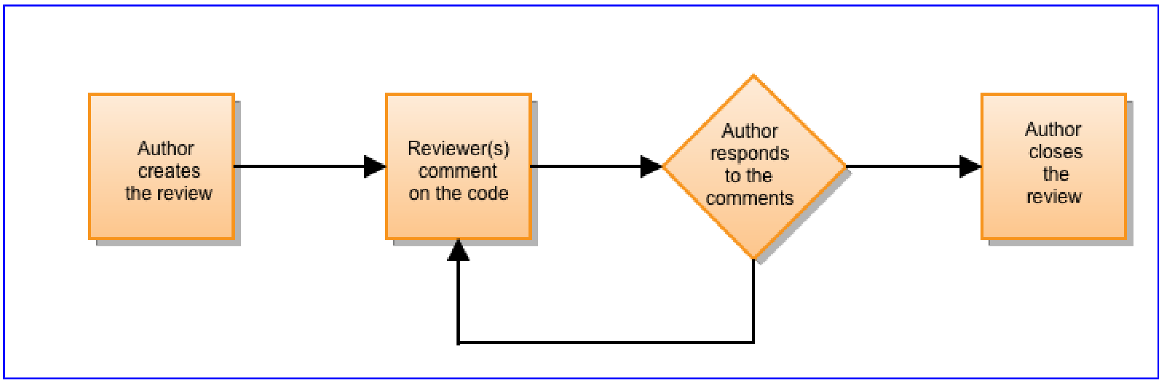
Lightweight code commenting with Crucible (individual)

1. Author commits new work.
2. Author creates the review, and adds comments using the easy web interface.
3. Author summarizes and closes the review, saving the code comments in Crucible (and not in the code repository).



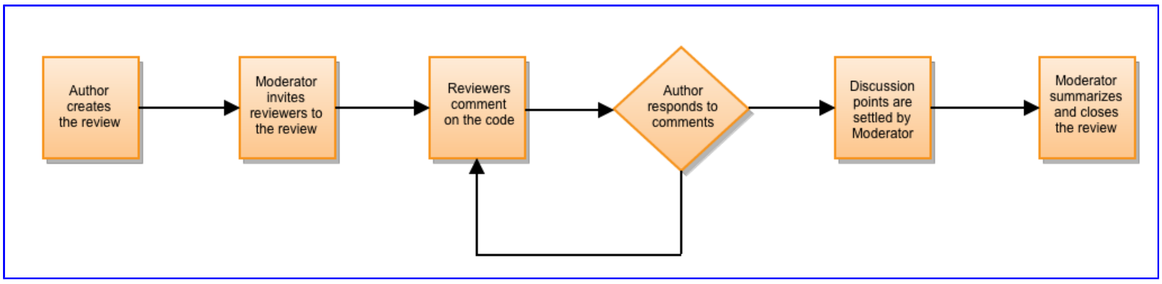
One-to-many reviews without a moderator (Agile team)

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


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

Formal group reviews (CMM team)

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


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

Roles and status classifications

This page explains the roles and status classifications in Crucible.

Roles

Author

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

Creator/Moderator

The *creator* is the person who creates the review. In most cases this person will also act as moderator. The *moderator* is the person responsible for creating the review, approving the review, determining when reviewing is finished, summarizing the outcomes and closing the review. By default, the moderator is the creator. See also author, the person whose changes to the code are to be reviewed.

Reviewer

A *reviewer* is a person assigned to review the change. Reviewers can make comments and indicate when they have completed their review. The moderator and author are implicitly considered to be participants of the review, but are not reviewers.

A Crucible admin can specify *default* reviewers, who are added automatically to all reviews in a project, and can also restrict participation in a project's reviews to just *allowed* users. See Editing a project.

User

A *user* is a person using Crucible.

Status classifications

Draft

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

Under Review

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

Summarized

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

Closed

Closed reviews are complete.

Abandoned

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

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

Creating a review

This page provides an overview of the steps to create a review in Crucible:

1. Create a review
2. Add content to the review
3. Choose the reviewers
4. Complete other details for the review
5. Start the review

See the considerations for when creating reviews at the end of this page.

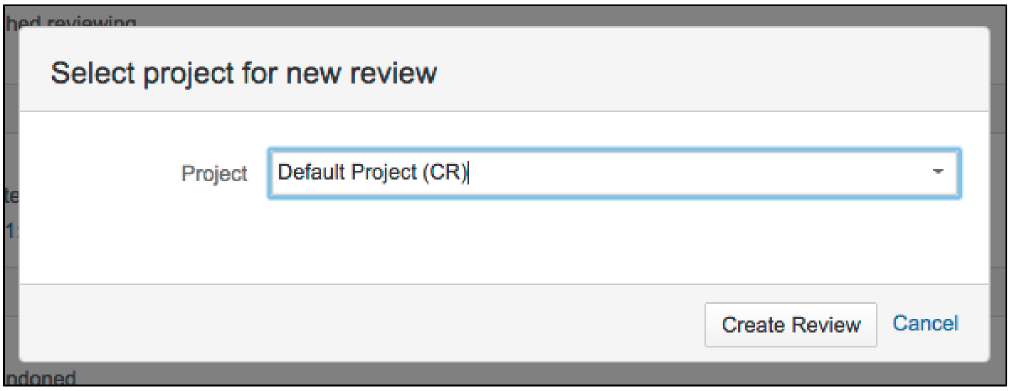
Note that only people with the 'Create' permission can create a review.

You can also create reviews by:

- Creating a review from JIRA
- Creating a review from a URL
- Creating a Snippet Review
- Creating reviews from the command line

Create a review

Begin by clicking **Create review** in the header and picking the project for the review (if you have multiple projects):



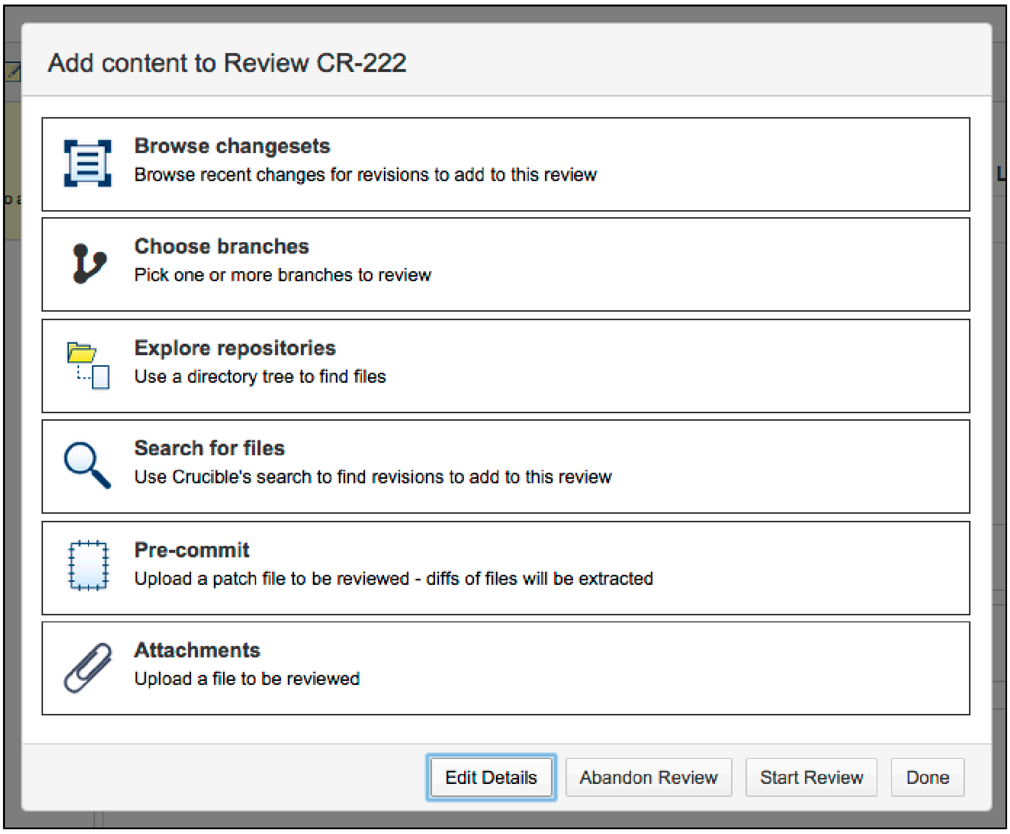
Select project for new review

Project: Default Project (CR)

Click **Create Review**.

Add content to the review

Click one of the content types to browse or search for files, branches and changesets you want to be reviewed:



Add content to Review CR-222

Browse changesets
Browse recent changes for revisions to add to this review

Choose branches
Pick one or more branches to review

Explore repositories
Use a directory tree to find files

Search for files
Use Crucible's search to find revisions to add to this review

Pre-commit
Upload a patch file to be reviewed - diffs of files will be extracted

Attachments
Upload a file to be reviewed

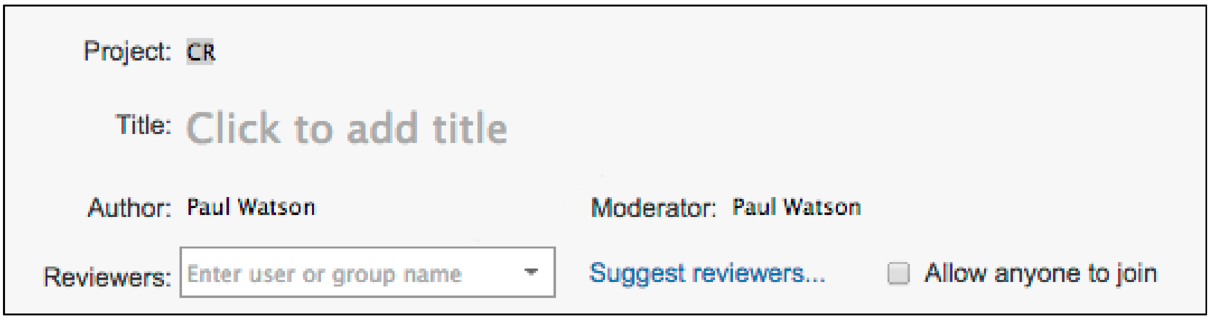
Reviews should be created with care to get the best value from them. We've found it's best not to overload the review – they should be focused on just a few necessary files.

See Adding content to the review for more details.

Click **Edit Details** to choose the reviewers and set other details for the review.

Choose the reviewers

You can choose individuals and groups, or allow anyone to join the review:



Project: CR

Title: Click to add title

Author: Paul Watson

Moderator: Paul Watson

Reviewers: Enter user or group name

We've found it's best not to overcrowd the review – reviewers should be selected with care, and should be guided individually on what to look for.

See Choosing reviewers and Enabling the moderator role for more details.

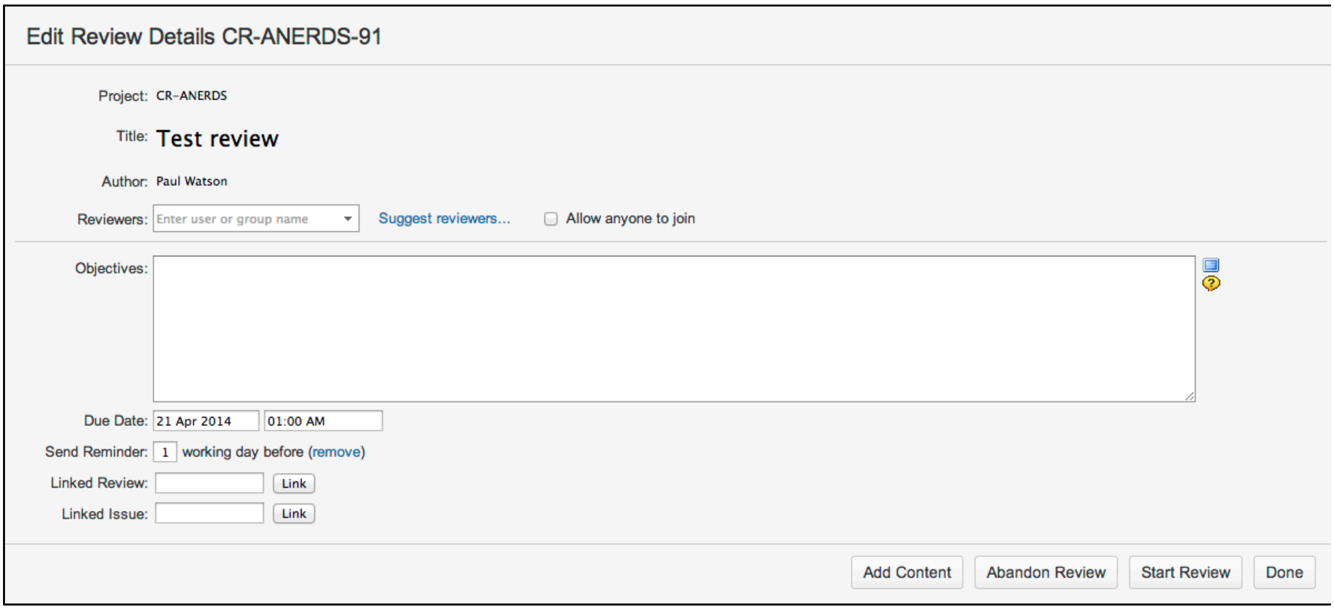
Complete other details for the review

You can set other details for the review, including:

- the title and description
- objectives – used to guide the reviewers on what to look for. See Setting default review objectives.
- a due date
- a reminder date
- linked reviews (if any)

- related JIRA issues (if any).

The project, moderator and author are pre-selected.



Edit Review Details CR-ANERDS-91

Project: CR-ANERDS

Title: Test review

Author: Paul Watson

Reviewers: Enter user or group name | Suggest reviewers... | □ Allow anyone to join

Objectives:

Due Date: 21 Apr 2014 | 01:00 AM

Send Reminder: 1 working day before (remove)

Linked Review: | Link

Linked Issue: | Link

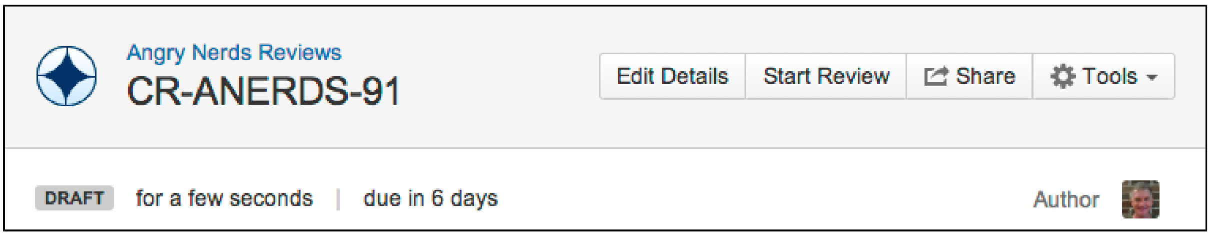
If you're ready, click **Start Review** to make the review available to your reviewers.

If you're not ready to start the review, click **Done**. You can come back and work on the draft review some more, later.

Start a draft review

The review will be displayed in draft mode. Here, you can check and edit the details as required.

Click **Start Review** to make the review available. See Performing the review.



Angry Nerds Reviews

CR-ANERDS-91

DRAFT for a few seconds | due in 6 days

When all the reviewers have performed their reviews, you can summarize and close the review.

Considerations when creating reviews

Review effectiveness

Based on our own experience of over 13000 reviews, we have found that reviews with fewer files and reviewers are more effective. We have seen effects such as:

- Time spent reviewing each file decreases as the number of files increases.
- Reviewers spend less time reviewing as the number of reviewers increases.
- Reviewers find fewer defects as the number of reviewers increases.

We recommend that reviews be created with care to get the best value from them:

- Avoid overloading the review. Reviews should be focused on just a few necessary files.
- Avoid overcrowding the review. Reviewers should be selected with care, and should be guided individually on what to look for.

Crucible performance

The performance of a Crucible instance can be seriously degraded if very large reviews are created.

To prevent a user from accidentally causing this, Crucible has a limit on the review content size when creating reviews. The limit is 800 file revisions.

Each version of a file in a review counts as one revision – so when a review is created for a single modified file, that is two revisions. Each subsequent change to the file you add to the review is one more revision. A 'whole file' in a review is only one revision.

If you really need to create a larger review, you can get your system administrator to set the `crucible.review.content.size.limit` property as described on the JVM system properties page, but remember that performance will be poor when creating and viewing very large reviews.

Adding an entire directory's contents to a Crucible review

To add an entire directory's contents to a Crucible review, you will need to search to find all the files, for example using "select revisions from dir /some/dir where is head and not is deleted", or similar logic.

ⓘ It is currently not possible in Crucible to add all the contents of a directory to a review with one click.

Creating a review from JIRA

This page describes how to create a Crucible review directly from an issue in JIRA, the Atlassian issue-tracking application.

? JIRA must be integrated with Crucible before you can do this. See Linking Crucible to JIRA for information on how to set up an application link with JIRA.

See also Creating JIRA issues from the review.

When using Crucible with JIRA 6.2.x and later

If your instance of Crucible (version 3.3 or later) is linked to JIRA 6.2 (or later), then you can start creating a review from a JIRA issue.

To create a review from a JIRA issue:

1. Go to the JIRA issue that relates to the work to be reviewed.
2. Click **Commits** in the Development panel.
3. Click the **FishEye / Crucible** tab.
4. Start the process for creating a review for either a single commit, or for all the commits related to the JIRA issue.
5. In Crucible, the new review is in edit mode:
   - The content of the changeset becomes the content (i.e. files) to be reviewed.
   - The author of the commit 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.
6. Choose **Tools** > **Start Review**, in Crucible, when you are ready.

When using Crucible with JIRA 6.1.x and earlier

If you have Crucible linked to a version of JIRA earlier than 6.2 (or if you have instances of Crucible 3.2, or earlier, linked to JIRA 6.2, or later), the integration functionality continues to behave as previously.

Click here if you're using JIRA 6.1 or earlier...

When Crucible is linked with JIRA 6.1.x, or earlier, Crucible content appears on the **Reviews** tab (and FishEye content appears on the **Source** tab) in JIRA.

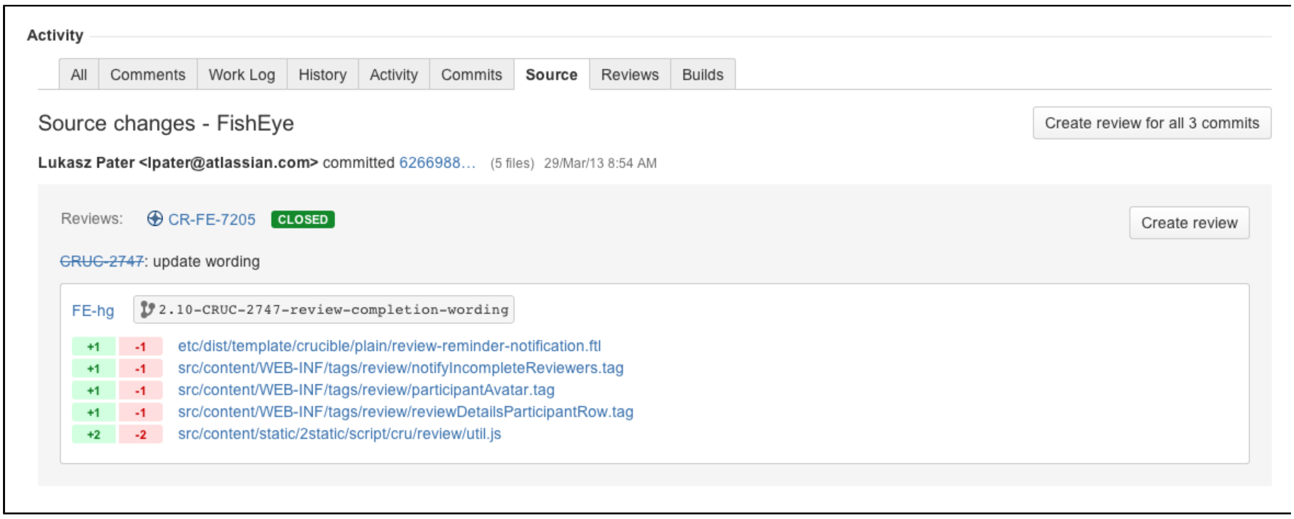
To create a review from a JIRA issue:

1. Go to the issue in JIRA that relates to the work to be reviewed.

2. Under 'Activity', click the **Source** tab.
3. Either:
   a. Click **Create review** to create a new review for a particular changeset.
   b. Click **Create review for all commits** to include all changesets from the JIRA issue in the new review.
4. If a similar review already exists, you can add the changesets to that.
5. In Crucible, the new review is in edit mode:
   - 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.
6. Choose **Tools** > **Start Review**, in Crucible, when you are ready.

The next step is to add reviewers.

Screenshot: Adding a review from within JIRA



Source changes - FishEye

Lukasz Pater <lpater@atlassian.com> committed 6266988… (5 files) 29/Mar/13 8:54 AM

Reviews: CR-FE-7205

~~CRUC-2747~~: update wording

FE-hg 2.10-CRUC-2747-review-completion-wording

+1	-1	etc/dist/template/crucible/plain/review-reminder-notification.ftl
+1	-1	src/content/WEB-INF/tags/review/notifyIncompleteReviewers.tag
+1	-1	src/content/WEB-INF/tags/review/participantAvatar.tag
+1	-1	src/content/WEB-INF/tags/review/reviewDetailsParticipantRow.tag
+2	-2	src/content/static/2static/script/cru/review/util.js

Creating a review from a URL

You can set up a URL that 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%2Fre
pob%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 projects if more than one project exists) in which to create your review. A new draft review will then be created, including the following information:

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

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

The next step is to add reviewers.

Creating a Snippet Review

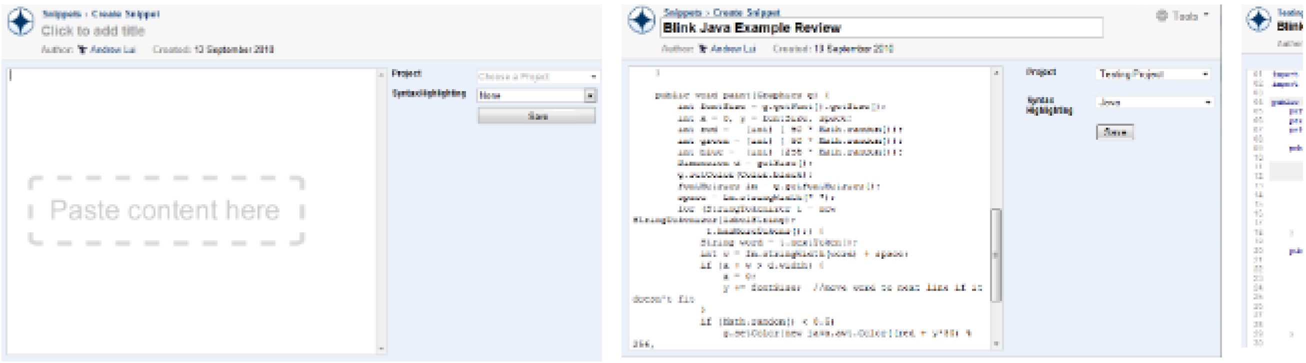
This page explains how to create a simple code review using the Crucible Snippet Review feature. Snippet Reviews are designed to be lightweight ad-hoc code reviews.

To create a snippet review:

1. Copy the code to be reviewed from the source to your system clipboard.
2. Click **Create snippet** from the **Create review** menu in the Crucible toolbar.
3. Enter details for the snippet review:
    - Paste the code into the panel, where indicated.
    - Click on **Click to add title** near the top to enter a title for your review. If you don't specify a title, one will be automatically created for you.
    - Select a project from **Project**.
    - Select a programming language from **Syntax Highlighting**.
4. Click **Save** to create the snippet review.
5. Invite anyone that you want to participate in the snippet review by sending them the link to the review. The link is the review key, just above the review title. Anyone who is allowed to view the snippet is allowed to comment on it, and can close it.
6. Click **Reply** on any comments to respond.
7. Choose from the **Tools** menu to either close or delete the snippet review. Anyone can re-open, re-review or close snippet reviews, however, only the creator of a snippet review can delete it.

You can see your own snippets, or everyone's snippets, by choosing **Reviews** > **Review dashboard**. See Searching Crucible for information about filtering snippet reviews.

Screenshots: Creating a Snippet Review (click to view larger images)



Step 1

Step 2

Creating reviews from the command line

You can use the Review CLI tool to create reviews in Crucible, for patches and commits, directly from your terminal. It takes the uncommitted changes in your workspace and creates a review for them in Crucible. The Review CLI tool may be especially useful if:

- you often create pre-commit reviews
- you want to submit a diff from an external tool for review

You can use the tool on Windows, Linux and Mac OS X, for repositories that are managed in:

- Subversion
- Perforce
- Git
- Mercurial
- CVS

The tool supports Crucible 3.0, and later versions. Python 2.7 must be installed on your local machine (Python 3 is not supported).

You can use the tool to create new reviews, and to update existing reviews with new patches.

Known limitations

- The CLI tool does not yet allow you to create a review for an existing commit.
- The CLI tool takes all the files that have been modified, added or removed in the working copy and submits those for review. It doesn't support choosing only files associated with a specific changeset.
- You can only use the CLI tool to create reviews in projects for which you already have review create permission.

On this page:

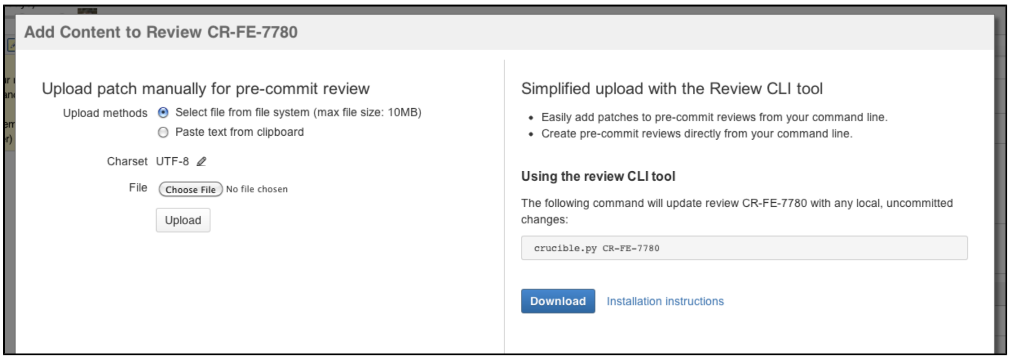
Related pages:

- Creating a review
- Adding content to the review

Installing the Review CLI tool

Download the Review CLI tool from either of the following locations in Crucible:

- When creating a review, click **Pre-commit** in the 'Add Content to Review' dialog, then click **Download**:



Add Content to Review CR-FE-7780

Upload patch manually for pre-commit review

Upload methods ● Select file from file system (max file size: 10MB)
○ Paste text from clipboard

Charset UTF-8

File Choose File No file chosen

Upload

Simplified upload with the Review CLI tool

- Easily add patches to pre-commit reviews from your command line.
- Create pre-commit reviews directly from your command line.

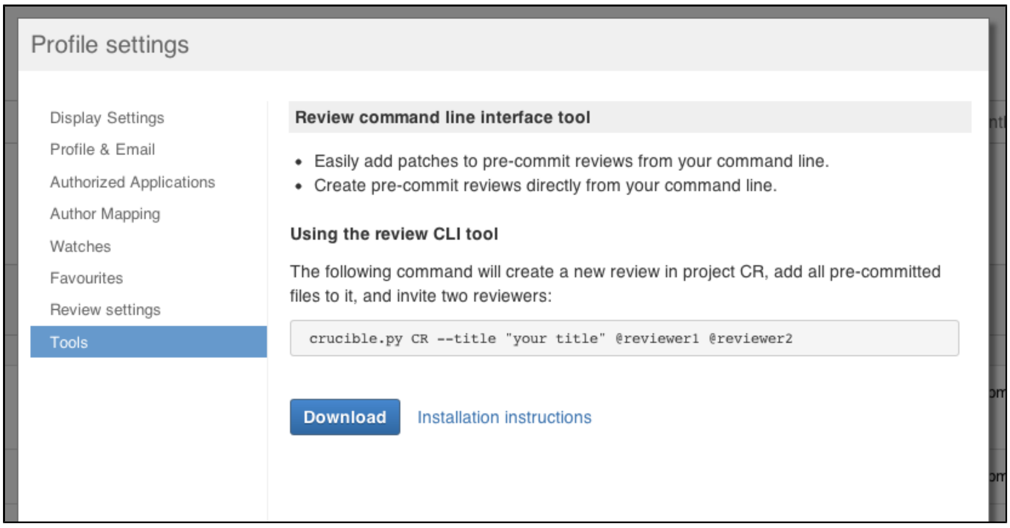
Using the review CLI tool

The following command will update review CR-FE-7780 with any local, uncommitted changes:

```
crucible.py CR-FE-7780
```

Download Installation instructions

- Choose **Profile settings** from your user menu, and then **Tools**. Click **Download**:



Review command line interface tool

- Easily add patches to pre-commit reviews from your command line.
- Create pre-commit reviews directly from your command line.

Using the review CLI tool

The following command will create a new review in project CR, add all pre-committed files to it, and invite two reviewers:

```
crucible.py CR --title "your title" @reviewer1 @reviewer2
```

The script comes pre-configured for the user for the given SCM server.

Copy the script to a location somewhere in your system path, for greatest ease of use.

Python

Python 2.7 is required.

Linux

Python 2.7 should come with your distribution. If not, or if 2.7 isn't installed, you will need to install the appropriate package. Please refer to your package manager for the appropriate version.

Windows

You can install Python by running the latest Python 2.7 Installer for Windows.

MacOS X

Python 2.7 should come with any modern version of MacOS X. You can also download Python 2.7 from http://www.python.org/getit/.

Using the Review CLI tool

To create a review using the Review CLI tool, run the script from a directory under SCM control that has local, uncommitted changes. The changes will be submitted to Crucible for review.

Call the script with the following command:

```
crucible.py <arguments list>
```

The Review CLI syntax is consistent with that for Crucible and FishEye smart commits. Examples of syntax usage are provided in the following table:

Action	Syntax	Result
View usage help	`crucible.py --help`	Displays the help with descriptions of all the valid commandline arguments
Create a pre-commit review interactively	`crucible.py`	Gets the patch from the current SCM, prompts for the project and review title, and creates a draft review

Create a pre-commit review with a given project and title	`crucible.py -m "the review title" CR-FE`	Gets the patch from the current SCM, creates a draft review in the CR-FE project with the given title
Create a pre-commit review with moderator and reviewers	`crucible.py CR-FE @matt @joe --moderator ted`	Gets the patch from the current SCM, creates a review in the CR-FE project, adds matt and joe as the reviewers and ted as a moderator, starts the review
Create a pre-commit review anchored to a specific repository	`crucible.py -r repo1`	Creates a pre-commit review interactively, trying to anchor the patch to the given repository
Add a patch to a review	`crucible.py CR-FE-1204`	Gets the patch from the current SCM and adds it to the review CR-FE-1204
Create a review from diff output	`hg diff | crucible.py`	Creates a pre-commit review interactively, taking the output of the first command as a patch
Create a review from a diff file	`crucible.py -f file.diff`	Creates a pre-commit review interactively, taking the output of the patch from the given file

Adding content to the review

This page explains how to add content, such as files and changesets, to a Crucible review.

We've found that reviews with fewer files and reviewers are more effective. Reviewers spend less time reviewing each file as the number of files increases, so don't overload the review – help your reviewers to focus on just the essential files.

Related pages:

- Creating a review
- Iterative reviews
- Choosing reviewers
- Performing the review

On this page:

Overview

Crucible supports post-commit and pre-commit reviews, depending on the type of content you add to the review:

- Post-commit reviews – for code changes that have *already* been committed to the SCM.
- Pre-commit reviews – for code changes that have *not yet* been committed to the SCM. Create a patch file for the code changes and then add the patch to the review.

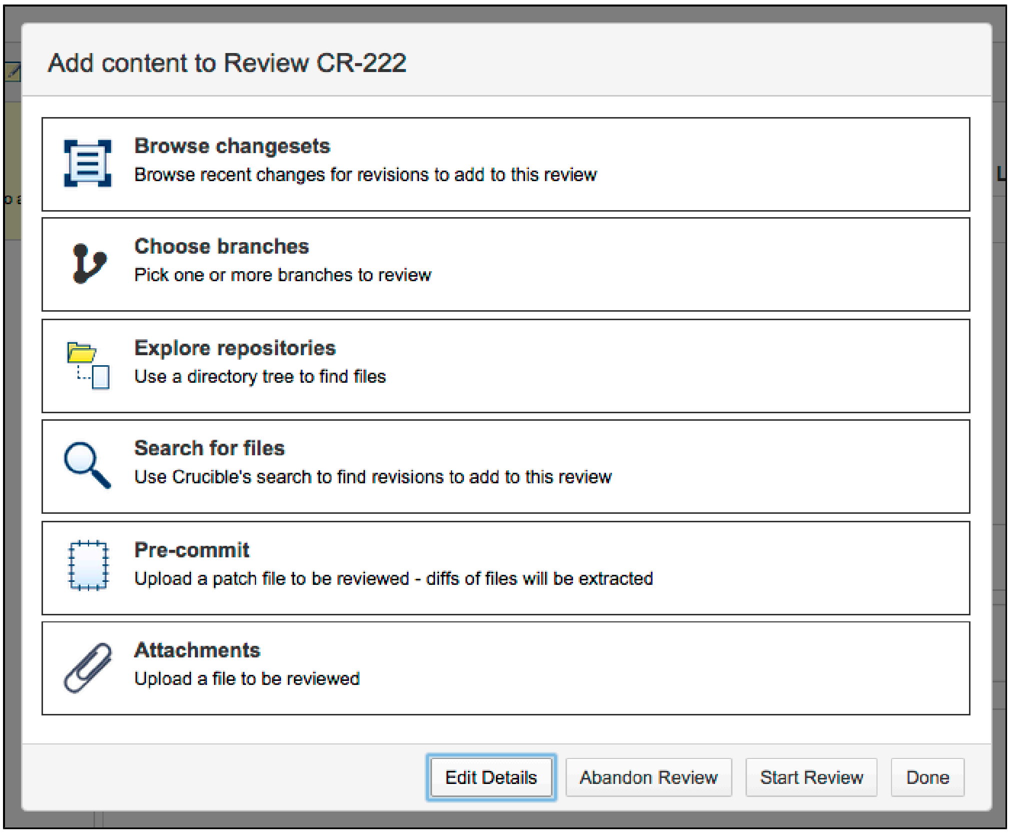
Crucible also supports iterative reviews – you can update the review content with new versions. The reviewer can see the different versions of the files, so they can understand the changes that have been made.

To add content to a review:

1. Log in to FishEye/Crucible and either:
   - Create a new review, as described on Creating a review, or
   - Open an existing review, for which you are the creator or moderator, and click the 'Add content' (⊞) button.

2. In the 'Add content to Review' dialog, click the option for the type of content you wish to add:

Post-commit reviews	
Browse changesets	Allows you to choose changesets from a repository. Go to Selecting changesets for review below.
Choose branches	Allows you to choose branches from a repository. Go to Choosing branches for review below.
Explore repositories	Allows you to browse for files in a repository, when FishEye is installed. Go to Selecting repository files for review below.
Search for files	Allows you to search a repository for files or changesets, when FishEye is installed. Go to Searching for files to review below.
Pre-commit	Allows you to upload patch files to a review, so you can review code changes that have not yet been committed to the SCM. Create a patch file for the code changes and then add the patch to the review. Go to Adding patch files to a pre-commit review below.
Attachments	Allows you to upload any file to a review, including binary files and files outside of a repository. Go to Adding attachments to a review below.



Add content to Review CR-222

Browse changesets
Browse recent changes for revisions to add to this review

Choose branches
Pick one or more branches to review

Explore repositories
Use a directory tree to find files

Search for files
Use Crucible's search to find revisions to add to this review

Pre-commit
Upload a patch file to be reviewed - diffs of files will be extracted

Attachments
Upload a file to be reviewed

Changesets

Click **Browse changesets** in the 'Add content to Review' dialog to add SCM changesets to your review.

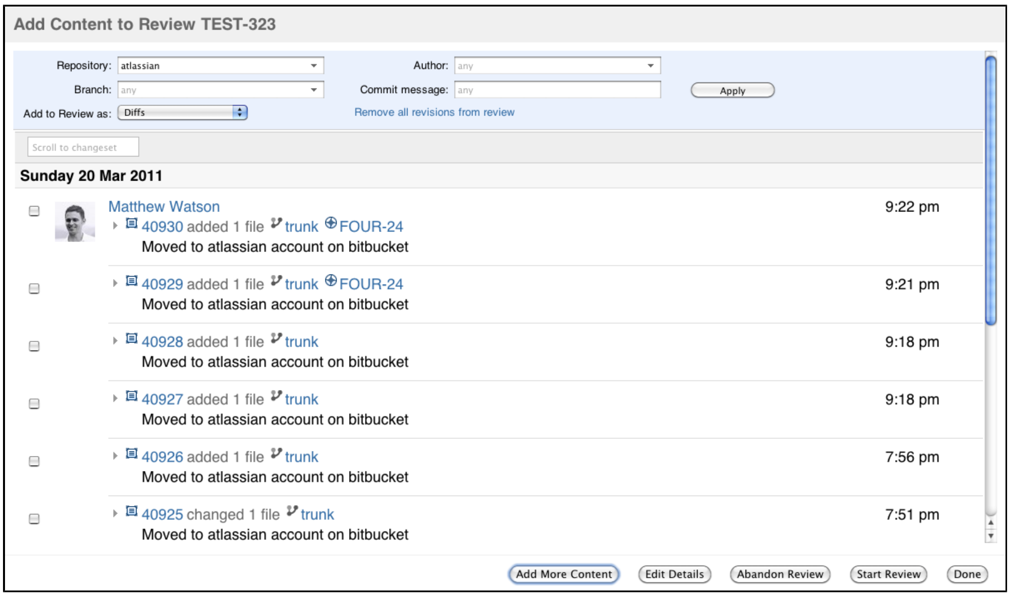
By default, Crucible displays a list of the review creator's changesets. You can see other changesets using the following options:

Repository	The repositories that contain files that can be reviewed.

Author	The authors who have made changes in the selected repository.
Branch	This shows just the recent changes on the selected branch from the selected repository.
Commit message	Filter for specific commit messages.
Scroll to changeset	Jump to a particular changeset by typing part of its changeset ID.

Select the checkbox next to a changeset ID to add the entire changeset. Note that:

- You cannot add individual file revisions to a review, although you can remove them once the changeset is added. Click **Remove all revisions from review** to remove all.
- You can choose how reviewers will see the files you have added (described below) by clicking **Add to Review as**.
- You cannot add changesets that are entirely svnprops changes (i.e. it has no non-metadata changes). For details, see How do I force reviews to include SVN property changes?



Add Content to Review TEST-323

Repository: atlassian
Author: any
Branch: any
Commit message: any
Add to Review as: Diffs

Sunday 20 Mar 2011

☐ Matthew Watson
40930 added 1 file trunk FOUR-24 — 9:22 pm
Moved to atlassian account on bitbucket

☐ 40929 added 1 file trunk FOUR-24 — 9:21 pm
Moved to atlassian account on bitbucket

☐ 40928 added 1 file trunk — 9:18 pm
Moved to atlassian account on bitbucket

☐ 40927 added 1 file trunk — 9:18 pm
Moved to atlassian account on bitbucket

☐ 40926 added 1 file trunk — 7:56 pm
Moved to atlassian account on bitbucket

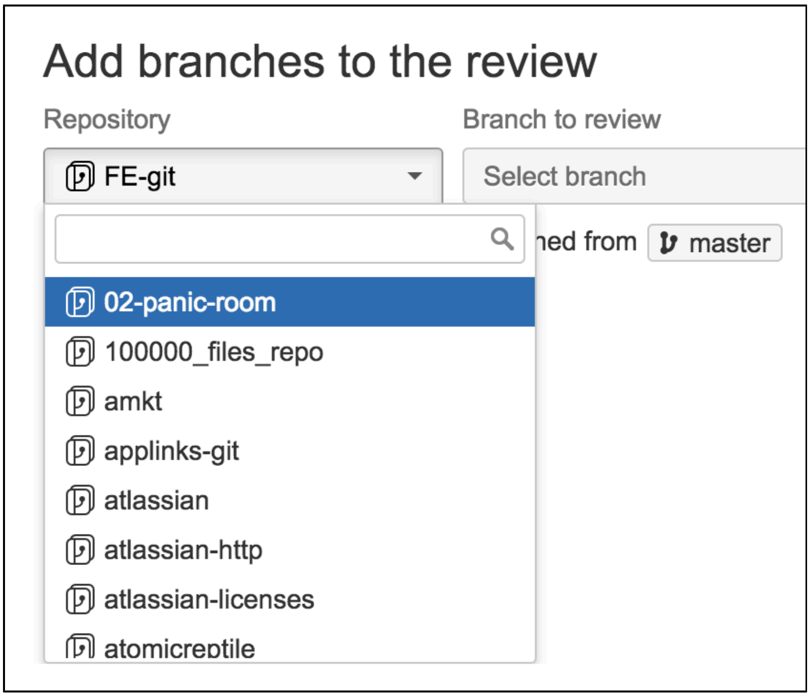
☐ 40925 changed 1 file trunk — 7:51 pm
Moved to atlassian account on bitbucket

Branches

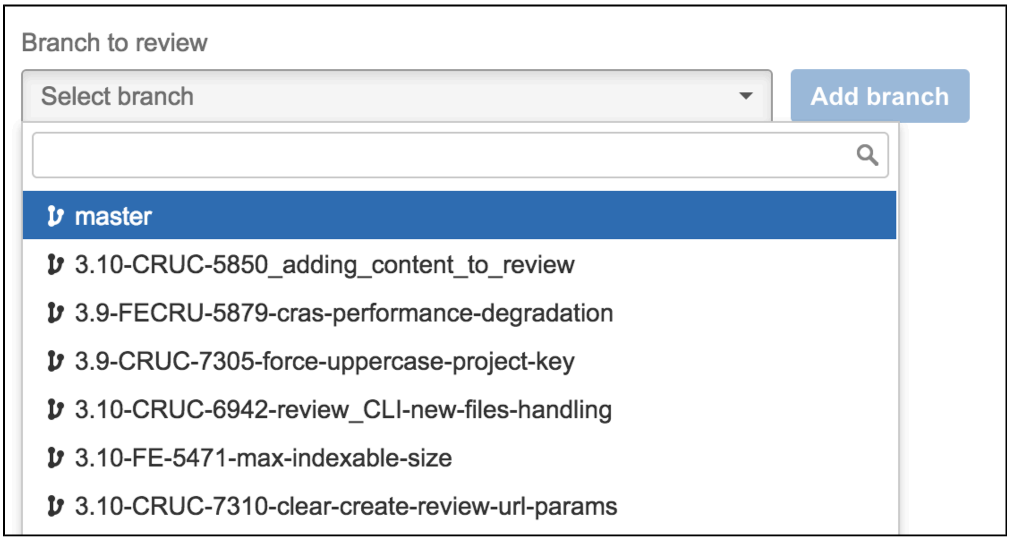
When you select a branch to be reviewed, Crucible displays a list of the changesets on the branch that have not yet been merged to the base branch – you see a quick preview of the changesets that will get added to the review.

Reviewers will only see the changes that have been made on the branch – irrelevant changes are hidden. Furthermore, your reviewers can continue reviewing even if some changes are merged from the branch.

To add a branch to a review, click **Choose branches** on the 'Add content to Review' dialog, then choose the repository that has the branch you want reviewed:



Now choose the branch:


Branch to review
Select branch
Add branch
master
3.10-CRUC-5850_adding_content_to_review
3.9-FECRU-5879-cras-performance-degradation
3.9-CRUC-7305-force-uppercase-project-key
3.10-CRUC-6942-review_CLI-new-files-handling
3.10-FE-5471-max-indexable-size
3.10-CRUC-7310-clear-create-review-url-params

You'll see a list of the recent commits. If that all looks good, click **Add branch to review** to finish.

You can also create a branch review directly from the activity stream. Once your changeset is indexed and visible in the activity stream, just click on the cog and choose **Create review for branch**, and continue as above.

Your branch review is quickly and automatically updated whenever new commits are made to the repository branch – review participants get notified and are able to resume their reviews immediately.

You can remove files from a branch review and they won't be added again on the next update, unless there were further changes to those files. Furthermore, your reviewers are able to continue with the review even if some of the changes are merged from the branch to the base branch.

Repository files

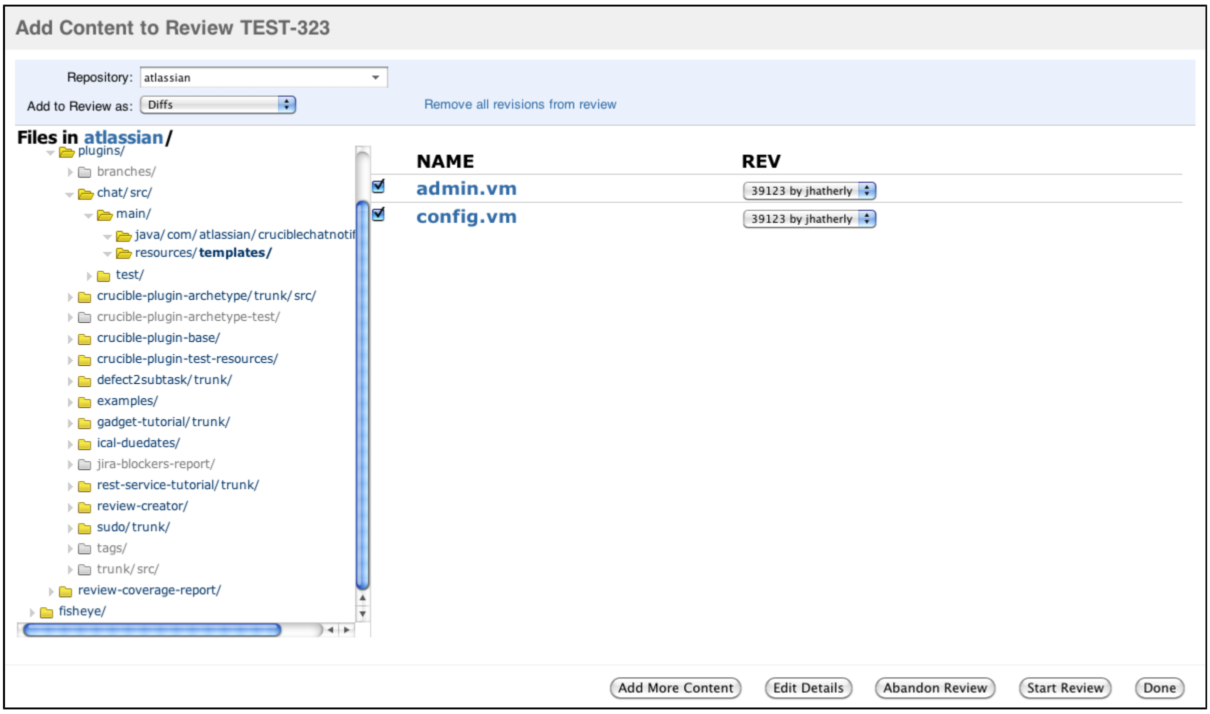
Click **Explore repositories** on the 'Add content to Review' dialog to browse the SCM repositories for files to add to your review:

- By default, the folders are sorted by path name but they can also be sorted by last-commit or first-commit.

- To select a particular revision of a file, select **Load full history...** from the revision number list. This will refresh the available options.
- You can choose how reviewers will see the files you have added, as described below.

Note that:

- Empty folders are greyed out.
- The 'Cog' menu has options to **Hide empty directories** and to **Hide deleted files**.



Add Content to Review TEST-323

Repository: atlassian

Add to Review as: Diffs

Remove all revisions from review

Files in atlassian/

- plugins/
  - branches/
  - chat/src/
    - main/
      - java/com/atlassian/cruciblechatnotif
      - resources/**templates/**
    - test/
  - crucible-plugin-archetype/trunk/src/
  - crucible-plugin-archetype-test/
  - crucible-plugin-base/
  - crucible-plugin-test-resources/
  - defect2subtask/trunk/
  - examples/
  - gadget-tutorial/trunk/
  - ical-duedates/
  - jira-blockers-report/
  - rest-service-tutorial/trunk/
  - review-creator/
  - sudo/trunk/
  - tags/
  - trunk/src/
- review-coverage-report/
- fisheye/

	NAME	REV
☑	admin.vm	39123 by jhatherly
☑	config.vm	39123 by jhatherly

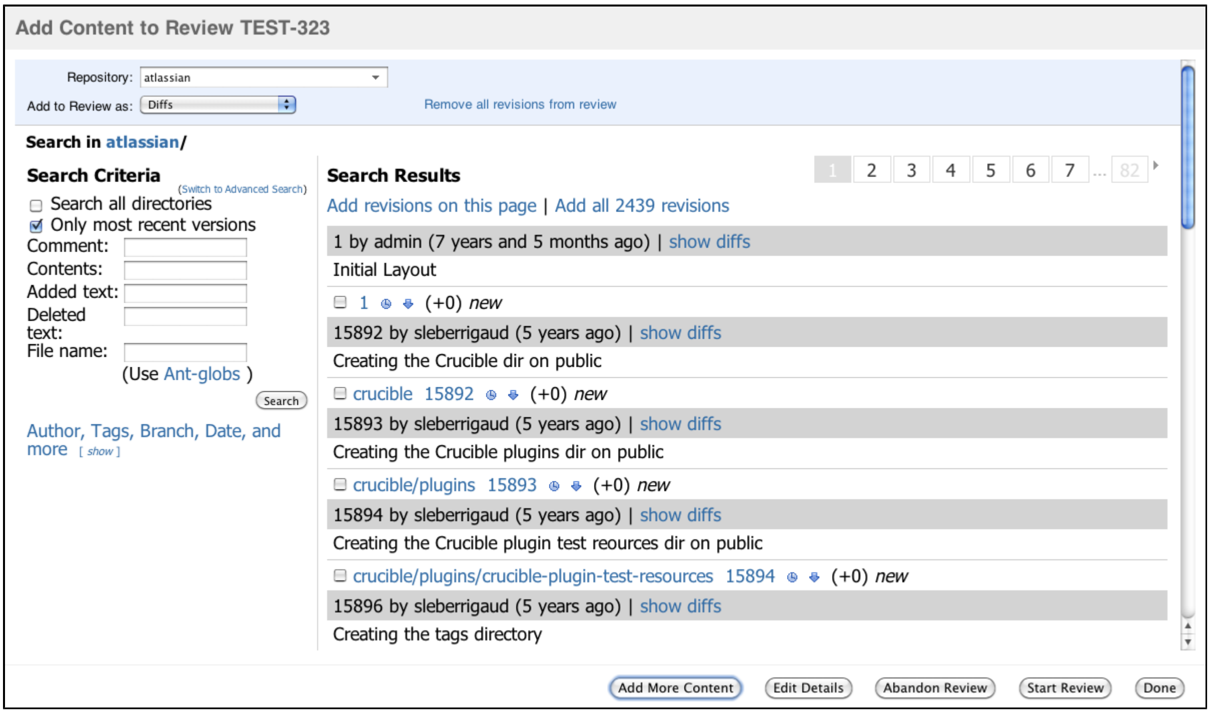
Search for files

Click **Search for files** on the 'Add content to Review' dialog to search for files to add to your review.
⚠ Search is only available when using FishEye with Crucible.

Adjust the search filters to find the files you need. If the simple filters are not enough, consider using EyeQL queries.

Read more about searching your repositories in the FishEye documentation.

You can choose how reviewers will see the files you have added, as described below.



Add Content to Review TEST-323

Repository: atlassian

Add to Review as: Diffs

Remove all revisions from review

Search in atlassian/

Search Criteria (Switch to Advanced Search)

□ Search all directories

☑ Only most recent versions

Comment:

Contents:

Added text:

Deleted text:

File name:

(Use Ant-globs)

Search

Author, Tags, Branch, Date, and more [show]

Search Results

1 2 3 4 5 6 7 ... 82

Add revisions on this page | Add all 2439 revisions

1 by admin (7 years and 5 months ago) | show diffs

Initial Layout

□ 1 (+0) *new*

15892 by sleberrigaud (5 years ago) | show diffs

Creating the Crucible dir on public

□ crucible 15892 (+0) *new*

15893 by sleberrigaud (5 years ago) | show diffs

Creating the Crucible plugins dir on public

□ crucible/plugins 15893 (+0) *new*

15894 by sleberrigaud (5 years ago) | show diffs

Creating the Crucible plugin test reources dir on public

□ crucible/plugins/crucible-plugin-test-resources 15894 (+0) *new*

15896 by sleberrigaud (5 years ago) | show diffs

Creating the tags directory

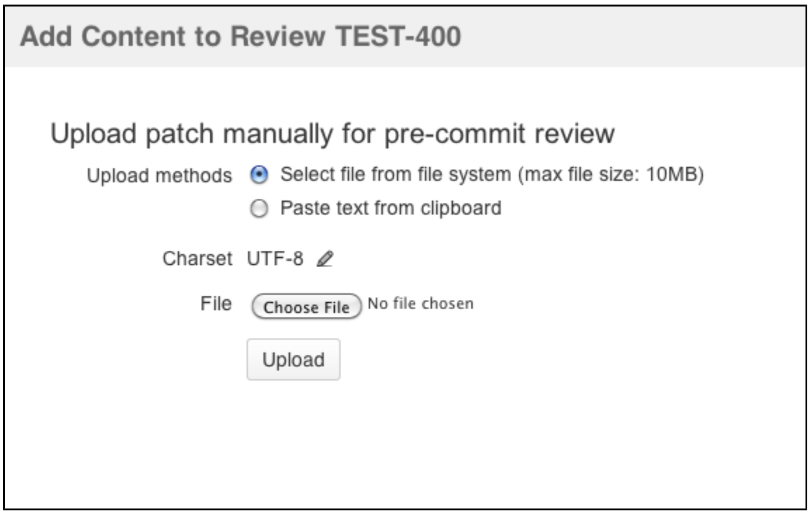
Patch files for a pre-commit review

Click **Pre-commit** on the 'Add content to Review' dialog to add previously created patch files to a pre-commit review.

Choose an upload method:

Select file from the file system	**Charset** – click the edit icon (✎) to choose the character set being used. UTF-8 is the default.
	Choose File – click to to browse for the file that you want to add to the review.
Paste text from clipboard	**Patch text** — paste your copied text in this text area.

For more information see Creating patch files for pre-commit reviews.



Add Content to Review TEST-400

Upload patch manually for pre-commit review

Upload methods ● Select file from file system (max file size: 10MB)
○ Paste text from clipboard

Charset UTF-8

File No file chosen

Patch anchoring

A short-coming of patches for code review is the reduced context around code changes because the patch does not include all lines of code from the file. Crucible 'patch anchoring' overcomes this by searching for the relevant file content in the connected repositories, and automatically anchoring the patch to the trunk or the branch with the most recent commit activity. Crucible can then seamlessly display more context, as required.

See Using Crucible patch anchoring for more information.

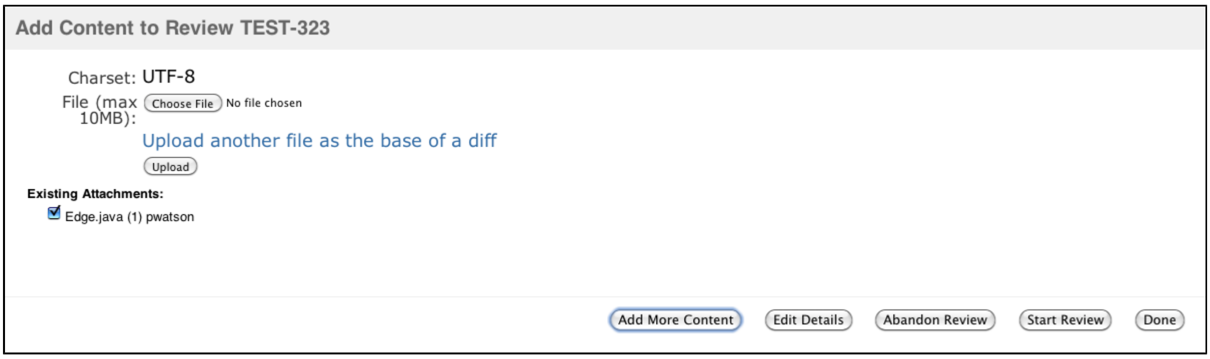
Attachments

Click **Attachments** on the 'Add content to Review' dialog to add attachments to your review.

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

Charset	Click to choose the character set being used. UTF-8 is the default.
Choose File	Browse for a file that you want to add to the review.
Upload	Browse for a file that you want to use as the base of a diff with a previously attached file.

To add another iteration of a file, upload a different version of the file with the same filename. It will be added as a new version.



Add Content to Review TEST-323

Charset: UTF-8

File (max 10MB): No file chosen

Upload another file as the base of a diff

Existing Attachments:
☑ Edge.java (1) pwatson

Choose how reviewers see the content

When you add files to a review, you can set how reviewers will see the files, for example as the whole file, or as a diff.

Choose one of the following options from the **Add to Review as** list:

Diffs	This is the default. This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
Whole Files	Adds the entire file with all content, rather than just a diff with context.
Diffs to Last Branch Point	This adds files with a diff to the revision each file was last branched.
Diffs to Last Reviewed Version	This adds files with a diff to the last reviewed changeset.
Diffs to... (a particular revision)	This allows you to specify the file to show the differences between two specific versions of a file.

Click **Done** once you have finished selecting the required files. The files will be added to your review and the review will be displayed.

Iterative reviews

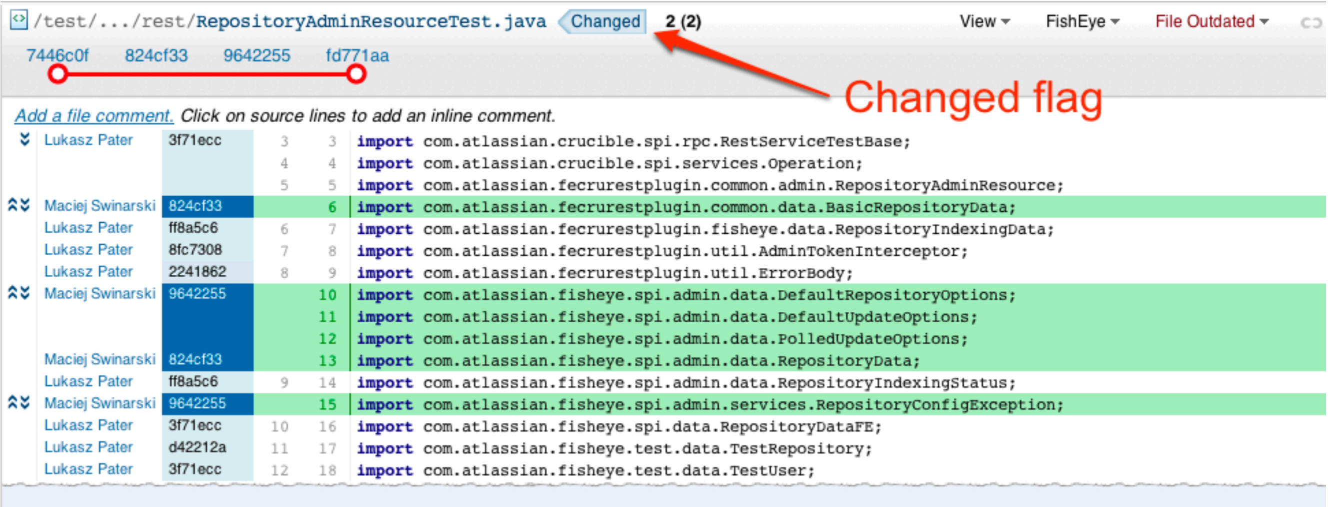
Crucible supports iterative (cumulative, or incremental) reviews for both post-commit and pre-commit reviews. This allows you to update the review with new versions of files, and changesets (for post-commit reviews) or patches (for pre-commit reviews) that have been created after the review was started.

Iterative reviews allow your team to discuss changing code in the context of a single review. This lets the reviewers see all the related changes together, and to more easily keep track of comments and defects.

On this page:

Iterative post-commit reviews

To set up an iterative post-commit review, you create a review, and add content to it, in the usual way. Crucible automatically recognizes when files under review have been updated in the repository, and provides the option to add the revision to the review.



Add a file comment. Click on source lines to add an inline comment.

```
Lukasz Pater       3f71ecc   import com.atlassian.crucible.spi.rpc.RestServiceTestBase;
                             import com.atlassian.crucible.spi.services.Operation;
                             import com.atlassian.fecrurestplugin.common.admin.RepositoryAdminResource;
Maciej Swinarski   824cf33   import com.atlassian.fecrurestplugin.common.data.BasicRepositoryData;
Lukasz Pater       ff8a5c6   import com.atlassian.fecrurestplugin.fisheye.data.RepositoryIndexingData;
Lukasz Pater       8fc7308   import com.atlassian.fecrurestplugin.util.AdminTokenInterceptor;
Lukasz Pater       2241862   import com.atlassian.fecrurestplugin.util.ErrorBody;
Maciej Swinarski   9642255   import com.atlassian.fisheye.spi.admin.data.DefaultRepositoryOptions;
                             import com.atlassian.fisheye.spi.admin.data.DefaultUpdateOptions;
                             import com.atlassian.fisheye.spi.admin.data.PolledUpdateOptions;
Maciej Swinarski   824cf33   import com.atlassian.fisheye.spi.admin.data.RepositoryData;
Lukasz Pater       ff8a5c6   import com.atlassian.fisheye.spi.admin.data.RepositoryIndexingStatus;
Maciej Swinarski   9642255   import com.atlassian.fisheye.spi.admin.services.RepositoryConfigException;
Lukasz Pater       3f71ecc   import com.atlassian.fisheye.spi.data.RepositoryDataFE;
Lukasz Pater       d42212a   import com.atlassian.fisheye.test.data.TestRepository;
Lukasz Pater       3f71ecc   import com.atlassian.fisheye.test.data.TestUser;
```

See Viewing diffs below for information about the slider and diffs.

Iterative pre-commit reviews

Pre-commit reviews make use of patch files that are uploaded to a review. Crucible allows revisions of patch files to be uploaded to the review, and can display diffs for files in the patches. This allows your team to set up and perform iterative pre-commit reviews.

See Creating patch files for pre-commit reviews.

Initial patch upload

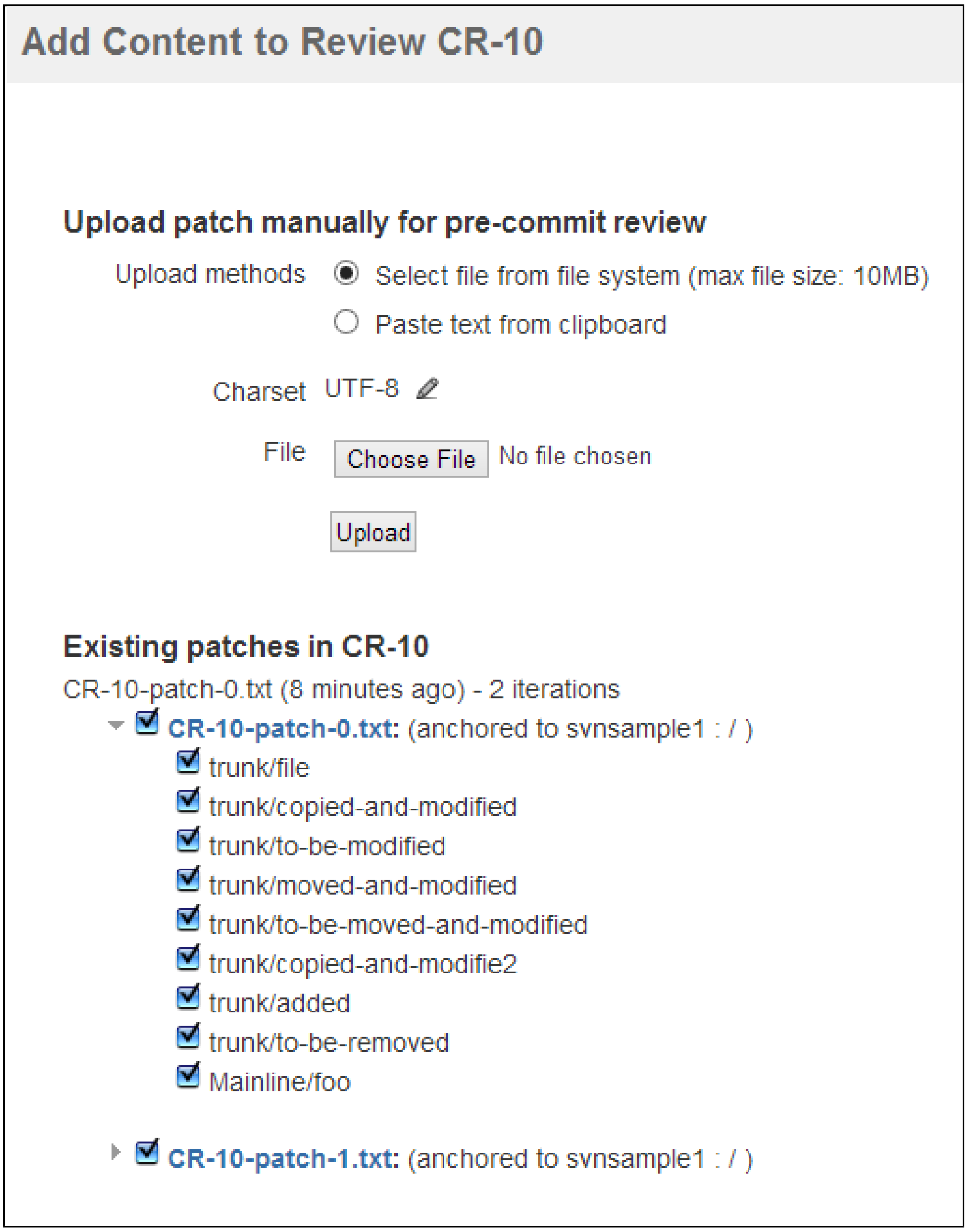
When uploading the initial patch for a review, Crucible must be able to anchor the patch to a repository if you subsequently want to upload patch iterations. If Crucible is unable to anchor the patch to a repository, you will only be able to upload the patches as separate files.

You upload the initial patch for a review in the usual way – see Adding content to the review.

Iterative patch uploads

When you add a new iteration of the patch to the review, you can choose which previously uploaded patch it is a revision of. The new patch must be anchored to the same repository as an existing patch.

Note that you cannot add unanchored patches, even if they include full-context diffs. You can include an unanchored *file* in the anchored patch, however Crucible is unable to provide full context for that.



Add Content to Review CR-10

Upload patch manually for pre-commit review

Upload methods ● Select file from file system (max file size: 10MB)
○ Paste text from clipboard

Charset UTF-8

File Choose File No file chosen

Upload

Existing patches in CR-10

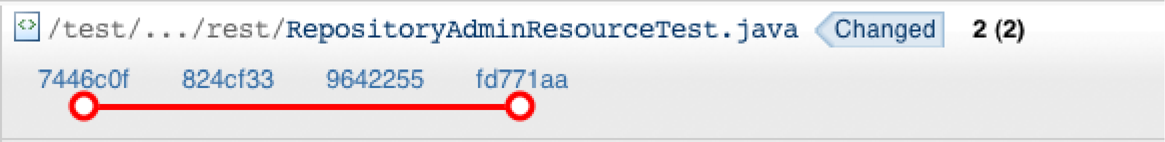
CR-10-patch-0.txt (8 minutes ago) - 2 iterations

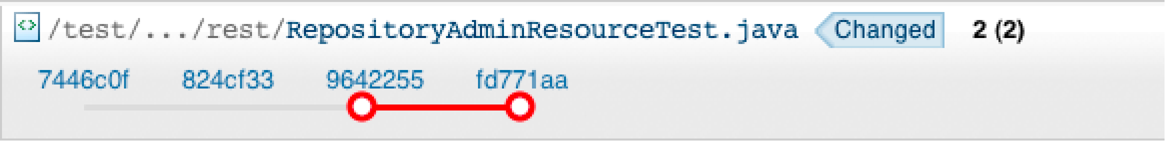
- ☑ **CR-10-patch-0.txt:** (anchored to svnsample1 : /)
  - ☑ trunk/file
  - ☑ trunk/copied-and-modified
  - ☑ trunk/to-be-modified
  - ☑ trunk/moved-and-modified
  - ☑ trunk/to-be-moved-and-modified
  - ☑ trunk/copied-and-modifie2
  - ☑ trunk/added
  - ☑ trunk/to-be-removed
  - ☑ Mainline/foo
- ☑ **CR-10-patch-1.txt:** (anchored to svnsample1 : /)

Viewing diffs

Crucible allows the reviewer to see the different revisions of a file within the same review. The 'slider' in the file view allows you to interactively select which revisions are compared in the displayed diff, and to see the full source content. Comments are connected to, and displayed with, a specific revision. This allows you to review every change that has occurred on a code file over a given range of commits, and lets you see the evolution of the file through various revisions (all within one Crucible review).

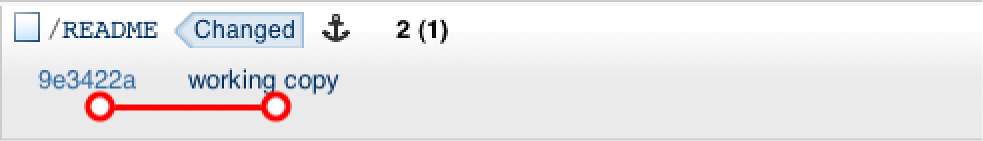
These screenshots show how, for a post-commit review, you can drag the slider 'handles' so as to display just the changes in the last commit:





Drag the 'handles' to the same commit to see the full source of that version of the file.

When viewing patch files in a pre-commit review, the slider displays the diff for the selected iterations, in a similar way to that for post-commit reviews. Each patch iteration is referred to as a 'working copy'.



Creating patch files for pre-commit reviews

This page describes how to create patch files from your local repository, how to attach them to a Crucible review and how to use Crucible's Patch Anchoring to retrieve more context from the original file.

A patch file is a portion of a source code file that contains the code changes that you have made – it's a diff that shows the differences between your working copy and the base revision.

A pre-commit review in Crucible allows a developer's code changes, in the form of a patch file, to be reviewed before those changes are committed to the SCM. A typical scenario is where the developer does not have write access to the repository.The developer creates the patch file and adds it to a Crucible review. Once reviewed, the patch is either committed to the repository or is sent back to the author.

You can create the patch file from your local repository:

- using tools in your IDE – described below
- using repository command-line tools
- using the Crucible Review CLI tool – see Creating reviews from the command line

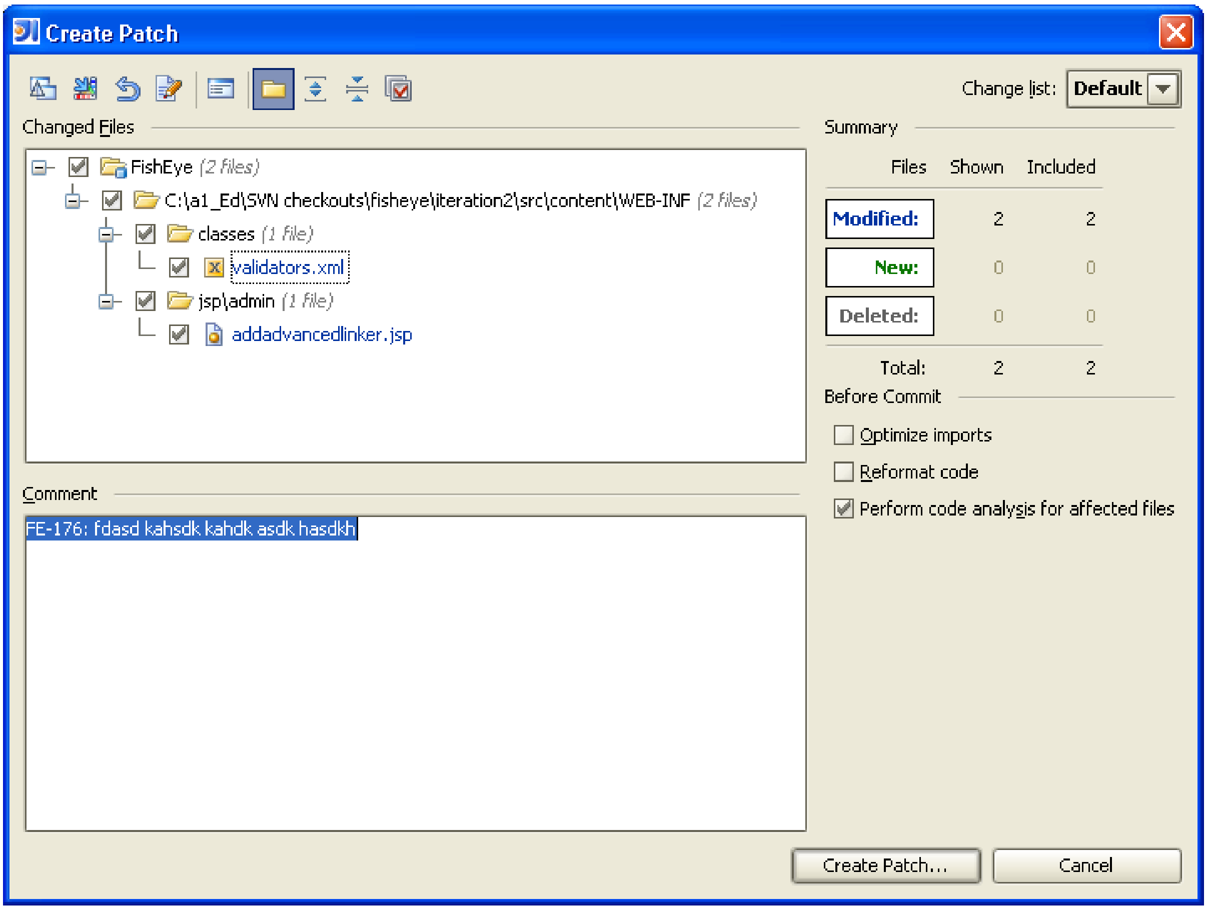
As an enhancement, Crucible's patch anchoring adds context:

- By default, patch files will only show a few lines of code surrounding each change, rather than the entire file and its changes. Patch anchoring overcomes this limitation.

On this page:

Creating a patch file from IntelliJ IDEA 7.0

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



Change list: Default

Changed Files

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

Comment

FE-176: fdasd kahsdk kahdk asdk hasdkh

Summary

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

Before Commit

☐ Optimize imports

☐ Reformat code

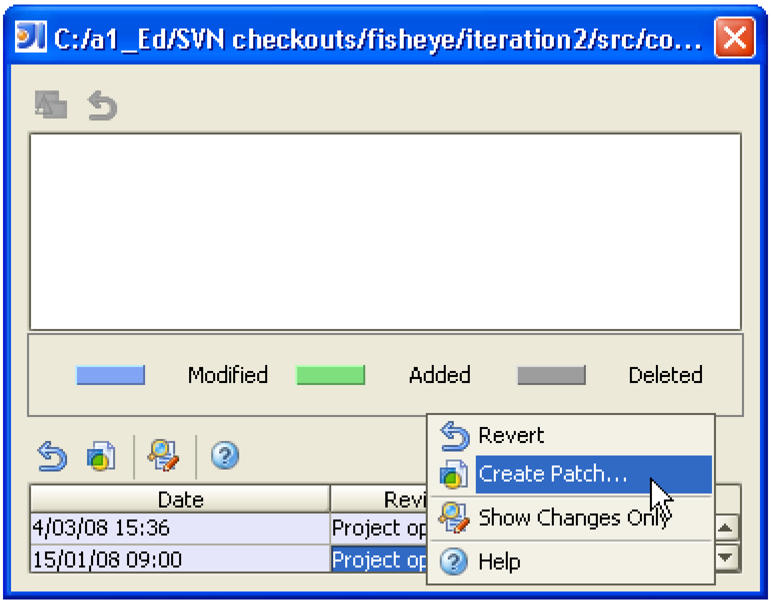
☑ Perform code analysis for affected files

3. Click **Create Patch**, choose a location to save the patch file to, and click **OK**.

If you do not have the Create Patch command available in IDEA

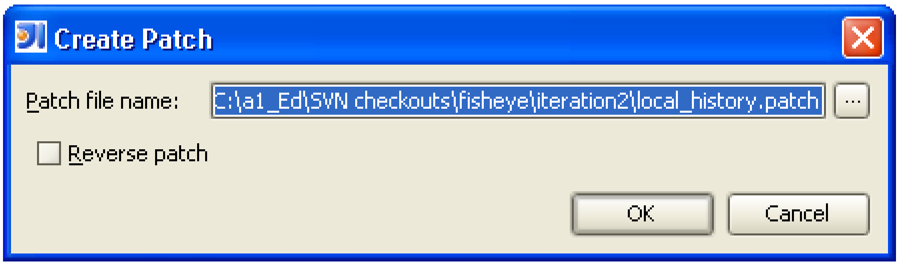
If you have not configured version control in IDEA, you may not have the **Create Patch** option available. If so, use the following steps to create a patch file in IDEA:

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



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

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



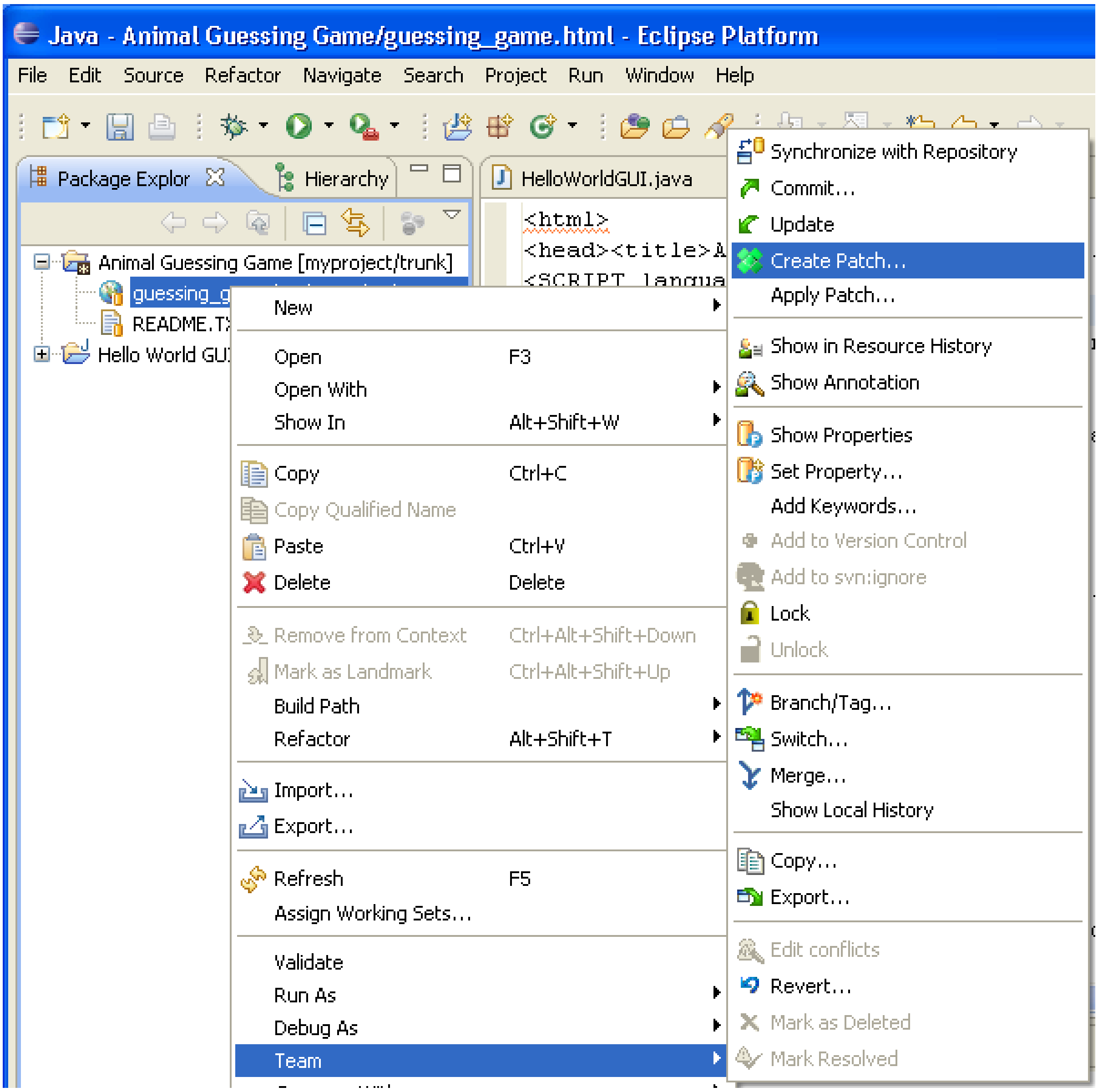
Patch file name: C:\a1_Ed\SVN checkouts\fisheye\iteration2\local_history.patch

☐ Reverse patch

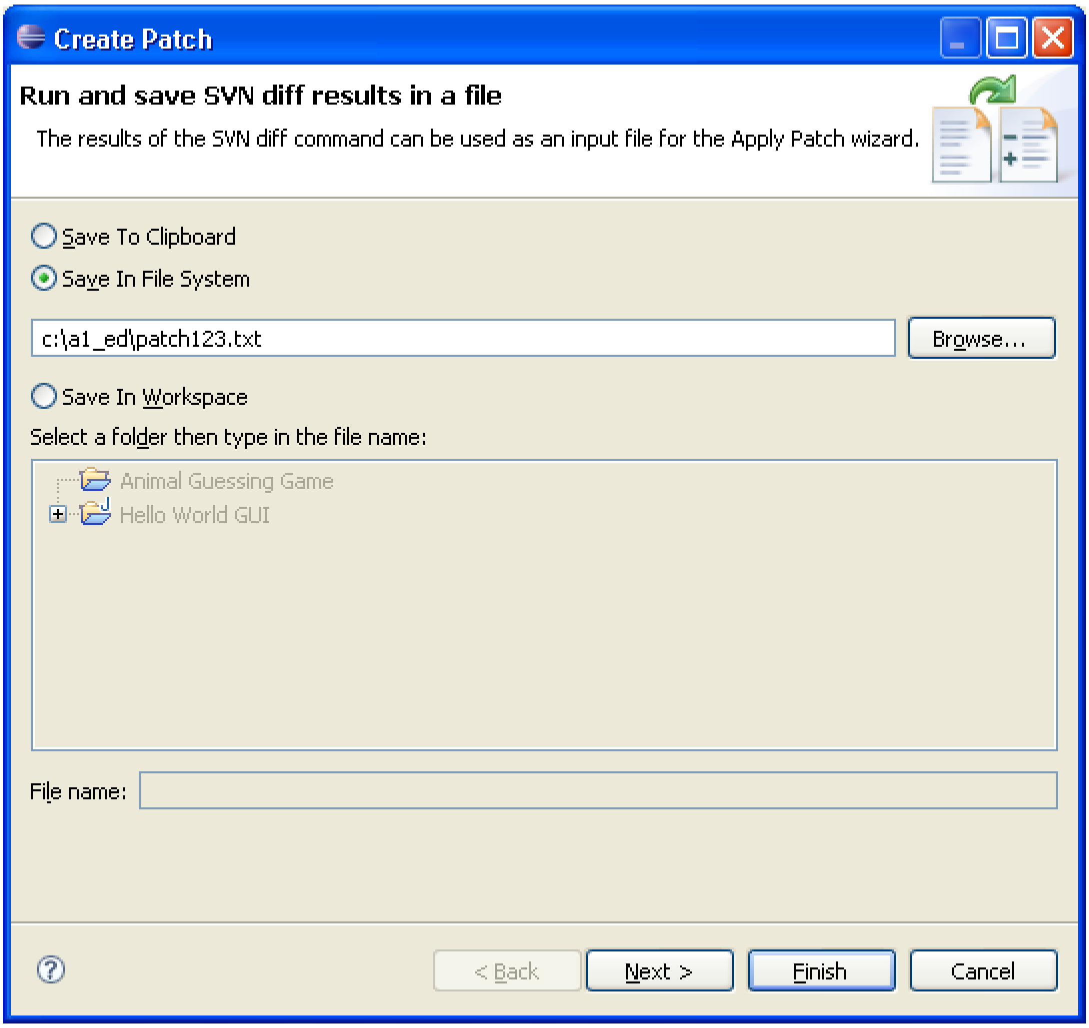
3. In the Create Patch dialog, choose a location for the patch file and a file name, then click **OK**.

Creating a patch file in Eclipse 3.3.1.1

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



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



Create Patch

Run and save SVN diff results in a file

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

○ Save To Clipboard

● Save In File System

c:\a1_ed\patch123.txt

○ Save In Workspace

Select a folder then type in the file name:

Animal Guessing Game

Hello World GUI

File name:

Creating a patch file from the command line

Create patches yourself, directly from the SCM, using the following commands, where `patch.txt` represents your name for the new patch file.

Then you can use the dedicated Crucible Review CLI tool to create reviews for your patches and commits, directly from your terminal. See Creating reviews from the command line.

SCM	Command	Notes
CVS	`cvs diff -Nu > patch.txt`	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context.
	`cvs diff -N -U 10000 > patch.txt`	Creates a patch file that shows all code in the file. `10000` refers to the number of code lines before and after each change that are included in the patch.
SVN	`svn diff > patch.txt`	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context. ⚠ `svn diff` does not print any information about files copied in the workspace.

	`svn diff --diff-cmd diff -x "-U 10000" > patch.txt`	Creates a patch file that shows all code in the file. • The built-in 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 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.
Perforce	Crucible 3.3.1 or earlier: `p4 diff -dcu > patch.txt` Crucible 3.3.2 or later: `p4 -z tag diff -du > patch.txt`	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context. The `-dcu` option provides a combination of "context format" and "unified format". It provides the diff in a standard unified diff format (which we need to parse the diff) as well as revision information (which we need to anchor to FishEye). Later versions of Perforce do not support `-dcu`, so `-z tag` should be used instead.
	`diff -u /dev/null path_to_added_file >> patch.txt`	Example of using GNU `diff` to append files individually to the patch in UNIX. Since Perforce diffs do not include added and deleted files, you should use `p4 opened` to find such files. 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.
	⚠ Perforce does not directly support creating patches that include all lines of code. A workaround is to check out '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.	
Mercurial	`hg diff > patch.txt`	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context. If you use Git-style diffs (--git), the revision information will not be provided. This means that we cannot anchor the patch to a FishEye repository.
Git	`git diff > patch.txt`	Creates a patch file with around three lines of code, before and after each change. Revision information is included by default, so if you have FishEye, you may be able to anchor the patch to get full context.

Using Crucible patch anchoring to automatically add full context

Crucible's Patch Anchoring feature allows you to add a regular patch (showing only a few lines of context) to a review. Then, Crucible will automatically search for the relevant file content in the connected repositories. When it finds the files, it will seamlessly add in more context from the files so that you can view all of the lines of code (greatly enhancing the review process).

To use patch anchoring:

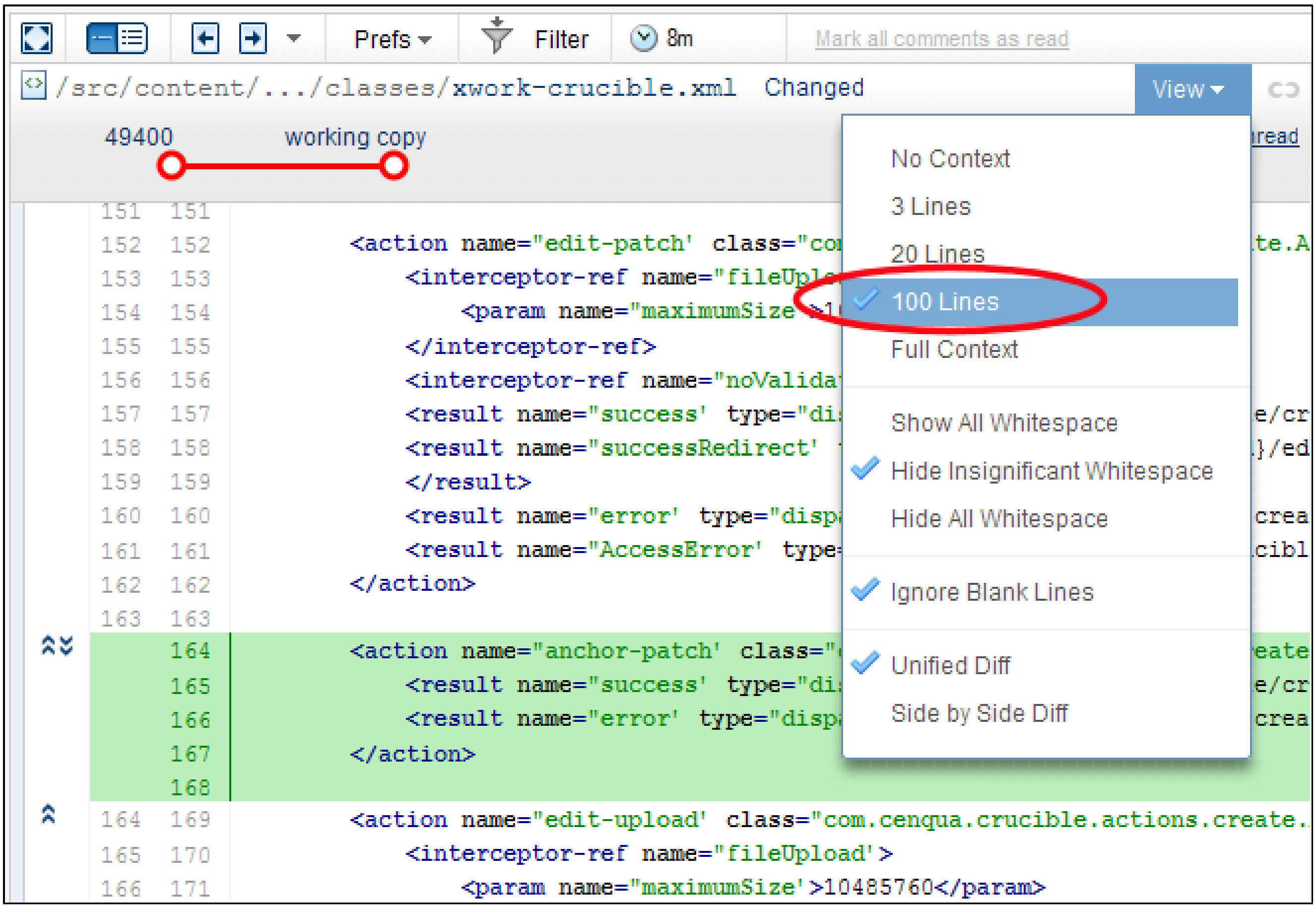
1. Click **Create review** in the Crucible header.

2. Click **Pre-commit**. In the dialog that appears, click **Choose File** to locate your file, then **Upload**. Crucible will now search for matches in the files in its database. Crucible will analyze all the paths in the patch, find the branches containing all those paths, then anchor the patch to the trunk or the branch with the most recent commit activity.

Crucible makes a 'best guess' in its processing – you should check that it has anchored the patch to the correct location in your repository.

When you start the review, and view a diff, you will be able to choose more than three lines of context from the **V iew** menu.

Screenshot: Viewing more than three lines with Patch Anchoring



/src/content/.../classes/xwork-crucible.xml Changed

49400 working copy

```
        <action name="edit-patch' class="co
            <interceptor-ref name="fileUpl
                <param name="maximumSize' >1
            </interceptor-ref>
            <interceptor-ref name="noValida
            <result name="success' type="di
            <result name="successRedirect'
            </result>
            <result name="error' type="disp
            <result name="AccessError' type
        </action>

        <action name="anchor-patch' class="
            <result name="success' type="di
            <result name="error' type="disp
        </action>

        <action name="edit-upload' class="com.cenqua.crucible.actions.create.
            <interceptor-ref name="fileUpload'>
                <param name="maximumSize'>10485760</param>
```

Choosing reviewers

This page describes how to add reviewers (as a mix of individual users and groups) to a new review, after it has been created. See Creating a review for information about creating reviews.

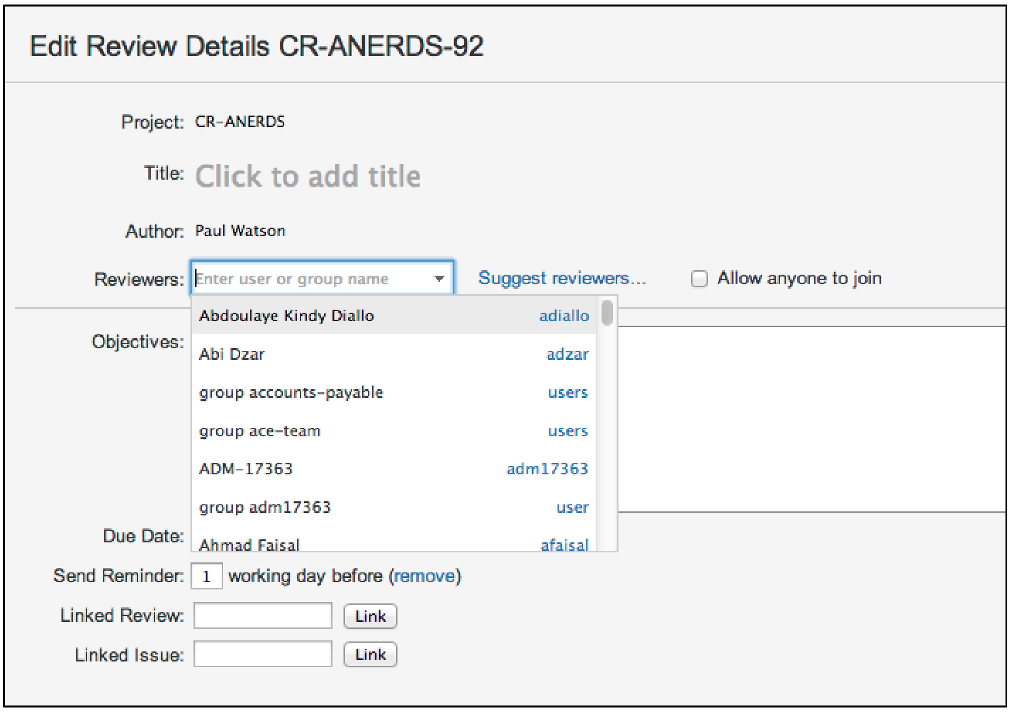
Before a review can be started, you need to choose reviewers. To add reviewers you just click **Edit Details** for the review.

Adding users to a review

You can choose reviewers from available Crucible users and groups by typing names into the **Reviewers** field. The users and groups that are available to be reviewers are determined by the project's settings for **Default Reviewers** – see Creating a project.

When adding a group to a review, only the first 9 users in the group are added – extra users must be added individually. Note that a Crucible administrator can set the number of users added from a group by using the `crucible.users.per.group.in.review` property – see JVM system properties.

You can also allow any Crucible user to add themselves as a reviewer by selecting **Allow anyone to join**.



Edit Review Details CR-ANERDS-92

Project: CR-ANERDS

Title: Click to add title

Author: Paul Watson

Reviewers: Enter user or group name — Suggest reviewers... □ Allow anyone to join

Abdoulaye Kindy Diallo	adiallo
Abi Dzar	adzar
group accounts-payable	users
group ace-team	users
ADM-17363	adm17363
group adm17363	user
Ahmad Faisal	afaisal

Objectives:

Due Date:

Send Reminder: 1 working day before (remove)

Linked Review: Link

Linked Issue: Link

Click **Done** to save the review as a draft for later issue.

Click **Start Review** to begin the review immediately.

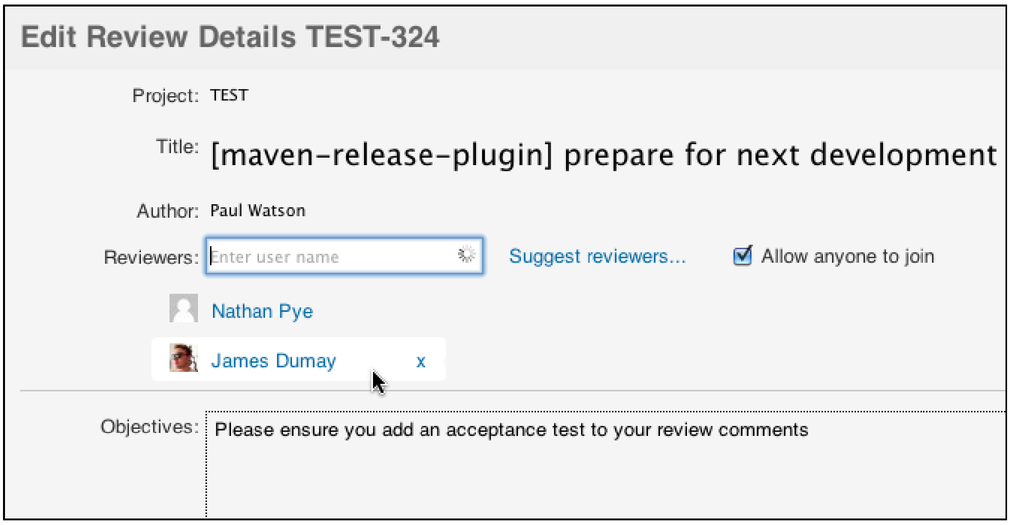
Suggested reviewers

Crucible will automatically suggest reviewers, by analyzing the users that have contributed to the files you've selected and also don't have a lot of open reviews. You can easily pick reviewers from the list of suggestions by clicking.

Removing reviewers

You can remove reviewers who no longer need to contribute to the review.

Click **Edit Details** for the review, hover over the reviewer's name, and click the cross.



Edit Review Details TEST-324

Project: TEST

Title: [maven-release-plugin] prepare for next development

Author: Paul Watson

Reviewers: Enter user name — Suggest reviewers... ☑ Allow anyone to join

Nathan Pye

James Dumay x

Objectives: Please ensure you add an acceptance test to your review comments

Checking the draft and starting the review

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

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

Next steps

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

Performing the review

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

On this page:

- Browse your reviews on the Dashboard
- When files change during a review
- Next steps

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

Browse your reviews on the Dashboard

When you first start Crucible, the Dashboard shows your current reviews and other activity related to you.

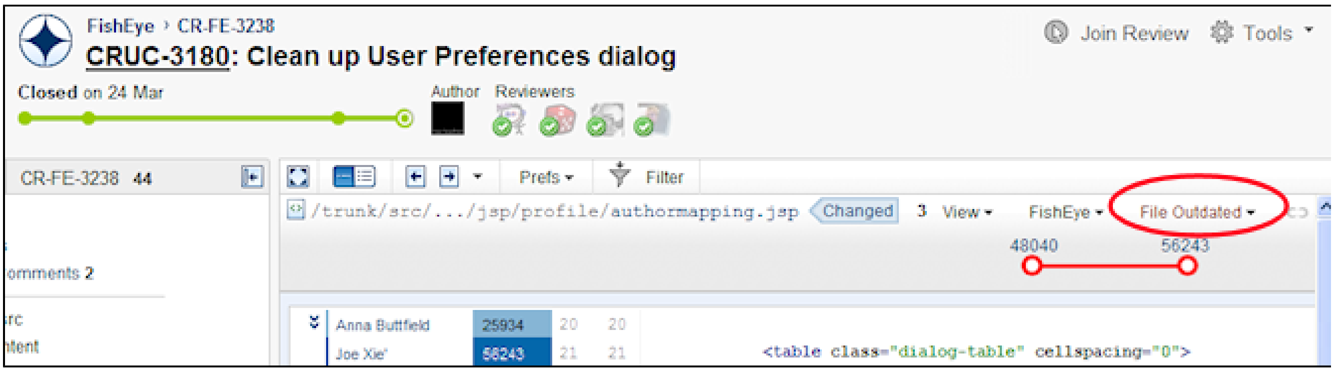
Use the Dashboard to manage your reviews – see Browsing all reviews.

All reviews that involve you in any role are listed when you click **Inbox** or **Outbox** under the **Reviews** menu. For example, choose **Reviews** > **Inbox** to see reviews that don't require further action from you, but are still active.

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

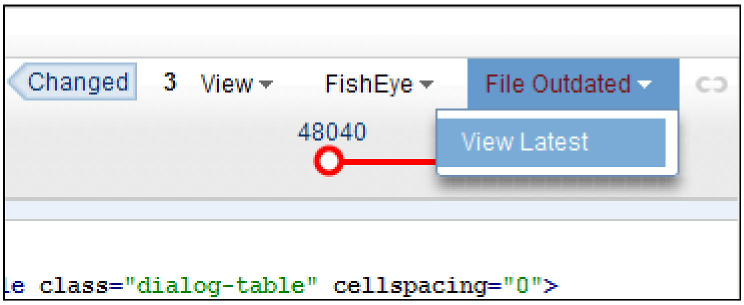
When files change during a review

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



```
<table class="dialog-table" cellspacing="0">
```

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



```
le class="dialog-table" cellspacing="0">
```

Next steps

- Starting a review
- Commenting on reviews
- Sending a review's comments via email
- Changeset discussions
- Flagging defects
- Viewing reports
- Completing your review
- Using the Review History dialog

Starting a review

On this page:

- Starting a review
- Editing a review once started

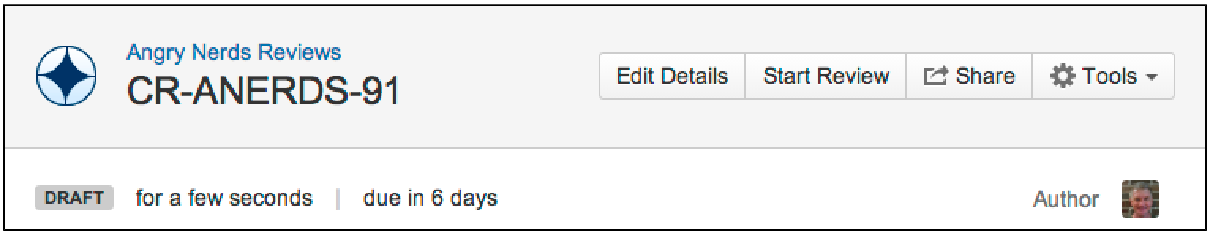
Starting a review

Starting a review simply means formally starting it – this makes the review available for the selected reviewers to begin reviewing.

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 start a review.

To start the review:

- If you are the moderator of the review, click **Start Review**. Note that only people with the 'Approve' permission can start a review.
- If you are not the moderator of your review, click **Send to Moderator**. This changes the state to 'Requires Approval' and notifies the moderator. The moderator can change any aspect of the review before starting it.



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

Editing a review once started

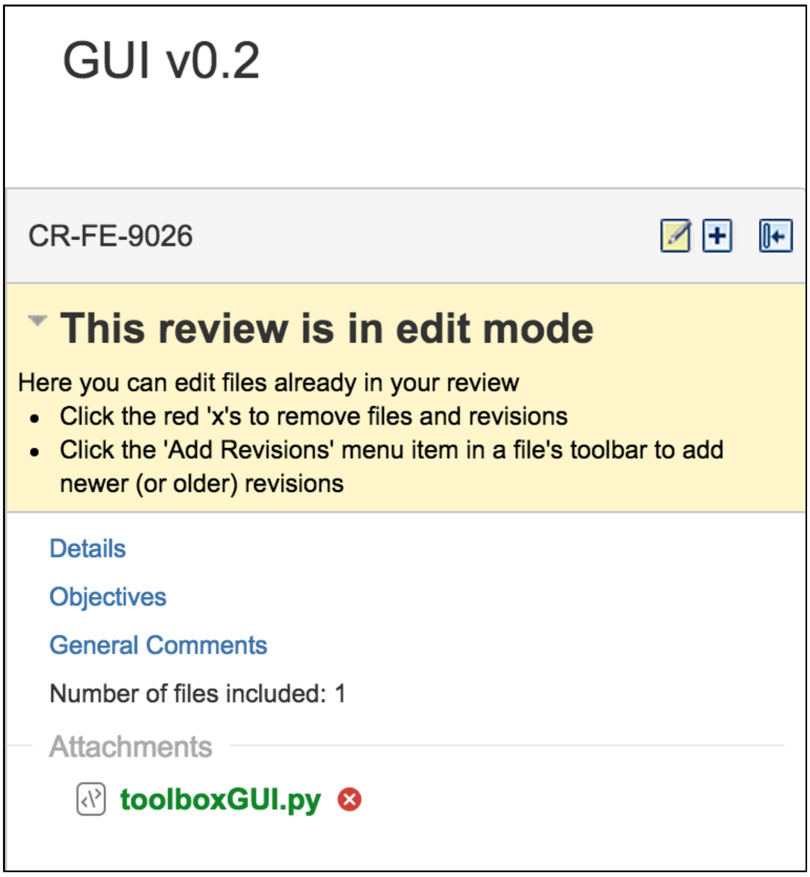
You can quickly add files to, or remove files from, the content of a review at any time.

To add files to the review:

- Click **Edit Details** near the top right of the review, and then**Add content** to add files.

To remove files from the review:

- click the 'Edit' button at the top of the left navigation panel (highlighted in yellow in the screenshot below), then clicking the red cross beside the files you want to remove.



GUI v0.2

CR-FE-9026

This review is in edit mode

Here you can edit files already in your review

- Click the red 'x's to remove files and revisions
- Click the 'Add Revisions' menu item in a file's toolbar to add newer (or older) revisions

Details

Objectives

General Comments

Number of files included: 1

Attachments

toolboxGUI.py

Commenting on reviews

Comments are at the core of the review experience – reviewers use comments to record discussion around suggested and required changes to the code.

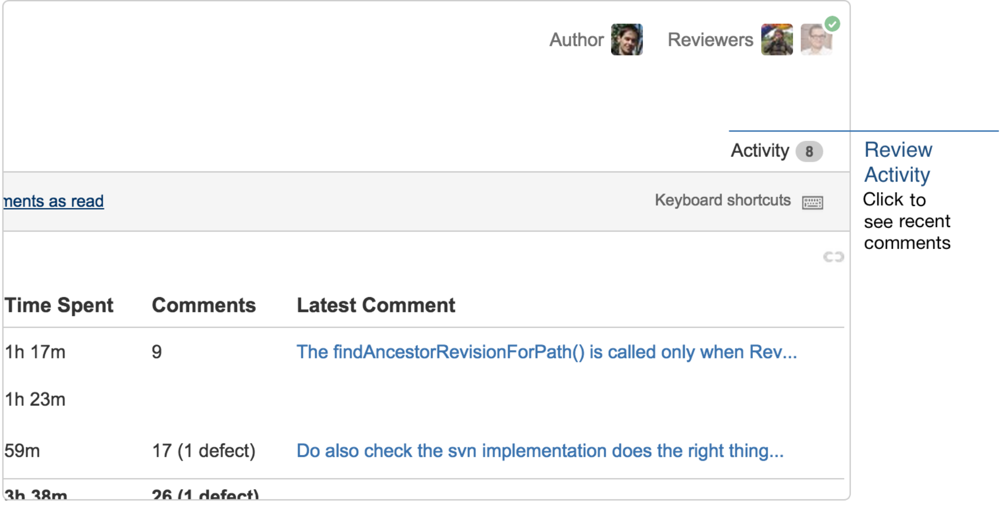
Comments can be added at the level of a review, a file, or a line. Of course, you can also reply to comments.

On this page:

Locating existing comments

Use the Review Activity

The easist way to browse for comments is to use the Review Activity – click **Activity** when viewing the review (or use the keyboard shortcut 'shift' + ']'):



Author Reviewers

Activity 8

nents as read

Keyboard shortcuts

Time Spent	Comments	Latest Comment
1h 17m	9	The findAncestorRevisionForPath() is called only when Rev...
1h 23m		
59m	17 (1 defect)	Do also check the svn implementation does the right thing...
3h 38m	26 (1 defect)	

Review Activity
Click to see recent comments

The Review Activity shows all the latest comments on the review, so it's easy to catch up on the review, or to find outdated or hidden comments.

The bubble beside the **Activity** link shows the number of unread comments.

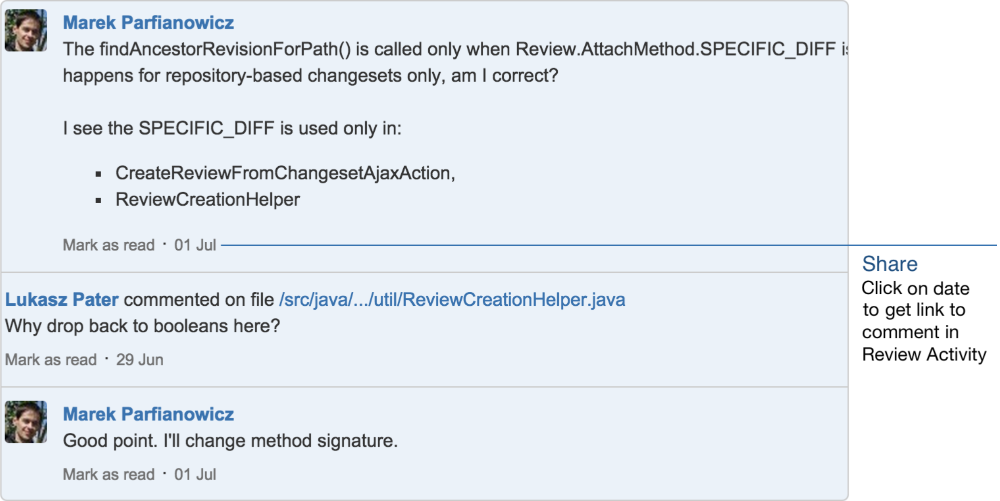
Scroll through the comments or use the keyboard shortcuts:

- 'n' – go to the next comment
- 'p' – go to the previous comment
- 'shift' + 'n' – go to the next unread comment
- 'shift' + 'p' – go to the previous unread comment

Click **Mark as read** or **Mark as unread** (or use the 'm' keyboard shortcut) to toggle the comment status.

To see the comment in context just click it (or press the 'return' key) to go to the file where the comment was made.

Use the date link for a comment to connect others to the comment:



Marek Parfianowicz
The findAncestorRevisionForPath() is called only when Review.AttachMethod.SPECIFIC_DIFF i
happens for repository-based changesets only, am I correct?

I see the SPECIFIC_DIFF is used only in:

- CreateReviewFromChangesetAjaxAction,
- ReviewCreationHelper

Mark as read · 01 Jul

Lukasz Pater commented on file /src/java/.../util/ReviewCreationHelper.java
Why drop back to booleans here?
Mark as read · 29 Jun

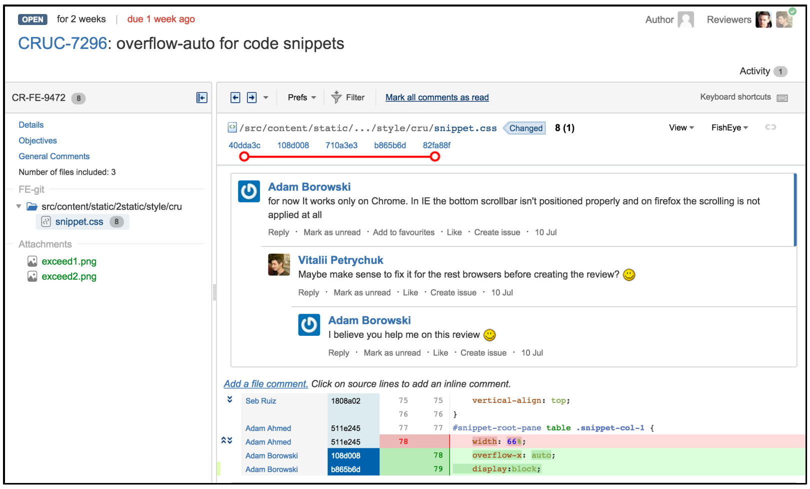
Marek Parfianowicz
Good point. I'll change method signature.
Mark as read · 01 Jul

Share
Click on date to get link to comment in Review Activity

Use the file tree

The number beside a filename, in the left-hand panel of a review, indicates the number of comments on that file.

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



CRUC-7296: overflow-auto for code snippets

Number of files included: 3

/src/content/static/.../style/cru/snippet.css Changed 8 (1)

Adam Borowski
for now It works only on Chrome. In IE the bottom scrollbar isn't positioned properly and on firefox the scrolling is not applied at all
Reply · Mark as unread · Add to favourites · Like · Create issue · 10 Jul

Vitalii Petrychuk
Maybe make sense to fix it for the rest browsers before creating the review? 🙂
Reply · Mark as unread · Like · Create issue · 10 Jul

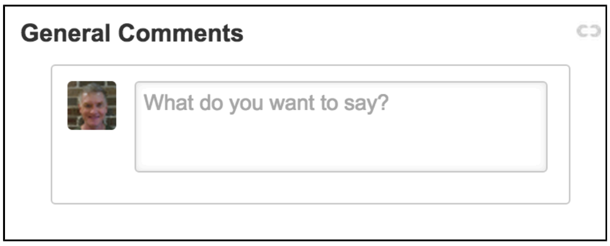
Adam Borowski
I believe you help me on this review 🙂
Reply · Mark as unread · Like · Create issue · 10 Jul

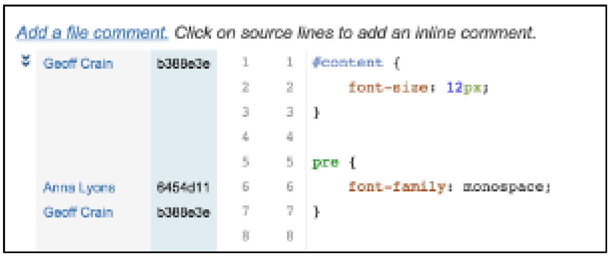
Add a file comment. Click on source lines to add an inline comment.

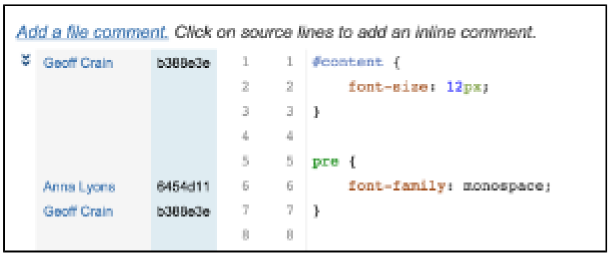
```
Seb Ruiz         1808a02   75  75     vertical-align: top;
                           76  76  }
Adam Ahmed       511e245   77  77  #snippet-root-pane table .snippet-col-1 {
Adam Ahmed       511e245   78         width: 66%;
Adam Borowski    108d008       78     overflow-x: auto;
Adam Borowski    b865b6d       79     display:block;
```

Adding comments

There are various types of comments that you can add in Crucible:

To comment on...	Do this...	
The whole review	Click **General Comments** (in the left-hand panel), and simply type your comment (under 'General Comments').	 General Comments What do you want to say?

A source file under review	Click on the file in the left-hand panel, then click **Add a file comment** (just above the source code listing).	
		Add a file comment. Click on source lines to add an inline comment.
		Geoff Crain b388e3e 1 1 #content {
		2 2 font-size: 12px;
		3 3 }
		4 4
		5 5 pre {
		Anna Lyons 6454d11 6 6 font-family: monospace;
		Geoff Crain b388e3e 7 7 }
		8 8

Lines of code	Click on a line of code in the displayed source file of a review.
	- You can click and drag to select multiple lines, and click individual lines to select or deselect them. The comment will appear in the source at the last line selected.
	- Hover over the comment to see the lines to which the comment applies.
	- To select text on the page without adding a comment, hold down the Alt key while dragging the cursor.
		
		Add a file comment. Click on source lines to add an inline comment.
		Geoff Crain b388e3e 1 1 #content {
		2 2 font-size: 12px;
		3 3 }
		4 4
		5 5 pre {
		Anna Lyons 6454d11 6 6 font-family: monospace;
		Geoff Crain b388e3e 7 7 }
		8 8

A revision or changeset	See Changeset discussions.	

To reply to a comment, click **Reply** at the bottom of the comment.

⚠ Only people with the 'Comment' permission can add comments. A comment can only be deleted by the author of the comment.

Read about flagging defects too.

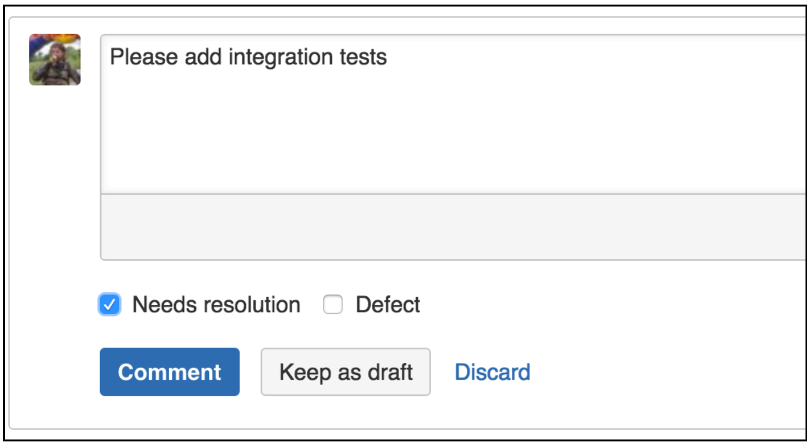
Draft comments

You can save your comment as a draft and then edit it later. When you complete the review, you will be prompted to post, discard or edit any remaining draft comments.

Resolving comments

You can use comments to track tasks that still need to be done for a review. Simply mark the comment as **needs resolution**.

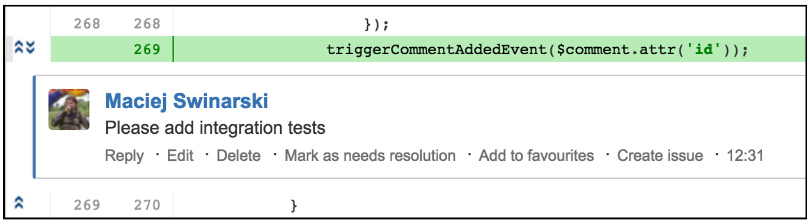
You can do this when adding a new comment:



Please add integration tests

☑ Needs resolution □ Defect

Or you can mark any existing comment:



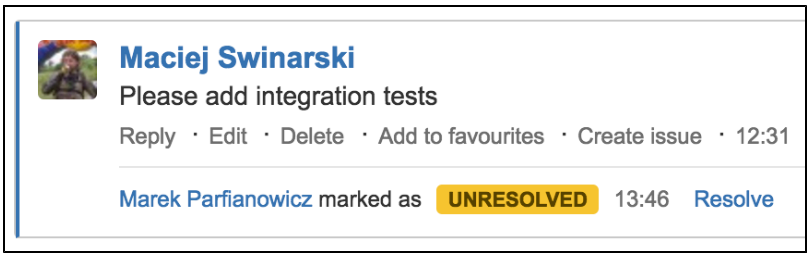
```
268  268                });
     269           triggerCommentAddedEvent($comment.attr('id'));
```

Maciej Swinarski
Please add integration tests

Reply · Edit · Delete · Mark as needs resolution · Add to favourites · Create issue · 12:31

```
269  270           }
```

Once a comment is marked, it is flagged as UNRESOLVED:

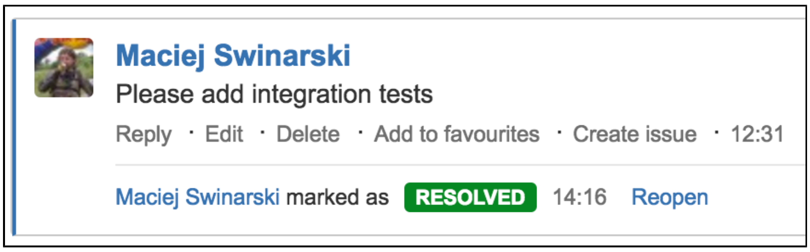


Maciej Swinarski
Please add integration tests

Reply · Edit · Delete · Add to favourites · Create issue · 12:31

Marek Parfianowicz marked as UNRESOLVED 13:46 Resolve

When the task requested by a comment is considered done, click **Resolve** to flag the comment as RESOLVED:

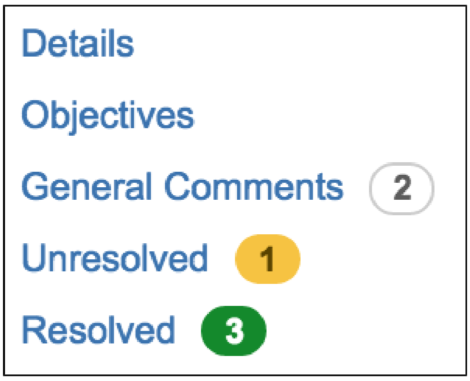


Maciej Swinarski
Please add integration tests

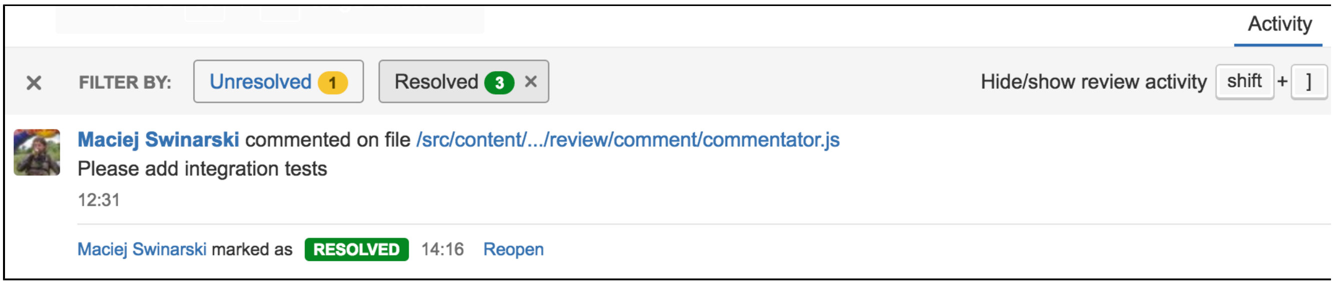
Reply · Edit · Delete · Add to favourites · Create issue · 12:31

Maciej Swinarski marked as RESOLVED 14:16 Reopen

All review participants can quickly track the unresolved comments with work still be done, and the comments that have already been resolved, by choosing **Unresolved** or **Resolved** at the left of the review:



On the **Review Activity** you can use the filter to quickly switch between all comments, comments marked as needing resolution and resolved comments:



Maciej Swinarski commented on file /src/content/.../review/comment/commentator.js
Please add integration tests
12:31

Maciej Swinarski marked as RESOLVED 14:16 Reopen

Sending a review's comments via email

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

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

1. In Crucible, navigate to the review in question.
2. Choose **Tools** > **Download as Text**.
3. Copy the on-screen text and paste it into an email.
4. Send.

Changeset discussions

When using Crucible with FishEye, you can have threaded discussions with other users, on any changeset. To start a discussion, you simply start by adding a comment to a changeset.

You need to be logged in to create changeset comments.

Adding comments to changesets

To add a comment to a changeset:

1. Click on a changeset on the **Commits** tab for the repository. Display comments by clicking **Discuss** at the upper right corner, or the speech bubble icon

   in the left margin.
2. Click **Add a comment** (under the repository details near the top left).
3. Type your comment. If required, you can tag your comment as being a defect note by clicking **Defect**.
4. Click **Post**.

Once submitted, others can respond to your comment by clicking **Reply**. Replies are threaded as separate comment discussions. You can right-click on the permalink icon to copy a link to the comment. The comment author can edit or delete their own comments.

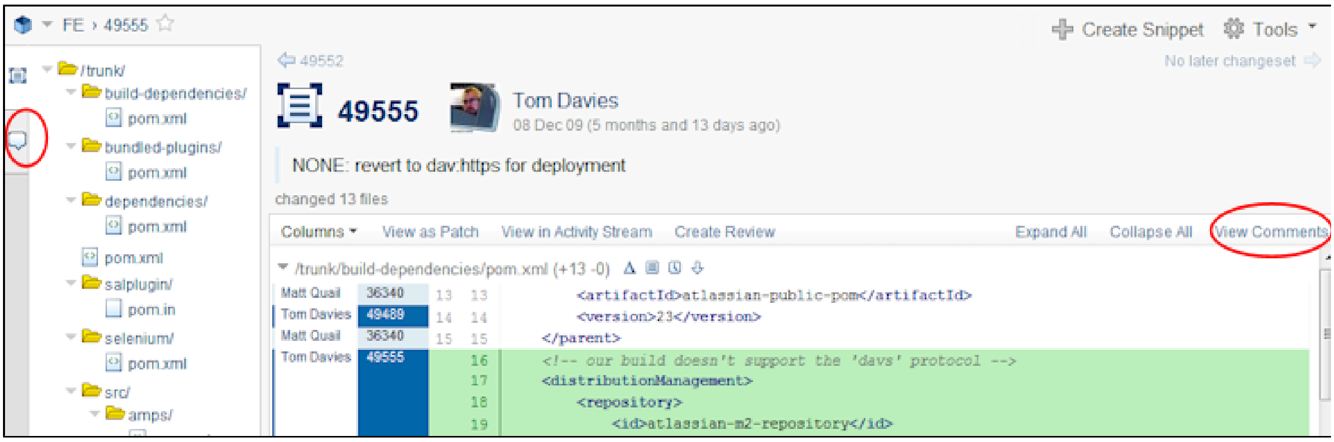
To hide the changeset comments, click the page icon

. You can display the comments panel by clicking the speech bubble icon

again.

As you compose a comment, it will auto-save periodically.

Screenshot: Opening Changeset Discussions



NONE: revert to dav:https for deployment

```
Matt Quail   36340  13  13          <artifactId>atlassian-public-pom</artifactId>
Tom Davies   49489  14  14          <version>23</version>
Matt Quail   36340  15  15      </parent>
Tom Davies   49555      16      <!-- our build doesn't support the 'davs' protocol -->
                        17      <distributionManagement>
                        18          <repository>
                        19              <id>atlassian-m2-repository</id>
```

Turning changeset discussions on and off

You can turn off changeset discussions in the Admin area:

1. In the Admin area, click **Repositories** (under 'Repository Settings' on the left).
2. Find your repository and choose **View** from the 'cog' menu in the Actions column.
3. Click **Other Settings** in the left panel.
4. Under 'Changeset Discussions' clear the **Allow changeset discussions** checkbox.

By default, changeset discussions are on.

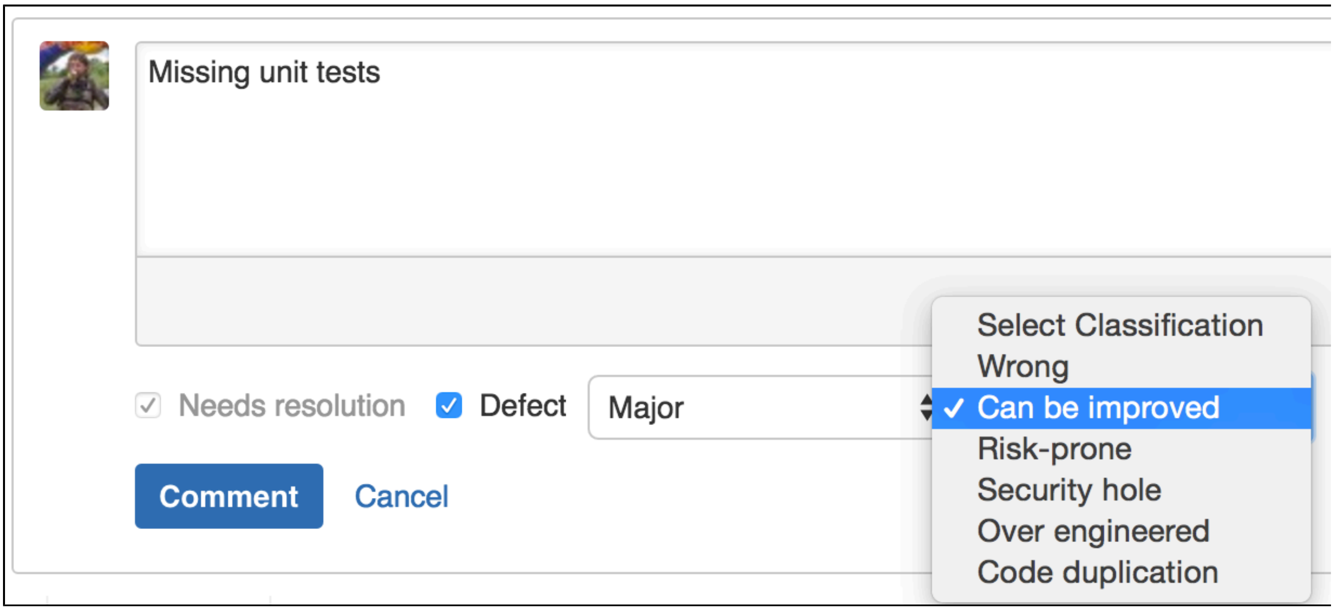
Notifications

- Comments show up in the activity stream,
- The author of the changeset will get email notifications when comments are added,
- Comment authors will get email notifications when someone replies to their comments.

Flagging defects

Comments in Crucible can be used to flag a defect in the code under review.

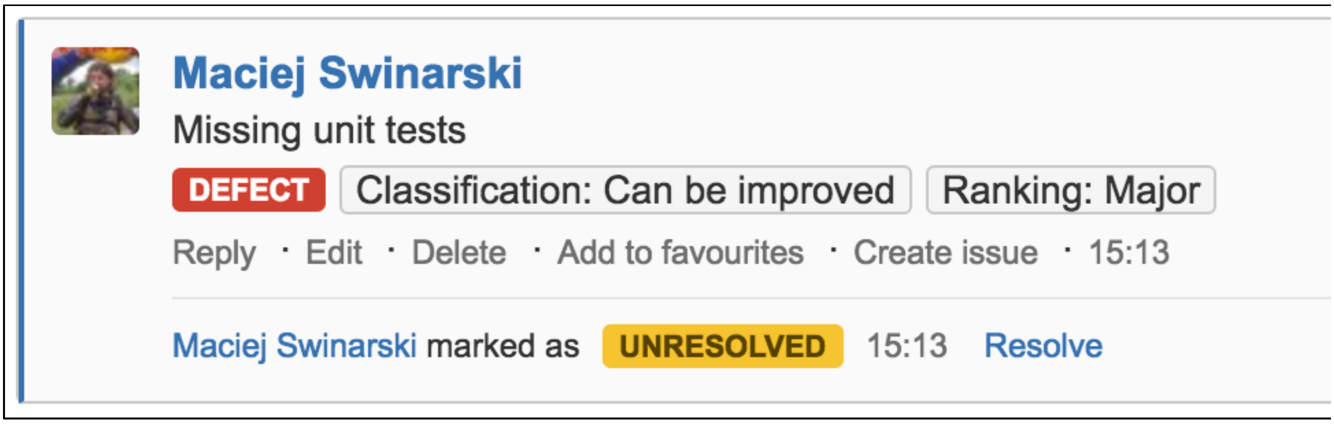
To do this, simply check **Defect** when adding a comment and select a category from the drop-down list.



Missing unit tests

☑ Needs resolution ☑ Defect Major

Any **Defect** comment will be also marked as **Needs resolution.**



Maciej Swinarski

Missing unit tests

DEFECT Classification: Can be improved Ranking: Major

Reply · Edit · Delete · Add to favourites · Create issue · 15:13

Maciej Swinarski marked as UNRESOLVED 15:13 Resolve

You may want to mark comments as defects in order to associate defect classifications, or simply to highlight to the author or moderator that the issue you raised in your comment requires attention. You can use the **with defects** filter to find files that have been flagged with defects.

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

ⓘ You can only use the defect classifications on comments that are not a reply to an existing comment.

Viewing reports

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

On this page:

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

See also:

- Viewing the Review Coverage report

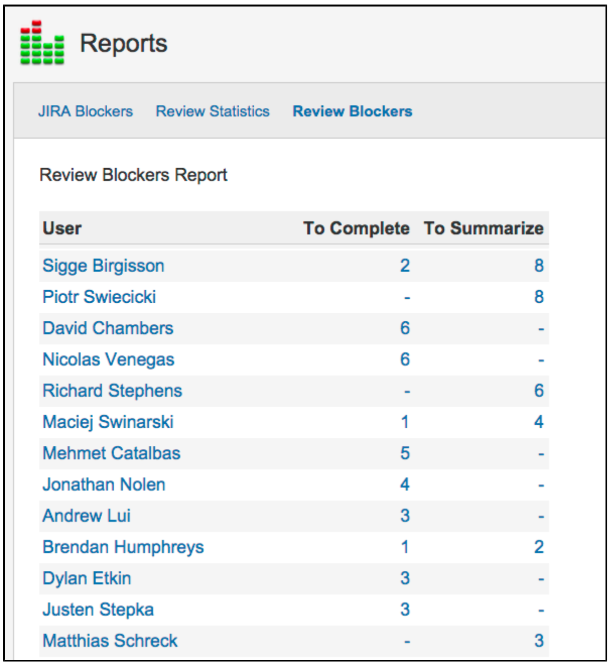
Viewing the Review Blockers report

To view a list of people who have open reviews assigned to them:

1. Click the dropdown arrow next to the **Reviews** tab at the top of the page and select **Reports**.
2. Click **Review Blockers** (under the 'Reports' sub-tab).

- Click a user's name to go to their 'Activity' screen.
- Click a number in the 'To Complete' or 'To Summarize' column to go to a list of reviews for that user.

Screenshot: 'Review Blockers' Report



Review Blockers Report

User	To Complete	To Summarize
Sigge Birgisson	2	8
Piotr Swiecicki	-	8
David Chambers	6	-
Nicolas Venegas	6	-
Richard Stephens	-	6
Maciej Swinarski	1	4
Mehmet Catalbas	5	-
Jonathan Nolen	4	-
Andrew Lui	3	-
Brendan Humphreys	1	2
Dylan Etkin	3	-
Justen Stepka	3	-
Matthias Schreck	-	3

Viewing the JIRA Blockers report

The 'JIRA Blockers' report shows you a list of users whose action is required on open reviews, for a particular set of JIRA issues. The reviews must be explicitly linked to a JIRA issue or mention a JIRA issue key in the summary or the objectives.

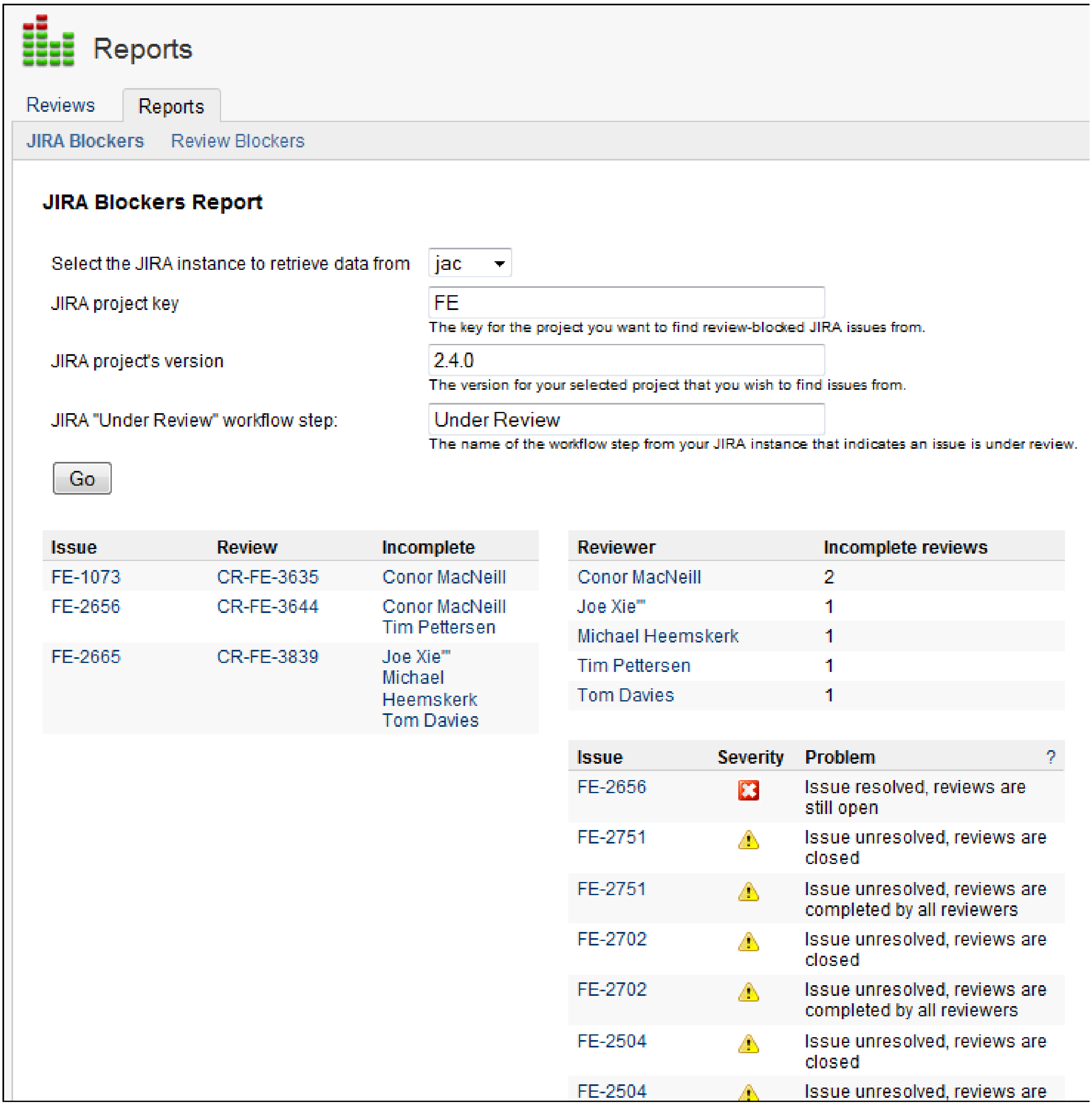
To view the 'JIRA Blockers' report:

1. Click the dropdown arrow next to the **Reviews** tab at the top of the page and select **Reports**.
2. Click **JIRA Blockers** (under the 'Reports**'** sub-tab).
3. Enter details for your JIRA server and project, and click **Go**. Note that the **JIRA project's version** field refers to the **Fix Version/s** field in your JIRA tickets.

The 'JIRA Blockers' report displays the following information:

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

Screenshot: 'JIRA Blockers' Report



JIRA Blockers Report

Select the JIRA instance to retrieve data from	jac
JIRA project key	FE The key for the project you want to find review-blocked JIRA issues from.
JIRA project's version	2.4.0 The version for your selected project that you wish to find issues from.
JIRA "Under Review" workflow step:	Under Review The name of the workflow step from your JIRA instance that indicates an issue is under review.

Issue	Review	Incomplete
FE-1073	CR-FE-3635	Conor MacNeill
FE-2656	CR-FE-3644	Conor MacNeill Tim Pettersen
FE-2665	CR-FE-3839	Joe Xie''' Michael Heemskerk Tom Davies

Reviewer	Incomplete reviews
Conor MacNeill	2
Joe Xie'''	1
Michael Heemskerk	1
Tim Pettersen	1
Tom Davies	1

Issue	Severity	Problem ?
FE-2656	☒	Issue resolved, reviews are still open
FE-2751	⚠	Issue unresolved, reviews are closed
FE-2751	⚠	Issue unresolved, reviews are completed by all reviewers
FE-2702	⚠	Issue unresolved, reviews are closed
FE-2702	⚠	Issue unresolved, reviews are completed by all reviewers
FE-2504	⚠	Issue unresolved, reviews are closed
FE-2504	⚠	Issue unresolved, reviews are

Review Coverage report

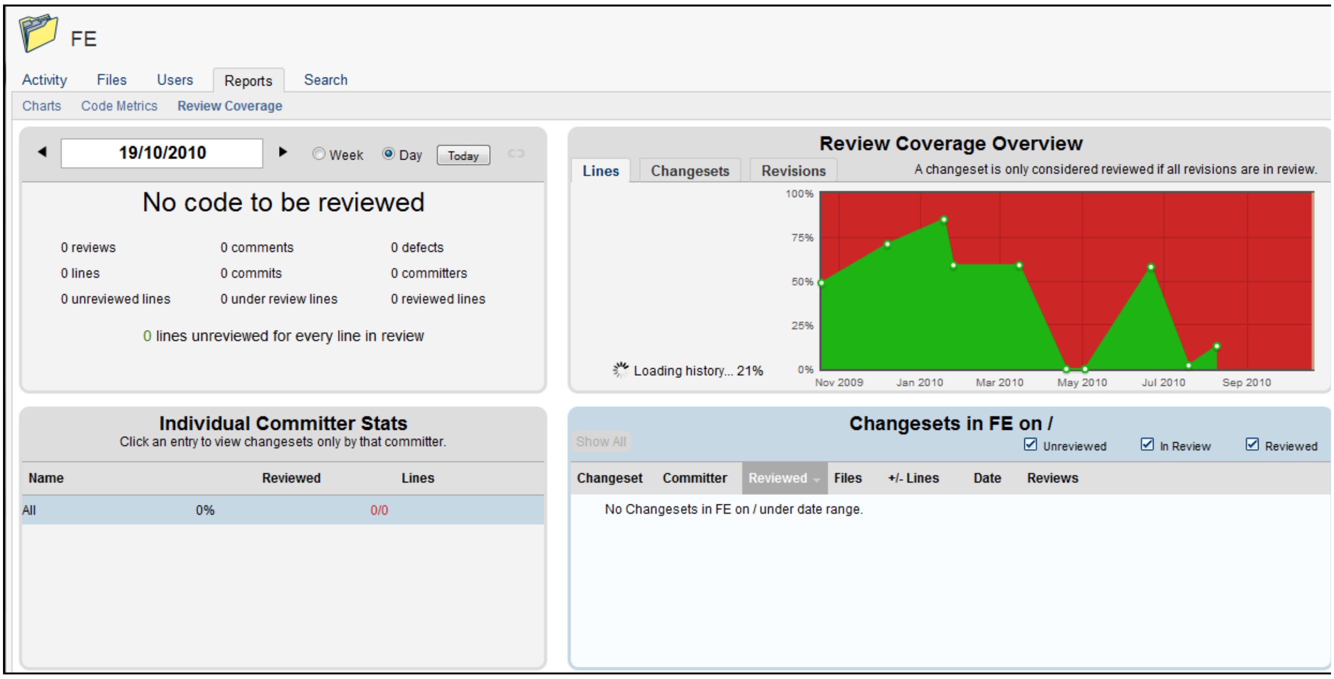
Crucible has useful reports that show you detailed statistics on review activity. The Review Coverage report allows you to see how much of the code, and which files, in your repository have been reviewed, and when. You can also access the reviews.

ⓘ This feature requires **FishEye integrated with Crucible**.

On this page:

- Opening the Review Coverage report
- Using the Summary Panel
- Using the Review Coverage Overview
- Using the Individual Committer Stats panel
- Using the Changesets panel

Screenshot: The Review Coverage report



No code to be reviewed

0 reviews | 0 comments | 0 defects
0 lines | 0 commits | 0 committers
0 unreviewed lines | 0 under review lines | 0 reviewed lines

0 lines unreviewed for every line in review

Review Coverage Overview

A changeset is only considered reviewed if all revisions are in review.

Individual Committer Stats

Click an entry to view changesets only by that committer.

Name	Reviewed	Lines
All	0%	0/0

Changesets in FE on /

Changeset	Committer	Reviewed	Files	+/- Lines	Date	Reviews

No Changesets in FE on / under date range.

Opening the Review Coverage report

To open the Review Coverage report:

1. Click **Repositories** and choose a repository. The repository you chose sets the scope for the report.
2. If desired, navigate down the tree (in the lefthand panel) and click the desired path you want to view coverage on.
3. Click **Reports** in the secondary toolbar.
4. Click **Review Coverage** from the list of reports in the upper panel.

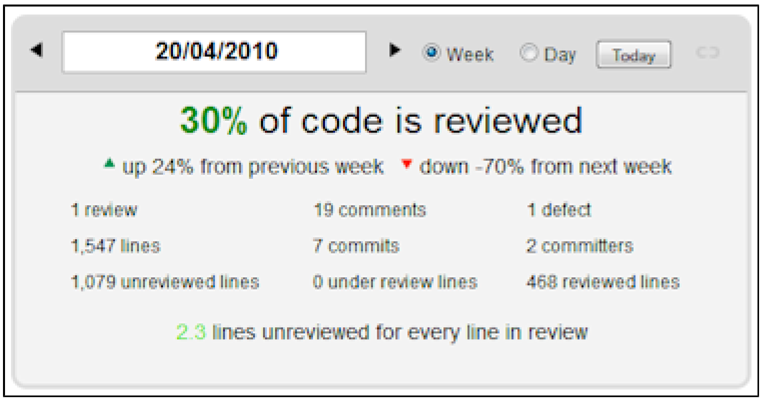
ⓘ You can view coverage of any path by navigating down the tree to the desired path you want to view coverage on, before clicking on the **Reports** tab.

Using the Summary Panel

The summary panel displays the following metrics for your selected repository:

- Overall review coverage percentage.
- Change in review coverage percentage since the last reporting period.
- Total number of reviews.
- Total number of comments.
- Total number of reported defects.
- Total number of Lines of Code (LOC).
- Total number of commits.
- Total number of committers.
- Total number of unreviewed lines.
- Total number of lines under review.
- Total number of reviewed lines.
- A ratio of the number of lines unreviewed against reviewed Lines of Code (LOC).

Screenshot: Summary Panel in the Review Coverage report



30% of code is reviewed

▲ up 24% from previous week ▼ down -70% from next week

1 review	19 comments	1 defect
1,547 lines	7 commits	2 committers
1,079 unreviewed lines	0 under review lines	468 reviewed lines

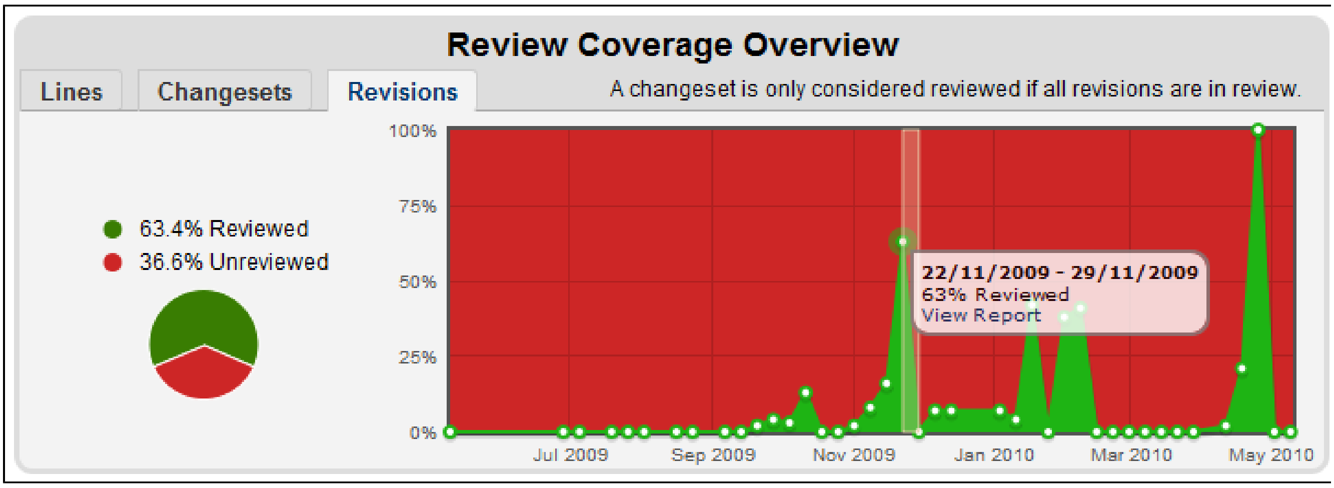
2.3 lines unreviewed for every line in review

Note: Code coverage metrics only apply for post-commit reviews.

Using the Review Coverage Overview

The Review Coverage Overview shows a timeline of reviews, compared against their percentage of coverage. Hover your mouse cursor over the data points on the graph to see granular information and click through to a detailed weekly report.
You can click the tabs to view the coverage expressed as a percentage of lines of code, changesets or revisions.

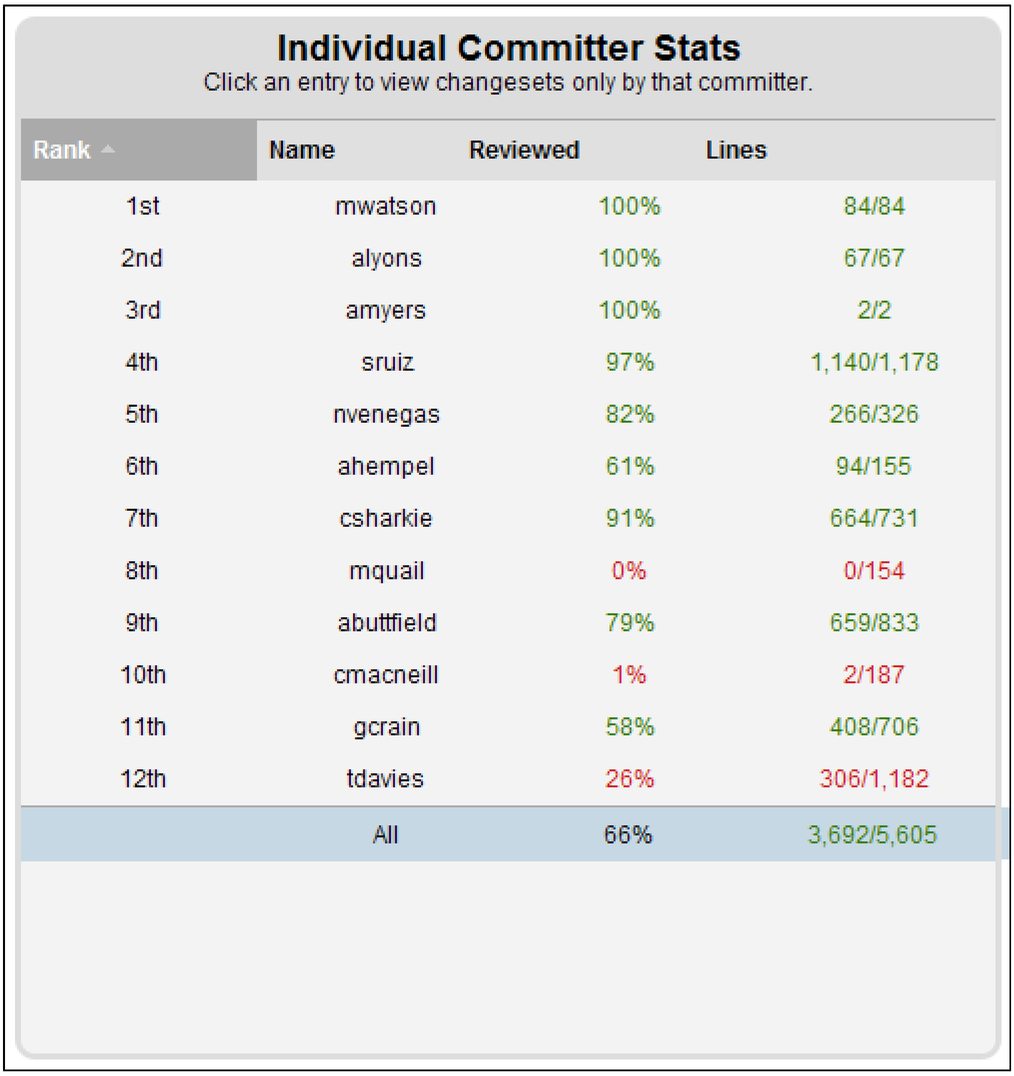
Screenshot: Overview Panel in the Review Coverage report



Using the Individual Committer Stats panel

The Individual Committer Stats panel lets you choose a user from your Crucible instance and see all the changesets by that committer.

Screenshot: Individual Committer Statistics in the Review Coverage report



Individual Committer Stats

Click an entry to view changesets only by that committer.

Rank	Name	Reviewed	Lines
1st	mwatson	100%	84/84
2nd	alyons	100%	67/67
3rd	amyers	100%	2/2
4th	sruiz	97%	1,140/1,178
5th	nvenegas	82%	266/326
6th	ahempel	61%	94/155
7th	csharkie	91%	664/731
8th	mquail	0%	0/154
9th	abuttfield	79%	659/833
10th	cmacneill	1%	2/187
11th	gcrain	58%	408/706
12th	tdavies	26%	306/1,182
	All	66%	3,692/5,605

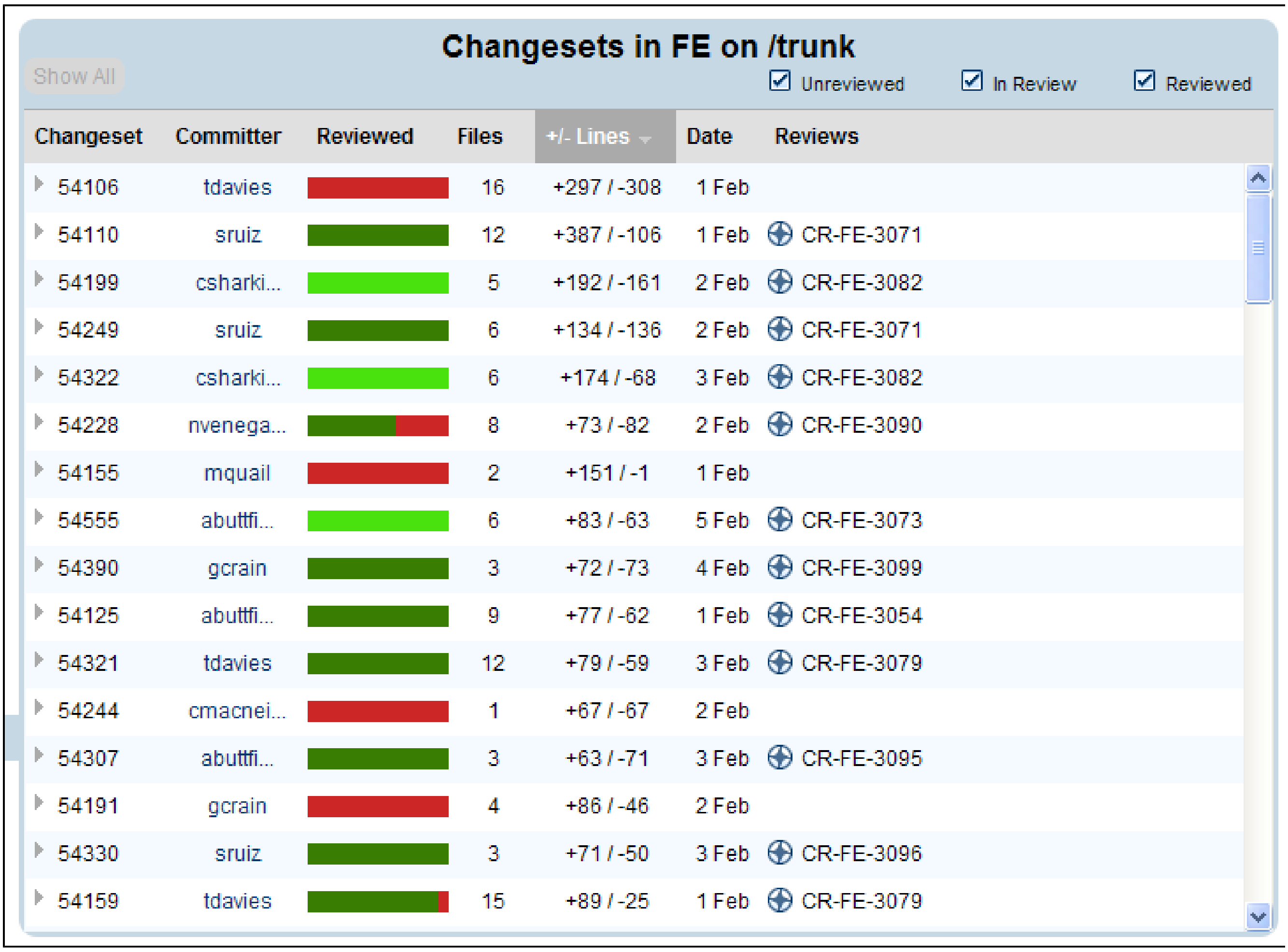
Using the Changesets panel

The Changesets panel lets you see changesets from your Crucible instance (for the time period of the report), and their level of review coverage. This information can be sorted by the columns in this view and uses color coding to denote review coverage (listed in the table below).

Color Key

Color	State
dark green	reviewed
light green	in review
red	not reviewed

Screenshot: Changesets panel in the Review Coverage report



Changesets in FE on /trunk

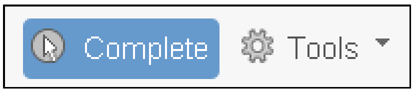
Show All ☑ Unreviewed ☑ In Review ☑ Reviewed

Changeset	Committer	Reviewed	Files	+/- Lines	Date	Reviews
54106	tdavies	🟥	16	+297 / -308	1 Feb	
54110	sruiz	🟩	12	+387 / -106	1 Feb	CR-FE-3071
54199	csharki...	🟩	5	+192 / -161	2 Feb	CR-FE-3082
54249	sruiz	🟩	6	+134 / -136	2 Feb	CR-FE-3071
54322	csharki...	🟩	6	+174 / -68	3 Feb	CR-FE-3082
54228	nvenega...	🟩🟥	8	+73 / -82	2 Feb	CR-FE-3090
54155	mquail	🟥	2	+151 / -1	1 Feb	
54555	abuttfi...	🟩	6	+83 / -63	5 Feb	CR-FE-3073
54390	gcrain	🟩	3	+72 / -73	4 Feb	CR-FE-3099
54125	abuttfi...	🟩	9	+77 / -62	1 Feb	CR-FE-3054
54321	tdavies	🟩	12	+79 / -59	3 Feb	CR-FE-3079
54244	cmacnei...	🟥	1	+67 / -67	2 Feb	
54307	abuttfi...	🟩	3	+63 / -71	3 Feb	CR-FE-3095
54191	gcrain	🟥	4	+86 / -46	2 Feb	
54330	sruiz	🟩	3	+71 / -50	3 Feb	CR-FE-3096
54159	tdavies	🟩🟥	15	+89 / -25	1 Feb	CR-FE-3079

Completing your review

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

To complete your individual review, go to the review and click **Complete** at the top of the screen, next to the **Too ls** menu:



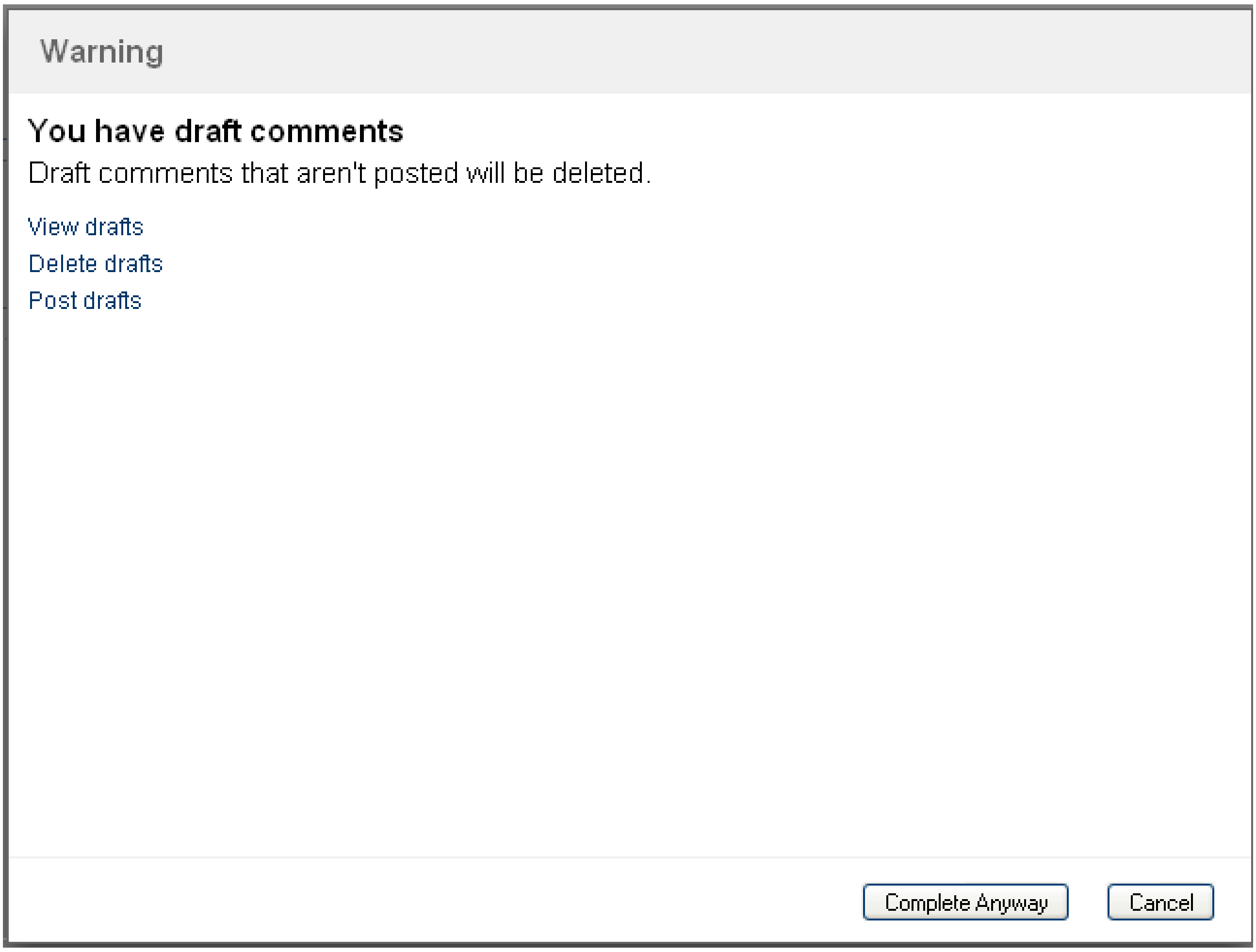
Only people with the 'Complete' permission can complete a review.

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

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

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

Screenshot: Draft comments

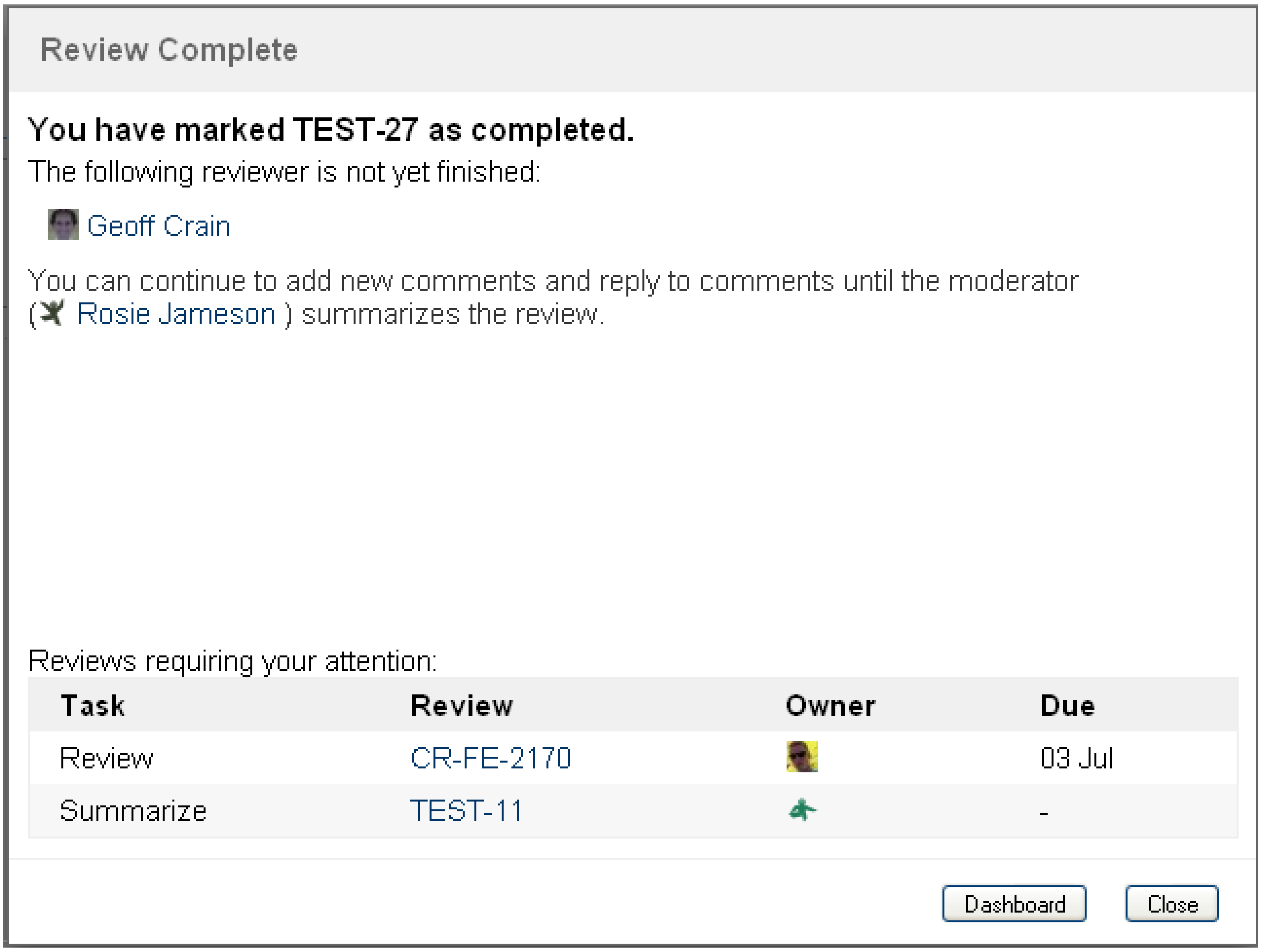


Warning

You have draft comments

Draft comments that aren't posted will be deleted.

Screenshot: Review complete



Review Complete

You have marked TEST-27 as completed.

The following reviewer is not yet finished:

Geoff Crain

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

Reviews requiring your attention:

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

Using the Review History dialog

The Review History dialog shows a chronological list of interactions within a review. You can see rich information about those interactions and control their display. You can sort the information by date, actor, or action.

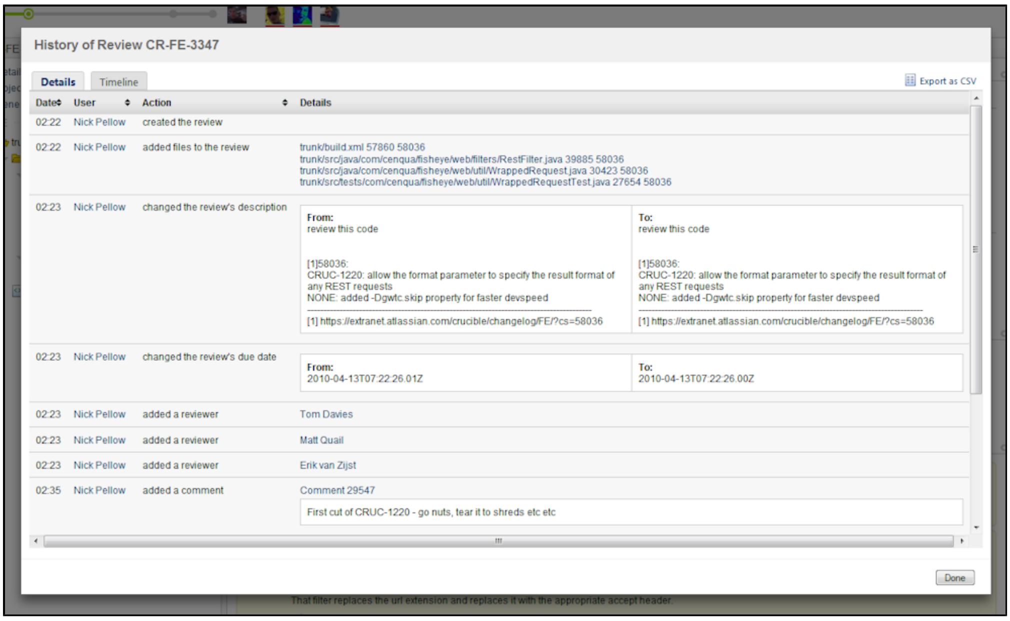
To open the Review History dialog:

1. Open a review in Crucible.
2. Choose **Tools** > **View Review History** from the top right of the screen.

Click the **Timeline** tab at the top of the History dialog to see the review's events in a horizontal calendar. You can drag the calendar and the timeline at the bottom to scroll to other events.

Click **Export as CSV** near the top right to export the entire review history, allowing for easy data import into a spreadsheet or other application.

Screenshot: The Crucible Review History dialog

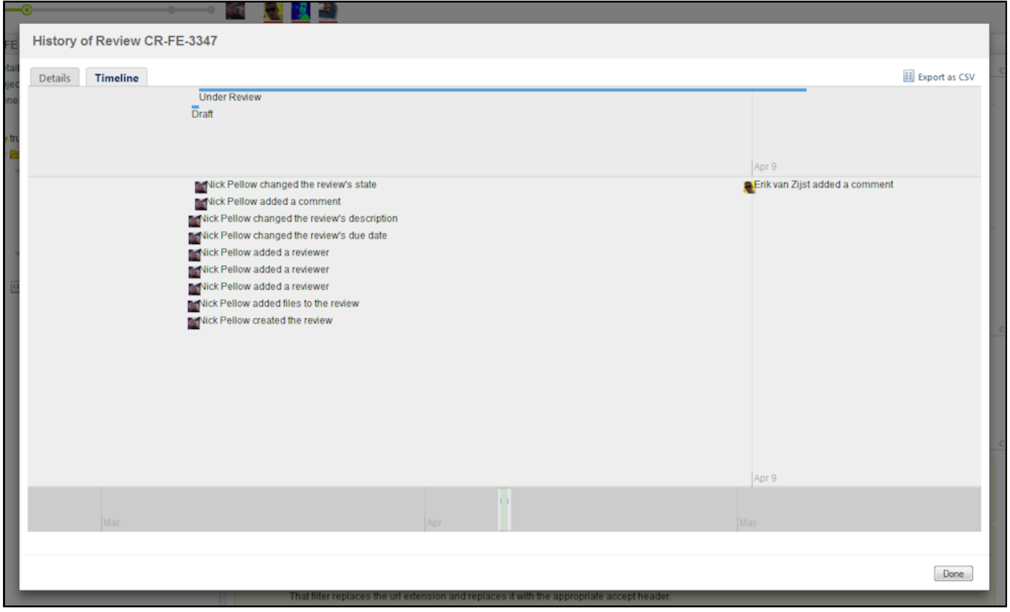


History of Review CR-FE-3347

Date	User	Action	Details
02:22	Nick Pellow	created the review	
02:22	Nick Pellow	added files to the review	trunk/build.xml 57860 58036 trunk/src/java/com/cenqua/fisheye/web/filters/RestFilter.java 39885 58036 trunk/src/java/com/cenqua/fisheye/web/util/WrappedRequest.java 30423 58036 trunk/src/tests/com/cenqua/fisheye/web/util/WrappedRequestTest.java 27654 58036
02:23	Nick Pellow	changed the review's description	**From:** review this code [1]58036: CRUC-1220: allow the format parameter to specify the result format of any REST requests NONE: added -Dgwtc.skip property for faster devspeed [1] https://extranet.atlassian.com/crucible/changelog/FE/?cs=58036 **To:** review this code [1]58036: CRUC-1220: allow the format parameter to specify the result format of any REST requests NONE: added -Dgwtc.skip property for faster devspeed [1] https://extranet.atlassian.com/crucible/changelog/FE/?cs=58036
02:23	Nick Pellow	changed the review's due date	**From:** 2010-04-13T07:22:26.01Z **To:** 2010-04-13T07:22:26.00Z
02:23	Nick Pellow	added a reviewer	Tom Davies
02:23	Nick Pellow	added a reviewer	Matt Quail
02:23	Nick Pellow	added a reviewer	Erik van Zijst
02:35	Nick Pellow	added a comment	Comment 29547 First cut of CRUC-1220 - go nuts, tear it to shreds etc etc

That filter replaces the url extension and replaces it with the appropriate accept header.

Screenshot: The Crucible Review History Dialog, Timeline View



History of Review CR-FE-3347

Nick Pellow changed the review's state
Nick Pellow added a comment
Nick Pellow changed the review's description
Nick Pellow changed the review's due date
Nick Pellow added a reviewer
Nick Pellow added a reviewer
Nick Pellow added a reviewer
Nick Pellow added files to the review
Nick Pellow created the review

Erik van Zijst added a comment

That filter replaces the url extension and replaces it with the appropriate accept header.

Summarizing and closing the review

Summarize is an optional step before closing a review.

To enable or disable the Summarize step, you will have to configure the permission in your Permission Scheme. Crucible ships with two permission schemes:

- 'Agile' - the summarize step is disabled for all users
- 'Default' - the summarize step is enabled for the moderator

You can choose to either *summarize a review* or *close a review* at any time, given that your Permission Scheme allows it. You can skip the summarize step by directly clicking **Close**.

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

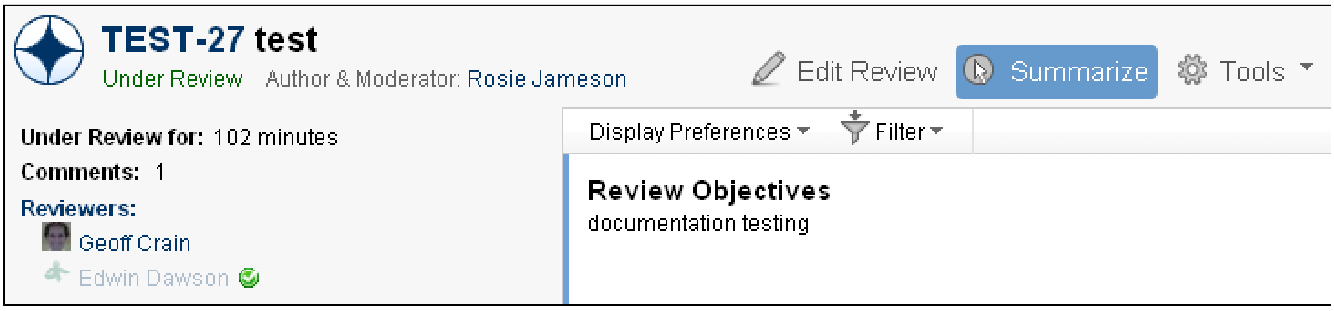
Normally, we recommend that you wait for all reviewers to complete their reviews, before summarizing or closing the review.
The reviews that the reviewers have completed will be in your **Ready to Close** menu on the Dashboard.

To summarize a review,

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

- On clicking **Summarize**, you may be prompted to confirm the action if there are incomplete reviews or draft comments in the review. These are warnings, however; the review can still be summarized and closed.

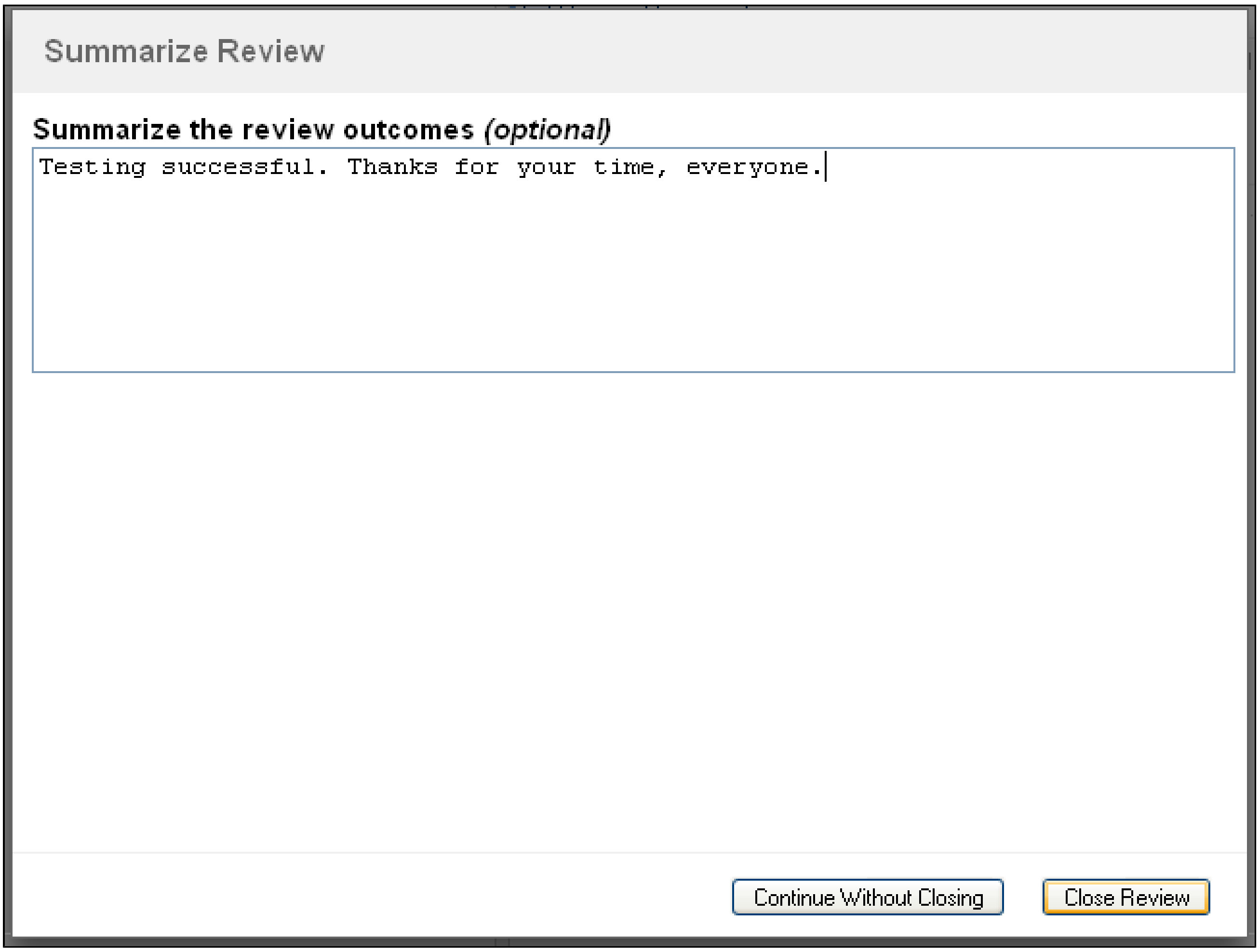
Screenshot: 'Summarize' button. We can see that Geoff Crain has still not finished reviewing, because there is no green tick next to his name.



Under Review for: 102 minutes
Comments: 1
Reviewers:
Geoff Crain
Edwin Dawson ✓

Review Objectives
documentation testing

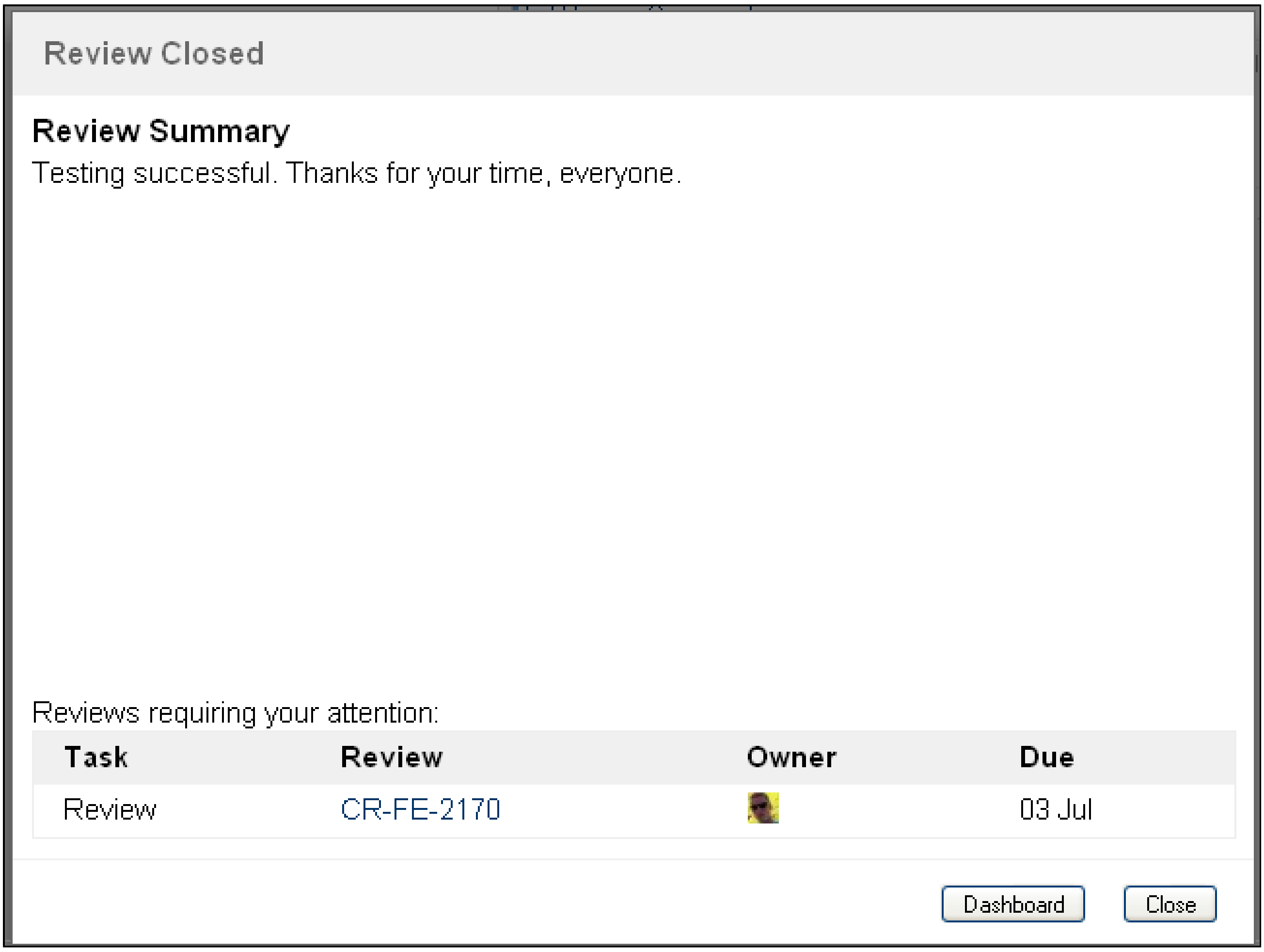
Once the review is in the 'Summarize' state, the moderator can optionally add a review summary, for example, to describe the outcomes/tasks/etc:



Summarize the review outcomes *(optional)*

Testing successful. Thanks for your time, everyone.

Screenshot: Review Closed



Review Closed

Review Summary

Testing successful. Thanks for your time, everyone.

Reviews requiring your attention:

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

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

- Reviews in the 'Review' or '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.
- See this page for a list of the states that a review can be in.

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

Managing your reviews

See:

- Using Review Reminders
- Moving a review to another project
- Using Progress Tracking
- Using Time Tracking
- Viewing people's statistics in Crucible
- Viewing Project Statistics
- Deleting a review

Using Review Reminders

Crucible will automatically send reviewers a reminder email one working day before the deadline.

Review authors and moderators can also do the following:

- Send manual reminders to reviewers whose work is still pending.
- Configure preset reminders for reviews that have a deadline.

Reminders are only sent if Crucible's SMTP server is configured. Please see Configuring the SMTP server.

Preset Reminders

When a review has a deadline (due date), you can have Crucible send a preset reminder to all of the pending reviewers, some number of working days before the deadline.

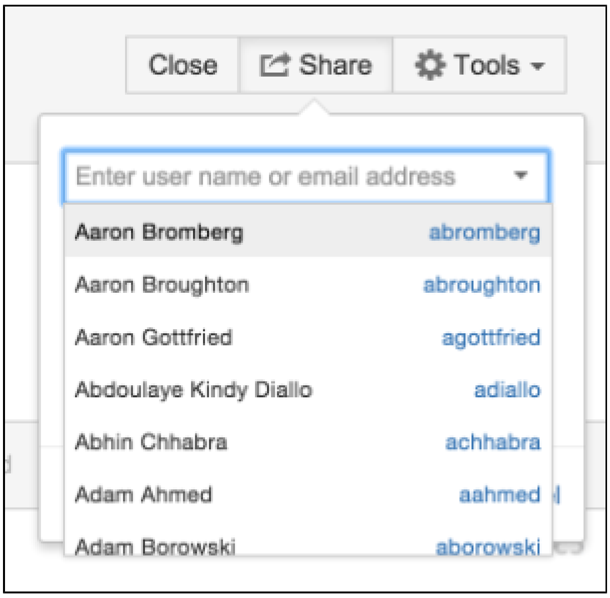
To add a reminder, firstly edit the review, then click **Add a reminder**. You can edit the reminder period.

ℹ The **Send Reminder** setting is only available if the review has a due date set.

Manual Reminders

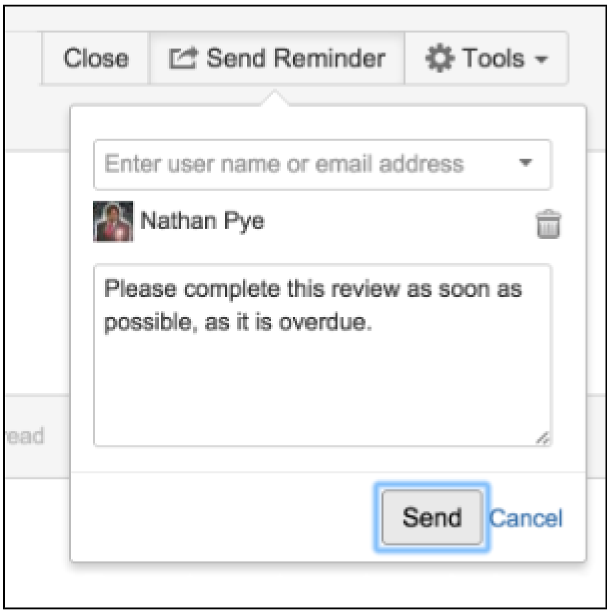
You can send a reminder to all the reviewers that have not yet completed their review:

Before the review is due, click **Share** and add recipients and a message. Besides sending reminders to participants, the **Share** option can also be used to share the review with people external to it:



Aaron Bromberg	abromberg
Aaron Broughton	abroughton
Aaron Gottfried	agottfried
Abdoulaye Kindy Diallo	adiallo
Abhin Chhabra	achhabra
Adam Ahmed	aahmed
Adam Borowski	aborowski

If the review is overdue, click **Send Reminder**. The message is pre-populated with recipients who have not yet completed their review.



Nathan Pye

Please complete this review as soon as possible, as it is overdue.

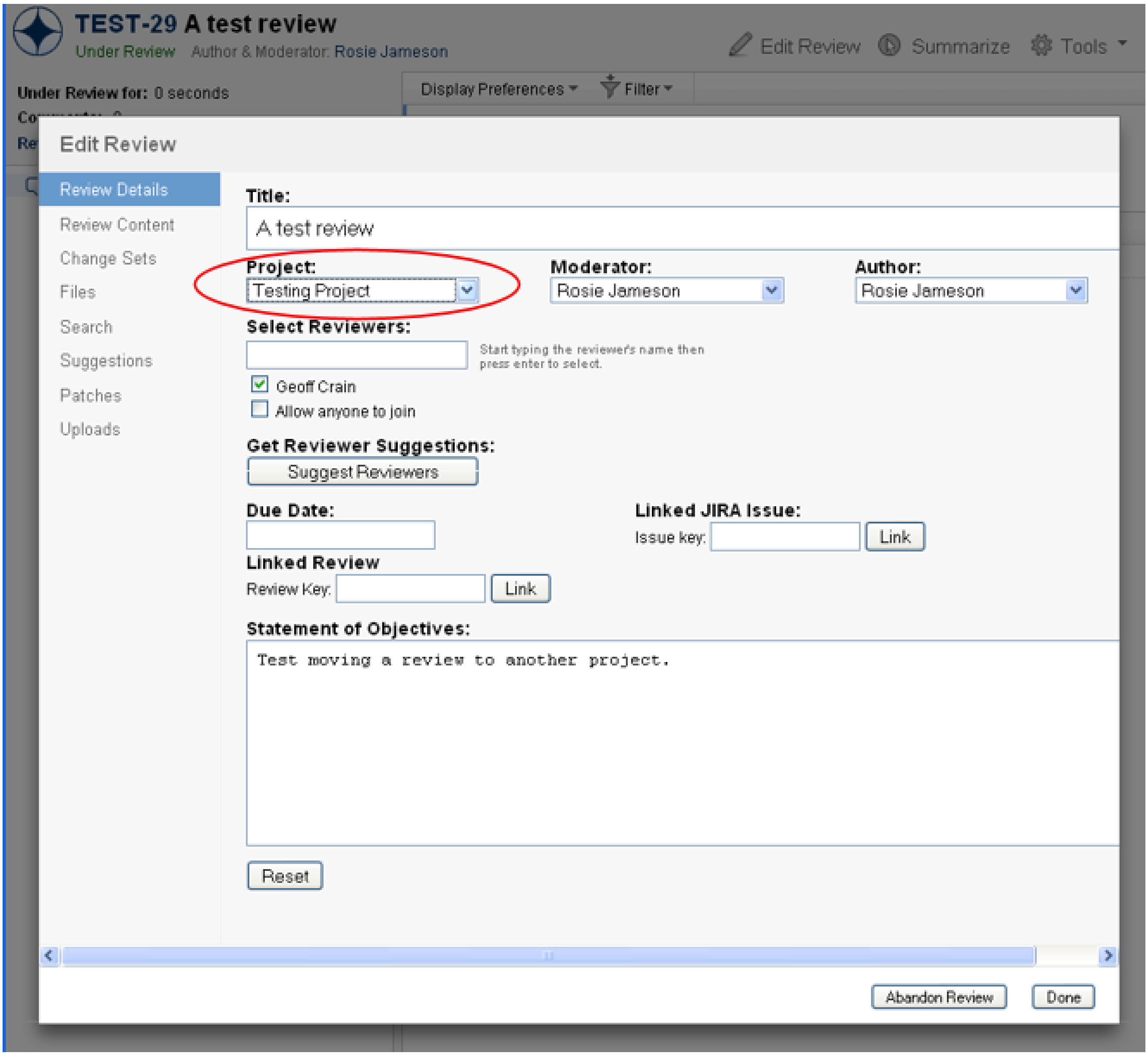
Moving a review to another project

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

To move a review between projects:

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

Screenshot: Changing a Review's Parent Project



TEST-29 A test review

Under Review Author & Moderator: Rosie Jameson

Under Review for: 0 seconds

Edit Review

Title: A test review

Project: Testing Project

Moderator: Rosie Jameson

Author: Rosie Jameson

Select Reviewers:

☑ Geoff Crain

□ Allow anyone to join

Statement of Objectives: Test moving a review to another project.

Using Progress Tracking

This page contains instruction on how to use progress tracking in Crucible.

On this page:

How progress tracking works in Crucible

As you work your way through the files in a review, Crucible tracks the ones you have viewed. Whenever you open a file for review, Crucible will automatically mark it as read.

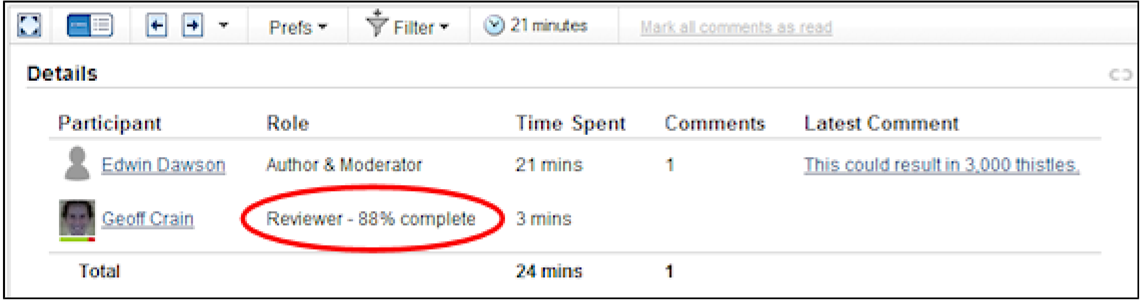
When participating in iterative reviews, progress tracking also takes lines of code and revisions into account.

Viewing the progress tracking totals

The 'Details' section shows a summary of the progress of each participant through the files in the review.

If there is only one file in the review, then the progress tracked will either show 0% or 100%.

Screenshot: Viewing the Progress Tracking Totals



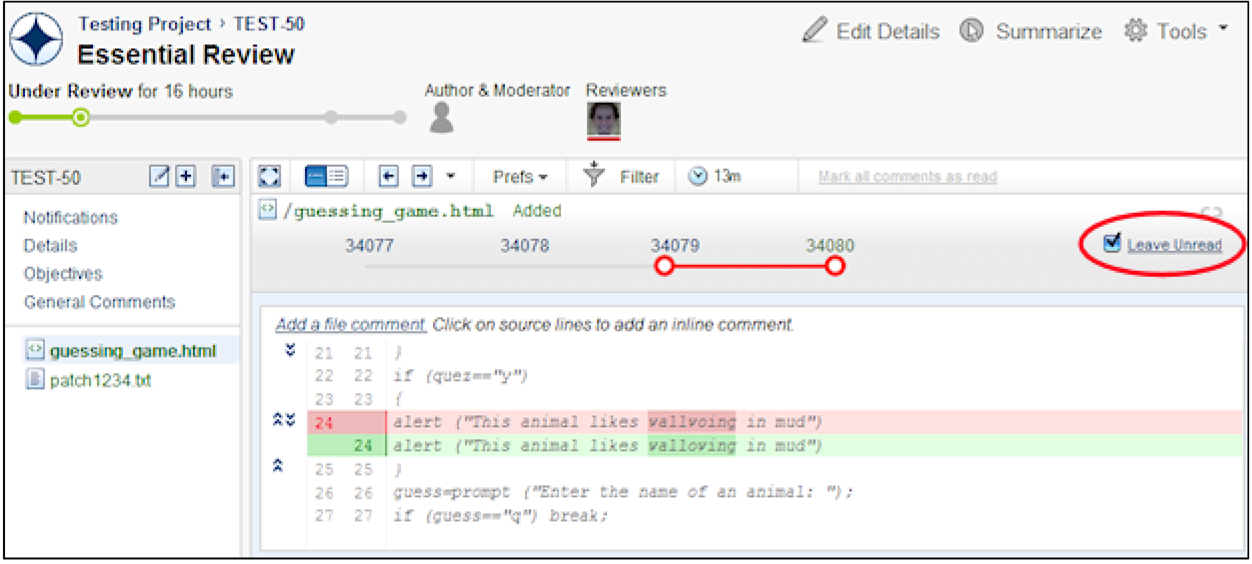
Details

Participant	Role	Time Spent	Comments	Latest Comment
Edwin Dawson	Author & Moderator	21 mins	1	This could result in 3,000 thistles.
Geoff Crain	Reviewer - 88% complete	3 mins		
Total		24 mins	1	

How to adjust progress tracking on a review

You can mark a file as unread by clicking on its name to view the file's contents, and then clicking **Leave Unread** (at the right of the file's toolbar). This file is now not included in your progress percentage.

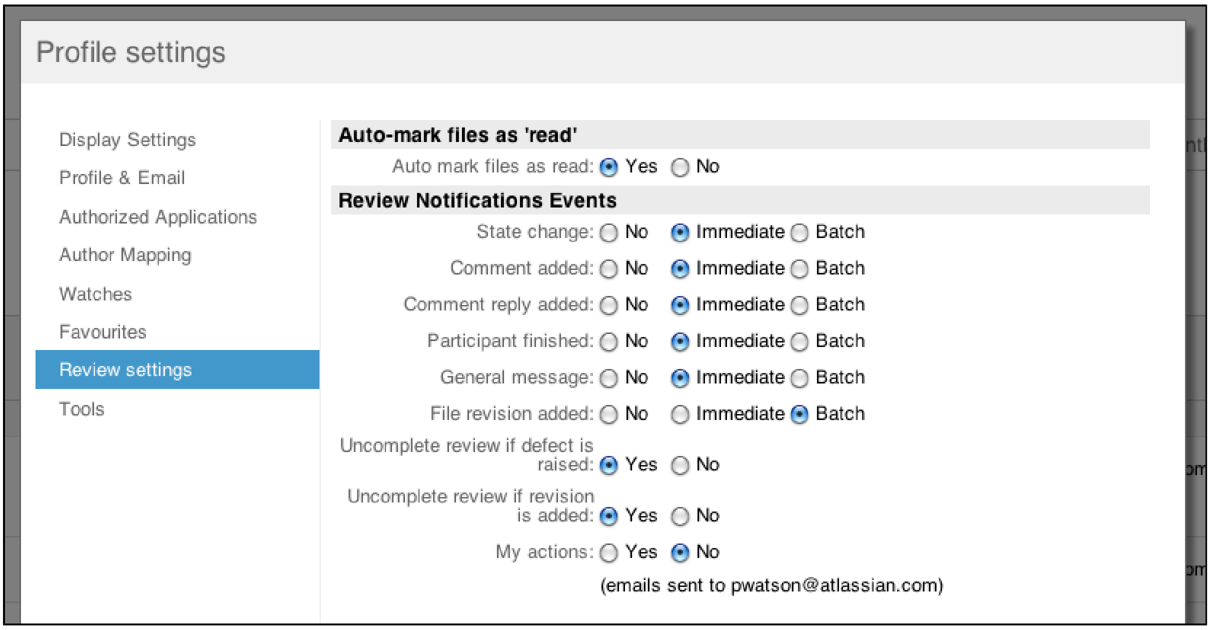
Screenshot: Marking a File as Unread



```
}
if (quez=="y")
{
alert ("This animal likes wallvoing in mud")
alert ("This animal likes walloving in mud")
}
guess=prompt ("Enter the name of an animal: ");
if (guess=="q") break;
```

Adjusting settings for progress tracking

Progress tracking is a configurable user preference – choose **Profile settings** from your user menu. On the **Rev iew settings** tab. **Auto-mark files as read** is on by default – when set to off, you have to mark files as read or unread yourself.



Profile settings

- Display Settings
- Profile & Email
- Authorized Applications
- Author Mapping
- Watches
- Favourites
- Review settings
- Tools

Auto-mark files as 'read'

Auto mark files as read: ● Yes ○ No

Review Notifications Events

State change: ○ No ● Immediate ○ Batch

Comment added: ○ No ● Immediate ○ Batch

Comment reply added: ○ No ● Immediate ○ Batch

Participant finished: ○ No ● Immediate ○ Batch

General message: ○ No ● Immediate ○ Batch

File revision added: ○ No ○ Immediate ● Batch

Uncomplete review if defect is raised: ● Yes ○ No

Uncomplete review if revision is added: ● Yes ○ No

My actions: ○ Yes ● No

(emails sent to pwatson@atlassian.com)

Further reading

You may also want to learn about Crucible's Time Tracking feature.

Using Time Tracking

This page contains instruction on how to use time tracking in Crucible.

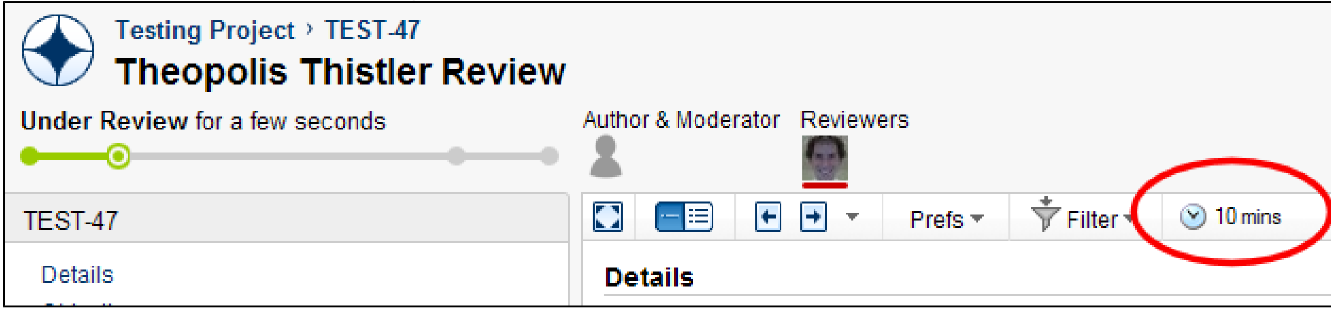
On this page:

- How time tracking works in Crucible
- How to adjust the time tracked on a review
- Viewing the time tracking totals
- JIRA integration
- Further reading

How time tracking works in Crucible

Crucible will automatically track the time you spend in a Crucible review. When you open a file for review, a counter in the Review Details panel starts. The time is added to your total when you leave the review screen.

Screenshot: Crucible Time Tracking



Testing Project › TEST-47

Theopolis Thistler Review

Under Review for a few seconds

Author & Moderator Reviewers

TEST-47

Details

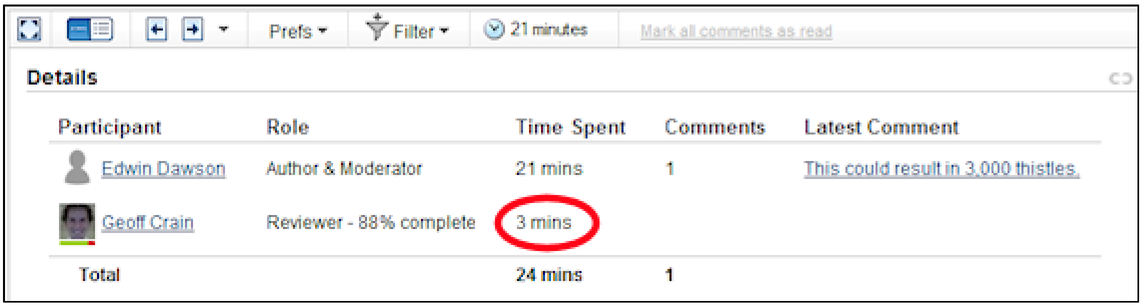
How to adjust the time tracked on a review

You can click and type in the time tracking control to adjust the time you have spent during the session.

Viewing the time tracking totals

The 'Details' section shows a summary of the progress and time tracked on each file.

Screenshot: Crucible Tracking Totals



Details

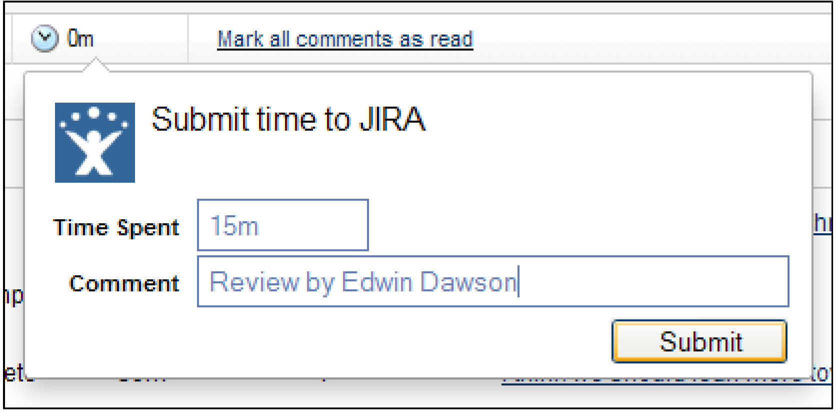
Participant	Role	Time Spent	Comments	Latest Comment
Edwin Dawson	Author & Moderator	21 mins	1	This could result in 3,000 thistles.
Geoff Crain	Reviewer - 88% complete	3 mins		
Total		24 mins	1	

JIRA integration

Using Crucible when integrated with JIRA, you can update time tracking from the following locations:

- The confirmation dialog for a reviewer completing a review,
- The confirmation dialog on closing a review,
- The regular toolbar location in Crucible.

Screenshot: JIRA Time Tracking Integration



Submit time to JIRA

Time Spent: 15m

Comment: Review by Edwin Dawson

Further reading

You may also want to learn about Crucible's Progress Tracking feature.

Viewing people's statistics in Crucible

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

On this page:

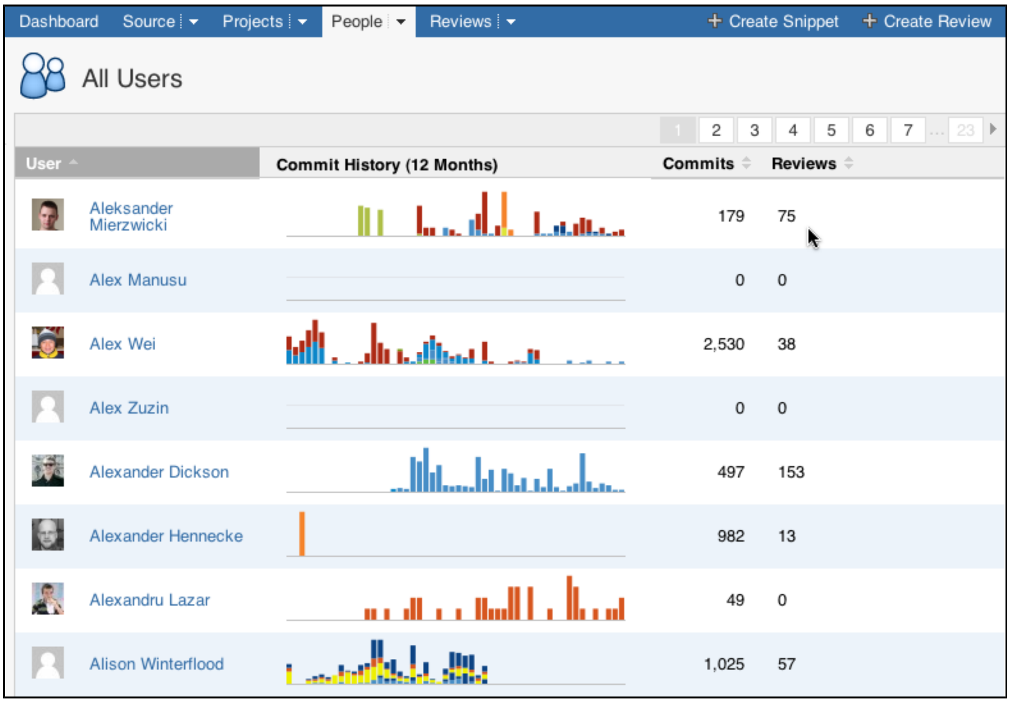
- Opening the list of People
- Viewing a Person's Activity Screen
- Viewing charts of a person's activity

Opening the list of People

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

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

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



User	Commit History (12 Months)	Commits	Reviews
Aleksander Mierzwicki		179	75
Alex Manusu		0	0
Alex Wei		2,530	38
Alex Zuzin		0	0
Alexander Dickson		497	153
Alexander Hennecke		982	13
Alexandru Lazar		49	0
Alison Winterflood		1,025	57

Viewing a Person's Activity Screen

Click on a username to see a listing of activity for them as well as charts showing statistics for their activity.

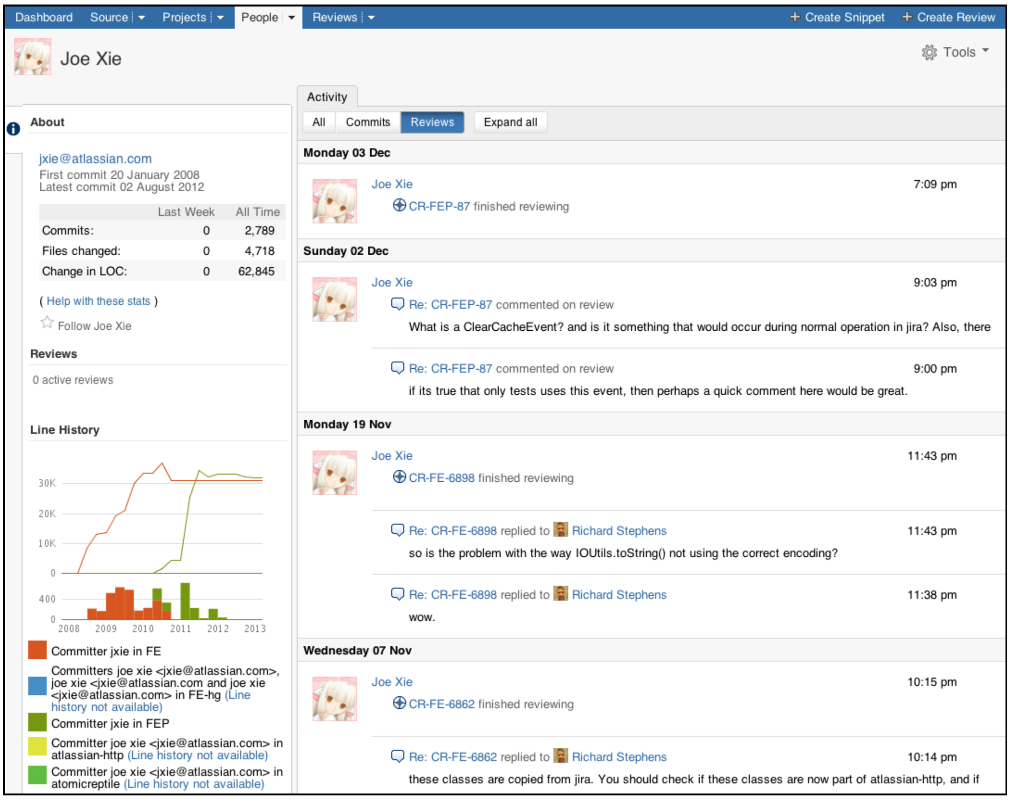
The right hand pane displays a list of all activity for this user. You can:

- click the icons to view full commit information in FishEye
- click JIRA issue names to open the work ticket on an item
- click the long button to see the list of files in context
- click the star icon to add an item to your favorites.

The left hand pane displays charts around this activity, including:

- number of active reviews
- charted history of lines of code
- code committing activity
- general statistics.

Screenshot: The People Activity Screen in Crucible



Joe Xie

About

jxie@atlassian.com
First commit 20 January 2008
Latest commit 02 August 2012

	Last Week	All Time
Commits:	0	2,789
Files changed:	0	4,718
Change in LOC:	0	62,845

(Help with these stats)

☆ Follow Joe Xie

Reviews

0 active reviews

Line History

🟥 Committer jxie in FE

🟦 Committers joe xie <jxie@atlassian.com>, joe xie <jxie@atlassian.com and joe xie <jxie@atlassian.com> in FE-hg (Line history not available)

🟩 Committer jxie in FEP

🟨 Committer joe xie <jxie@atlassian.com> in atlassian-http (Line history not available)

🟩 Committer joe xie <jxie@atlassian.com> in atomicreptile (Line history not available)

Monday 03 Dec

Joe Xie 7:09 pm
CR-FEP-87 finished reviewing

Sunday 02 Dec

Joe Xie 9:03 pm
Re: CR-FEP-87 commented on review
What is a ClearCacheEvent? and is it something that would occur during normal operation in jira? Also, there

Re: CR-FEP-87 commented on review 9:00 pm
if its true that only tests uses this event, then perhaps a quick comment here would be great.

Monday 19 Nov

Joe Xie 11:43 pm
CR-FE-6898 finished reviewing

Re: CR-FE-6898 replied to Richard Stephens 11:43 pm
so is the problem with the way IOUtils.toString() not using the correct encoding?

Re: CR-FE-6898 replied to Richard Stephens 11:38 pm
wow.

Wednesday 07 Nov

Joe Xie 10:15 pm
CR-FE-6862 finished reviewing

Re: CR-FE-6862 replied to Richard Stephens 10:14 pm
these classes are copied from jira. You should check if these classes are now part of atlassian-http, and if

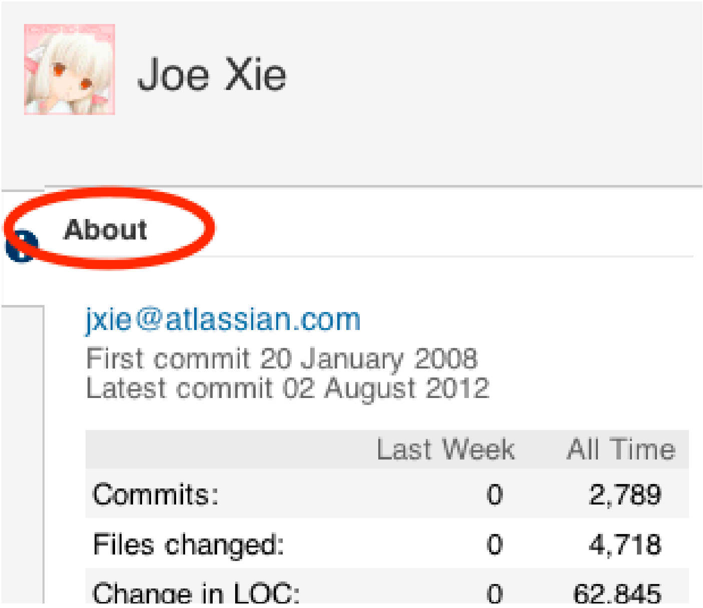
> Some users may not appear to have the correct number of Files Changed or LOC, despite regularly committing. In this situation, if they have committed to a directory which is not covered by the regexes in your symbolic definition (i.e. they have committed to a directory that is neither trunk, branches or tags) then that directory will be counted as part of trunk. Also note that creating tags and branches themselves does not count toward the totals.

Viewing charts of a person's activity

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

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

Screenshot: People Activity Charts in Crucible



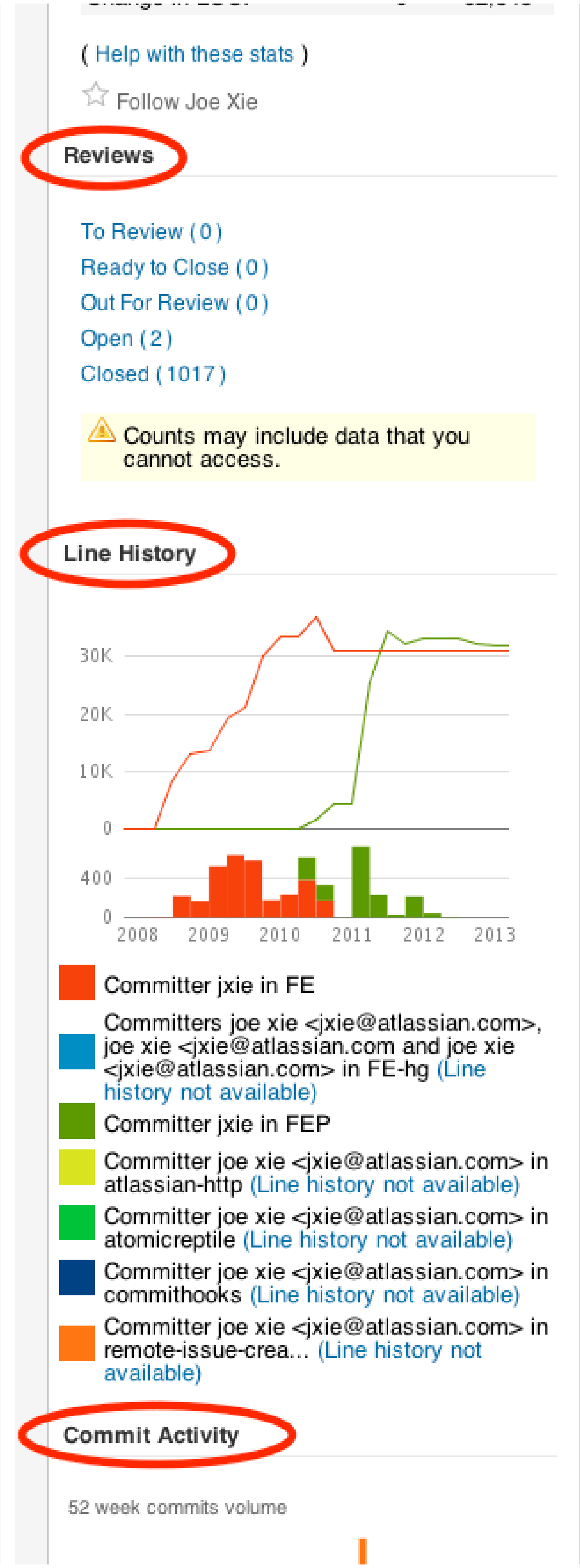
Joe Xie

About

jxie@atlassian.com
First commit 20 January 2008
Latest commit 02 August 2012

	Last Week	All Time
Commits:	0	2,789
Files changed:	0	4,718
Change in LOC:	0	62,845

- **About**
  The username section shows the email address, then the first and latest commit dates for the person in context.
  Also displayed are data points for the previous week and all-time. It shows number of commits, number of files changed and number of lines changed.



(Help with these stats)

☆ Follow Joe Xie

Reviews

To Review (0)
Ready to Close (0)
Out For Review (0)
Open (2)
Closed (1017)

⚠ Counts may include data that you cannot access.

Line History

🟥 Committer jxie in FE

🟦 Committers joe xie <jxie@atlassian.com>, joe xie <jxie@atlassian.com and joe xie <jxie@atlassian.com> in FE-hg (Line history not available)

🟩 Committer jxie in FEP

🟨 Committer joe xie <jxie@atlassian.com> in atlassian-http (Line history not available)

🟩 Committer joe xie <jxie@atlassian.com> in atomicreptile (Line history not available)

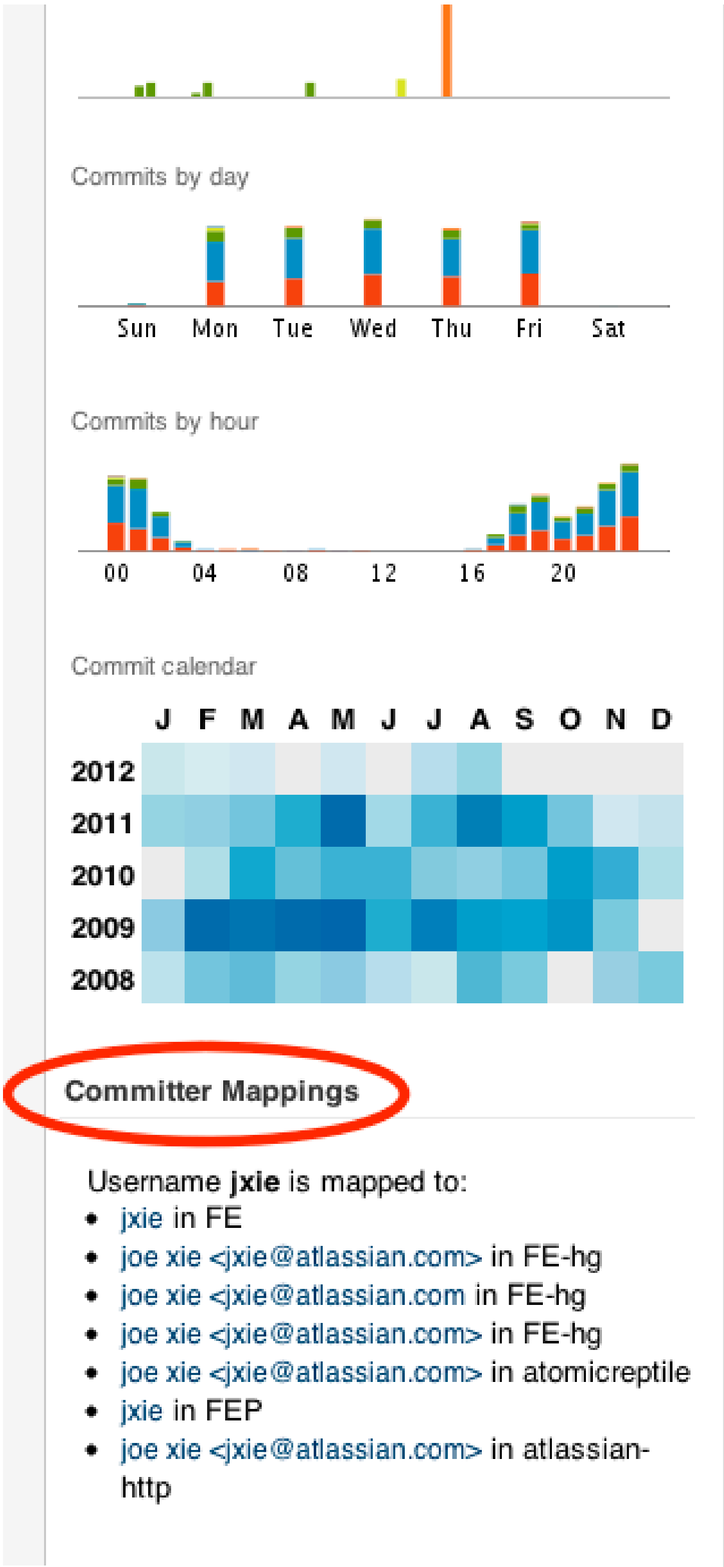
🟦 Committer joe xie <jxie@atlassian.com> in commithooks (Line history not available)

🟧 Committer joe xie <jxie@atlassian.com> in remote-issue-crea... (Line history not available)

Commit Activity

52 week commits volume

- **Reviews**
  The Reviews section shows several filters that you can click to constrain the review items shown in the right-hand pane. The options are **To Review**, **Ready to Close**, **Out For Review**, **Open** and **Closed**.
- **Line History**
  The Line History section shows a graph with the number of lines committed to the repository, charted over time.
- **Commit Activity**
  The Commit Activity section shows four smaller charts; the first showing the volume of commits over a 52 week period; the second showing the relative number of commits on days of the week; the third showing the relative number of commits by the hour of the day when they were lodged; the last shows a commit calendar.
- **Committer Mappings**
  The Committer Mappings section displays username mappings from various systems if they have several usernames in play.



Committer Mappings

Username **jxie** is mapped to:

- jxie in FE
- joe xie <jxie@atlassian.com> in FE-hg
- joe xie <jxie@atlassian.com in FE-hg
- joe xie <jxie@atlassian.com> in FE-hg
- joe xie <jxie@atlassian.com> in atomicreptile
- jxie in FEP
- joe xie <jxie@atlassian.com> in atlassian-http

Viewing Project Statistics

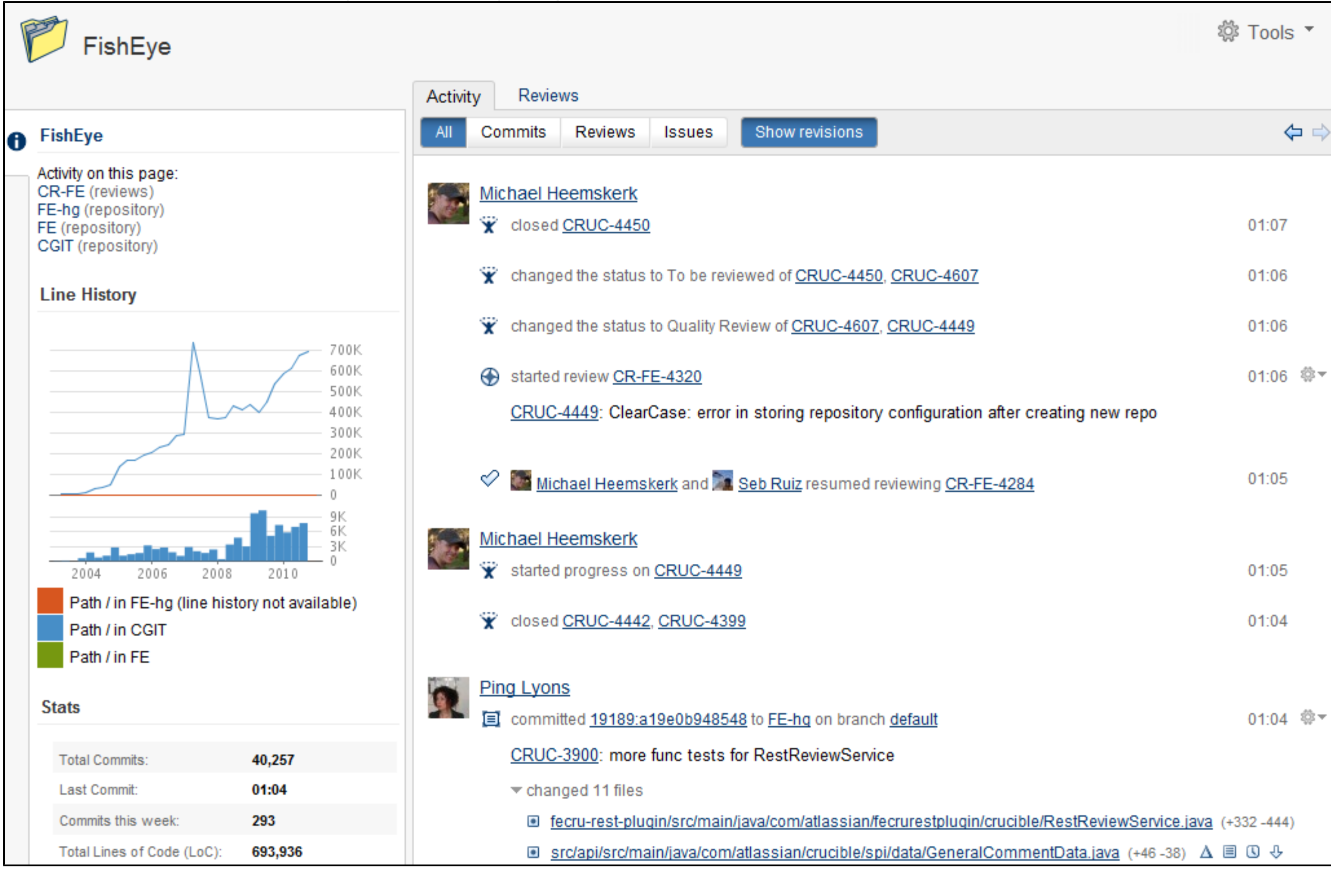
This page explains the layout of the Project Summary page.

On this page:

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

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

Screenshot: The Crucible Project Summary Page



Activity on this page:
CR-FE (reviews)
FE-hg (repository)
FE (repository)
CGIT (repository)

🟥 Path / in FE-hg (line history not available)
🟦 Path / in CGIT
🟩 Path / in FE

Total Commits:	40,257
Last Commit:	01:04
Commits this week:	293
Total Lines of Code (LoC):	693,936

Michael Heemskerk
closed CRUC-4450 01:07
changed the status to To be reviewed of CRUC-4450, CRUC-4607 01:06
changed the status to Quality Review of CRUC-4607, CRUC-4449 01:06
started review CR-FE-4320 01:06
CRUC-4449: ClearCase: error in storing repository configuration after creating new repo

Michael Heemskerk and Seb Ruiz resumed reviewing CR-FE-4284 01:05

Michael Heemskerk
started progress on CRUC-4449 01:05
closed CRUC-4442, CRUC-4399 01:04

Ping Lyons
committed 19189:a19e0b948548 to FE-hg on branch default 01:04
CRUC-3900: more func tests for RestReviewService
changed 11 files
fecru-rest-plugin/src/main/java/com/atlassian/fecrurestplugin/crucible/RestReviewService.java (+332 -444)
src/api/src/main/java/com/atlassian/crucible/spi/data/GeneralCommentData.java (+46 -38)

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

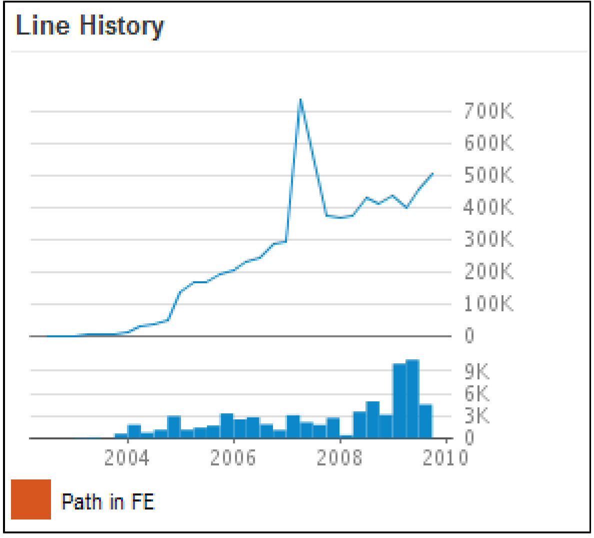
Project Name Panel

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

Project Line History Panel

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

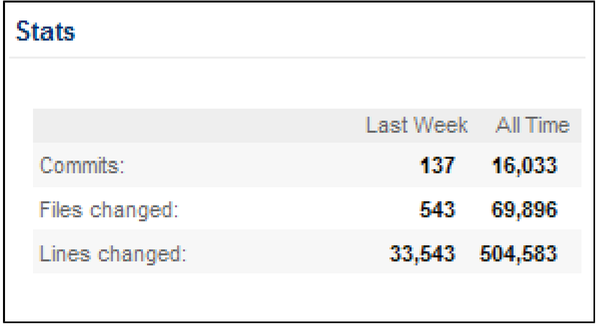
Screenshot: The Project Line History Panel



Project Stats Panel

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

Screenshot: The Project Stats Panel



Stats

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

Project Commit Activity Chart

This panel contains a number of charts:

Chart	Description
.	**52 week commits volume** This chart shows the amount of commits, shown by week over a one year period. **Commits by day** This chart shows the amount of commits, graphed by day over the past week. **Commits by hour** This chart shows the amount of commits, graphed by hours over the past day. **Commit calendar** This chart shows the amount of commits (shown as darker colors to indicate more commits) graphed by month, over years that the repository has been running.

Deleting a review

To delete a review you must first abandon the review. To do that, follow the instructions below.

Deleted reviews cannot be retrieved.

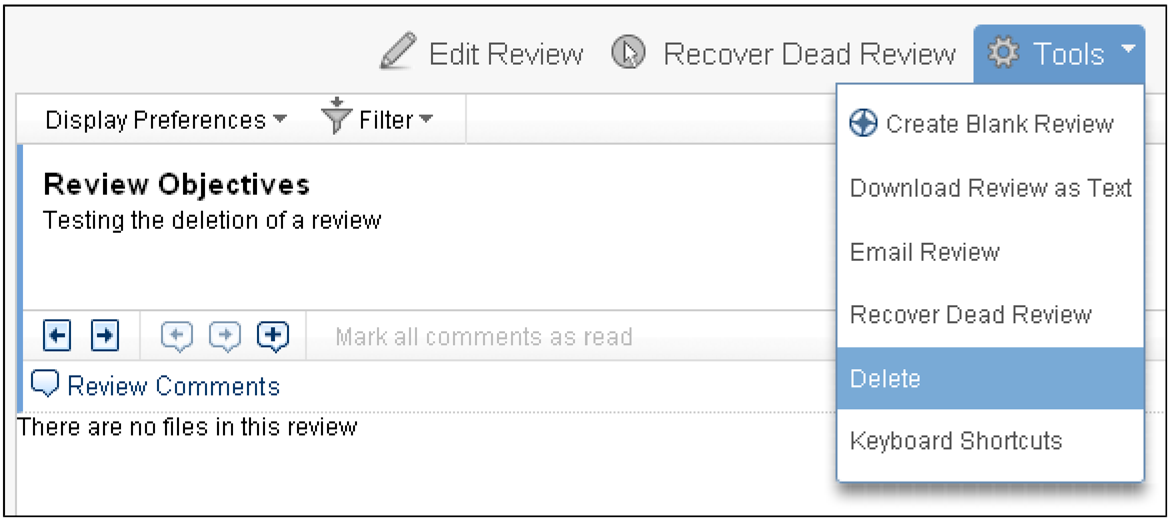
Related page:

- Deleting hung reviews manually

To abandon and then delete a review:

1. Open the review.
2. Choose **Tools** > **Abandon**.
3. Now, on the Crucible dashboard, click **My Abandoned Reviews** in the left-hand navigation bar.
4. In the list of abandoned reviews, click the name of the review you wish to remove.
5. Once the review details are displaying, choose **Tools** > **Delete**. The review will be instantly deleted.

Screenshot: Deleting a review in Crucible



Review Objectives

Testing the deletion of a review

There are no files in this review

Searching Crucible

Crucible has a powerful search engine that allows you to find reviews. There are two methods for searching in Crucible:

- **Quick Search** — The Quick Search allows you search all Crucible projects by entering a single search string. This search is the default search and will suggest "quick nav" results (header search box only). Results are weighted by most recent edit date.
- **Filtering Reviews** — An alternative method for searching for reviews is to display all reviews and apply a custom filter to the list. This is generally slower than searching, but allows you to specify filter criteria against a range of fields.
- **Comment Search** — If you want to find specific review comments, Crucible provides a powerful comment search.

See also Searching FishEye.

On this page:

Using the Quick Search

Before you begin:

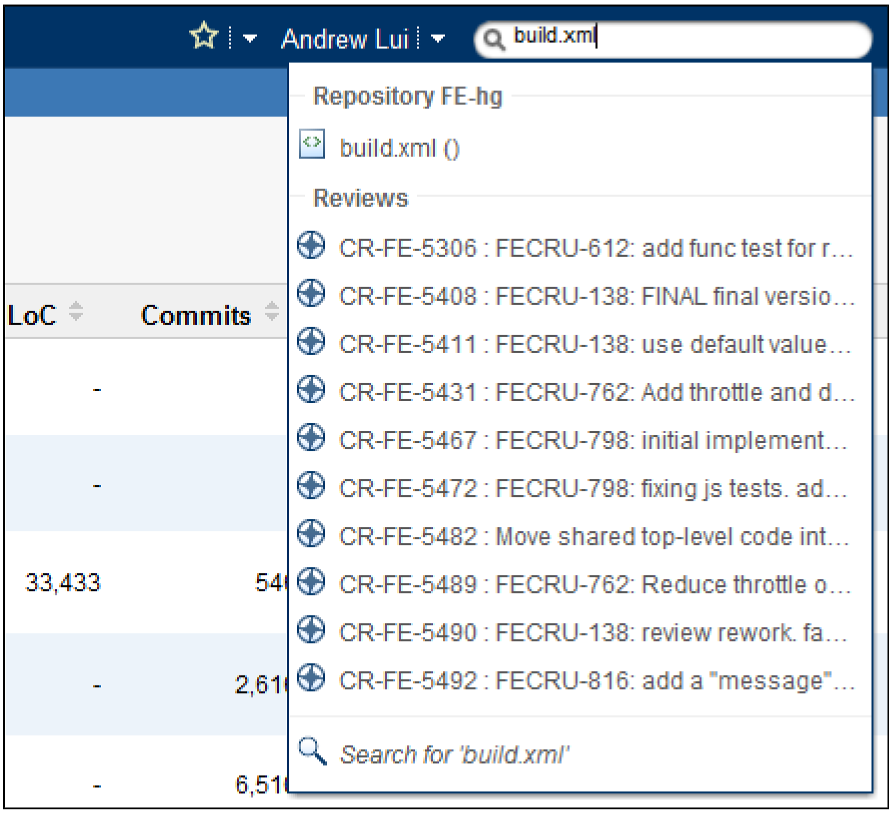
- The Quick Search will also return changesets and files, if you are using **FishEye** with Crucible. For information on searching FishEye, see Searching FishEye in the FishEye documentation.

To search Crucible using the Quick Search:

1. Enter your search terms in the search box in the Crucible header. Crucible offers a number of criteria that you can use to refine your results, see Refining your Quick Search Criteria below.
2. Results will appear in a dropdown, as you type. Results will attempt to be matched against the review name, project and user.
   - If you want to use a quick search result, use the up- and down-arrows on your keyboard and press enter or use your mouse to select the item.
   - If the quick search results don't have what you are looking for, press enter to run a search. Ensure that no items in the dropdown are selected when you press enter.
3. The Quick Search results page will be displayed. You can filter your results further, as described in Filtering Quick Search Results below.
   Results are sorted by relevance and boosted if they were edited recently. A maximum of 10 results are displayed per page.
   - If you have **integrated your Crucible instance with a JIRA instance**, you can display a summary of any JIRA issues referenced in your search results by hovering over the issue key. For more details, see JIRA integration in Crucible.
4. If you want to run another search, enter your new criteria in the main search box or in the search box in the header.

Note, only the search box in the header provides "quick nav" results.

Screenshot: Quick Search displaying "quick nav" matches


Andrew Lui
build.xml
Repository FE-hg
build.xml ()
Reviews
CR-FE-5306 : FECRU-612: add func test for r...
CR-FE-5408 : FECRU-138: FINAL final versio...
CR-FE-5411 : FECRU-138: use default value...
CR-FE-5431 : FECRU-762: Add throttle and d...
CR-FE-5467 : FECRU-798: initial implement...
CR-FE-5472 : FECRU-798: fixing js tests. ad...
CR-FE-5482 : Move shared top-level code int...
CR-FE-5489 : FECRU-762: Reduce throttle o...
CR-FE-5490 : FECRU-138: review rework. fa...
CR-FE-5492 : FECRU-816: add a "message"...
Search for 'build.xml'
LoC
Commits
-
-
33,433
54
-
2,61
-
6,51

Refining your Quick Search Criteria

You can refine your search criteria before executing the search:

Search Tool	Description	Example
Field Handles	Use a field handle in your criteria to restrict your search to a particular field. Note, you cannot have multiple field handles in a query. • `detail` — Search against title, objective, key, linked reviews and linked issues. • `reviewcomment` — Search against review comments.	
Searching for Discrete Strings	Enter a specific string within quotation marks and Crucible will match against the exact string. Note, this search is not case-sensitive.	Enter "`CR-2818`" and Crucible will only return results that match that exact string, i.e. it will not return a result with `CR-FE-2818` or `CR-28189`.

Filtering Quick Search Results

You can filter Quick Search results using the controls in the left panel of the Quick Search page:

Filter	Description
All repositories	Type a repository name into the field, or click the down arrow to seea list of repositories.

Source type	Click **Files and directories**, **Commit messages**, **Diffs**, **Content** or **Committers** to restrict the search results to just that source type.
All projects	Select or enter the name of the project that you want to restrict your results to. For example, if you enter 'CR' then the search results page will refresh to display only reviews in the 'CR' project. ⚠ *If you are using Fisheye with Crucible, there will be a repositories dropdown in the 'Source' section. Selecting a FishEye repository in this dropdown will not filter the Crucible search results. It is only used to filter files and changesets returned in the search results. See Searching FishEye.*
Reviews	Click this link to restrict your results to reviews that have a title, objective, key, linked reviews or linked issues that match the search criteria.
Comments	Click this link to restrict your results to reviews that have comments that match the search criteria.
Last modified	Filter by the date of the last change.
By	Filter by author name.

Filtering reviews

Crucible allows you to view all the reviews/snippets that you are involved with, as well as everybody's reviews/snippets. You can filter these lists to find reviews.

To filter a list of reviews:

1. Click **Reviews** in the header.
2. Click the list of reviews that you want to start with, in the sidebar of the 'Review Dashboard', e.g. 'All Open Reviews'.
3. Click **Custom Filter** in the reviews sidebar.
4. Update the filters with your search criteria (see table below) and then click **Apply Filter**.

Filter	Description
Title	Find reviews by searching for words within the title.
Project	Find reviews under a particular project.
Author	Find reviews moderated by a particular authors.
Moderator	Find reviews moderated by a particular moderators.
Creator	Find reviews created by a particular creator.
Reviewer	Find reviews that are reviewed by a particular reviewer. This will default to the user logged in.
Reviewer Status	This is reliant on the above filter and is used to show reviews that have either been completed by the reviewer, not completed or all reviews.
Type	Choose either **Review** or **Snippet**.
Match Roles	To use all the above filters, choose **all**. To use any of the filters, choose **any.**
Review state checkboxes	Check any of the review state checkboxes (e.g. **Draft**, **Pending Approval**) to filter for reviews in those states.

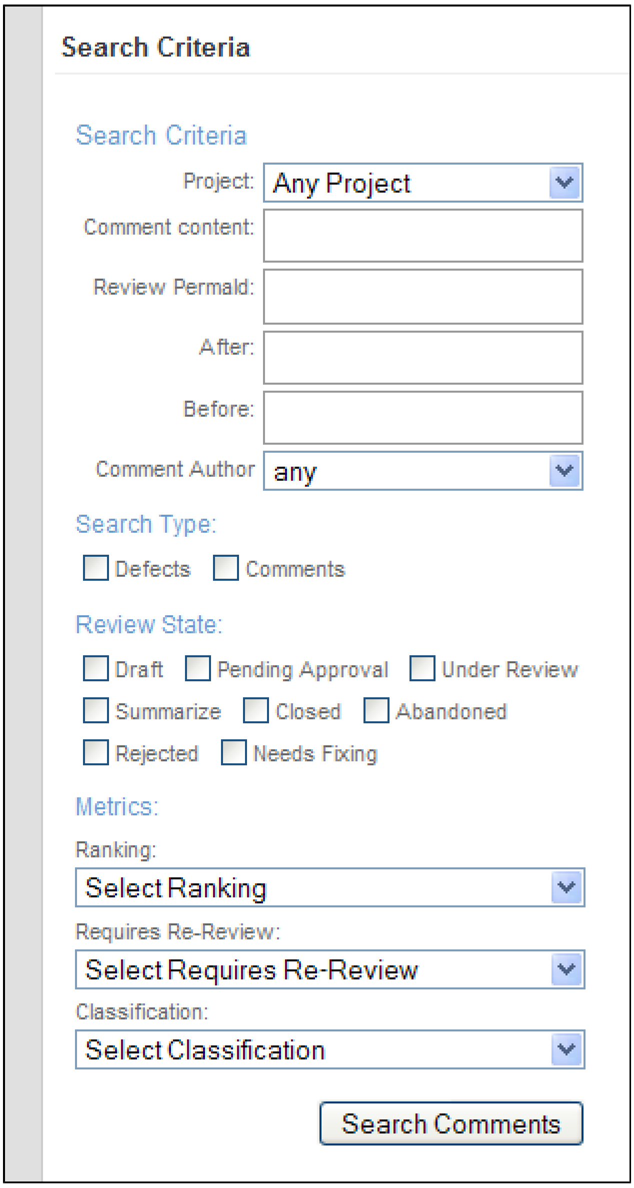
Searching for Review Comments

To search for review comments:

1. Click **Reviews** in the header.
2. Enter your search string in the 'Comment Search' section at the bottom of the reviews sidebar.
3. Click **Search Comments.**
4. The 'Comment Search' page will display your results. You can refine your search using the search criteria on the page:

Project	Find comments on reviews under a particular project.
Comment content	Find comments that contain the specified text.
Review PermaId	Find comments made on the specified review.
After	Find comments made after a particular date.
Before	Find comments made before a particular date.
Comment Author	Find comments made by a particular user.
Search Type	Filter for comments marked as **Defects**. Check **Comments** to find comments that are not flagged as Defects.
Review State	Find comments on reviews that are in a particular state. See *Review State Filter* (above).
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').

Screenshot: Search Comment Filter Options



Search Criteria

Search Criteria

Project: Any Project

Comment content:

Review Permald:

After:

Before:

Comment Author any

Search Type:

□ Defects □ Comments

Review State:

□ Draft □ Pending Approval □ Under Review

□ Summarize □ Closed □ Abandoned

□ Rejected □ Needs Fixing

Metrics:

Ranking:

Select Ranking

Requires Re-Review:

Select Requires Re-Review

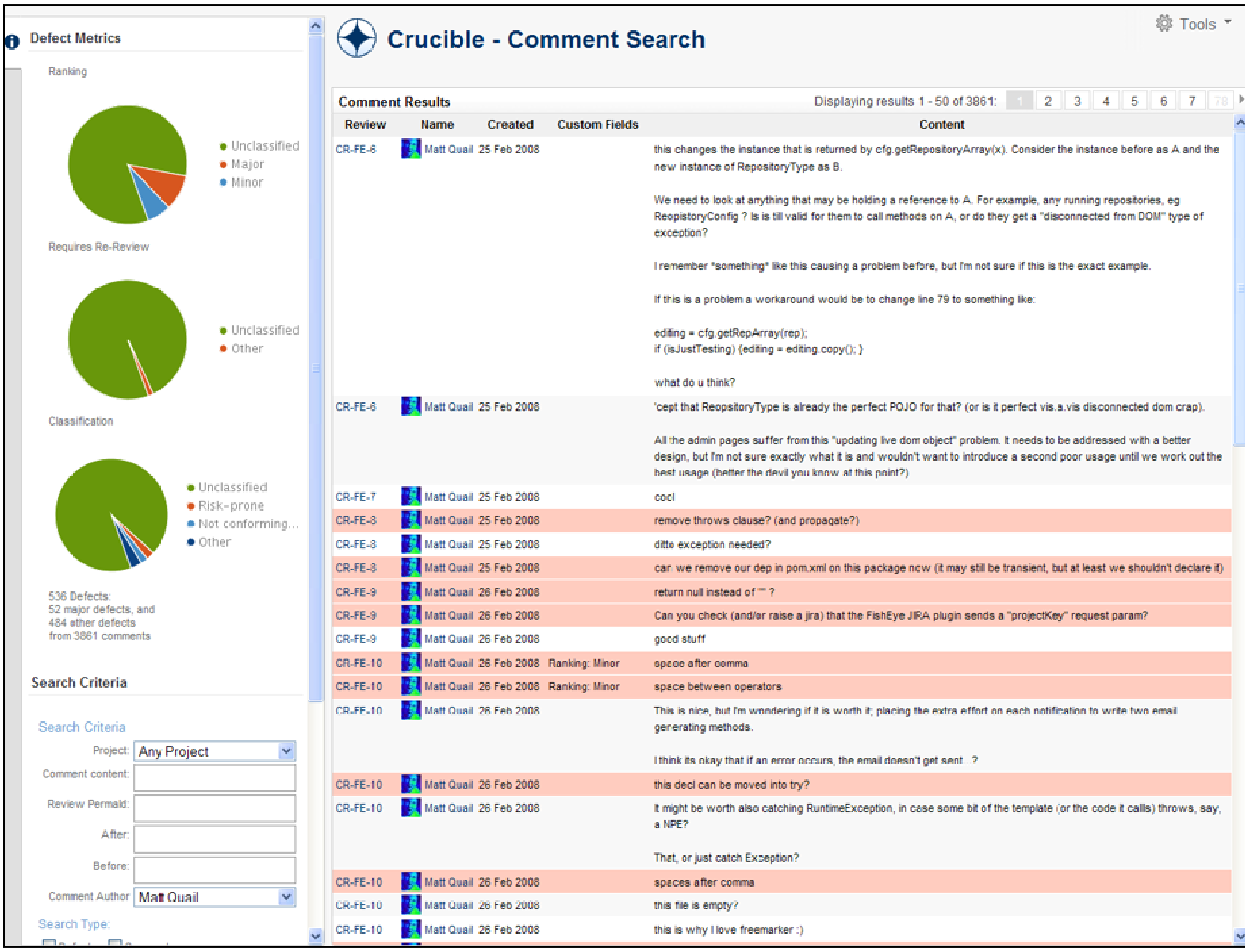
Classification:

Select Classification

Displaying Defect Metric Charts for Comment Search Results

Once you have retrieved results for a review comment search, you can click **Defect Metrics** in the left navigation pane to display defect classification charts.

Screenshot: Comment Search Results



536 Defects:
52 major defects, and
484 other defects
from 3861 comments

Search Criteria

Search Criteria

Project:	Any Project
Comment content:	
Review Permaid:	
After:	
Before:	
Comment Author	Matt Quail

Search Type:

Crucible - Comment Search

Comment Results

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

JIRA integration in Crucible

When Crucible is integrated with JIRA Software, you and your team get all the benefits described on this page:

In Crucible, you can:

- See all the Crucible reviews related to a JIRA Software issue
- Create a Crucible review directly from an issue in JIRA Software
- Link your Crucible review to a JIRA Software issue
- Create a JIRA Software issue from a review comment
- Transition JIRA Software issues automatically
- Transition JIRA Software issues from within Crucible
- See issues from multiple instances of JIRA Software
- See open reviews or unreviewed commits for an entire version within JIRA Software

Related pages:

- Creating a review from JIRA
- Creating JIRA issues from the review
- Transitioning JIRA issues
- Linking Crucible to JIRA

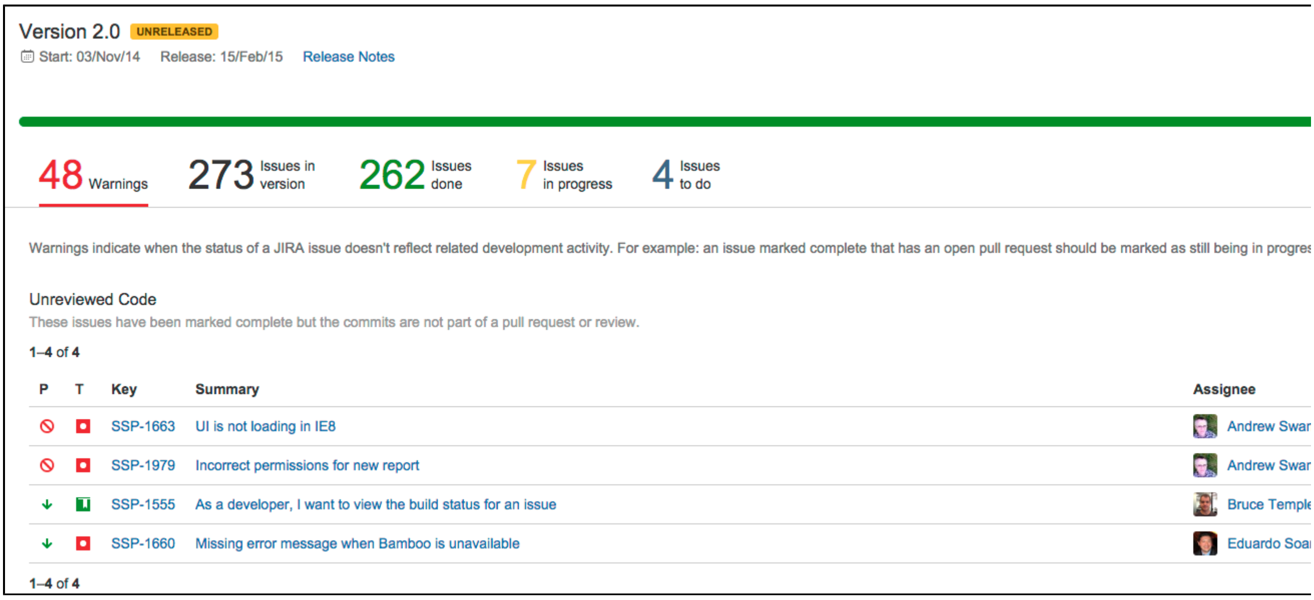
Note that your Crucible and JIRA Software instances must be linked to make use of these JIRA Software integration features. See Linking Crucible to JIRA.

Check development progress of a version in JIRA Software

CRUCIBLE 3.3+

JIRA 6.4+

The Release Hub in JIRA Software shows the relevant issues and development information for a version – so you can determine which issues are likely to ship at a glance. With JIRA Software and Bitbucket Server connected, the release page can warn you about open reviews or unreviewed commits that could cause problems for your release.



Version 2.0 UNRELEASED

Start: 03/Nov/14 Release: 15/Feb/15 Release Notes

48 Warnings 273 Issues in version 262 Issues done 7 Issues in progress 4 Issues to do

Warnings indicate when the status of a JIRA issue doesn't reflect related development activity. For example: an issue marked complete that has an open pull request should be marked as still being in progres

Unreviewed Code

These issues have been marked complete but the commits are not part of a pull request or review.

1–4 of 4

P	T	Key	Summary	Assignee
⊘	🟥	SSP-1663	UI is not loading in IE8	Andrew Swar
⊘	🟥	SSP-1979	Incorrect permissions for new report	Andrew Swar
↓	🟩	SSP-1555	As a developer, I want to view the build status for an issue	Bruce Temple
↓	🟥	SSP-1660	Missing error message when Bamboo is unavailable	Eduardo Soai

1–4 of 4

From the Release Hub you can also:

- Release a version
- Mark a version as complete
- Move incomplete issues to other versions
- Trigger release builds (if JIRA Software is connected to Bamboo)
- Warnings that help you reconcile what is happening in development with JIRA data.

To view the Release Hub (with the project sidebar enabled), navigate to a project, click on **Releases**, then select a version listed. See Checking the progress of a version more detailed information about using the Release Hub in JIRA Software.

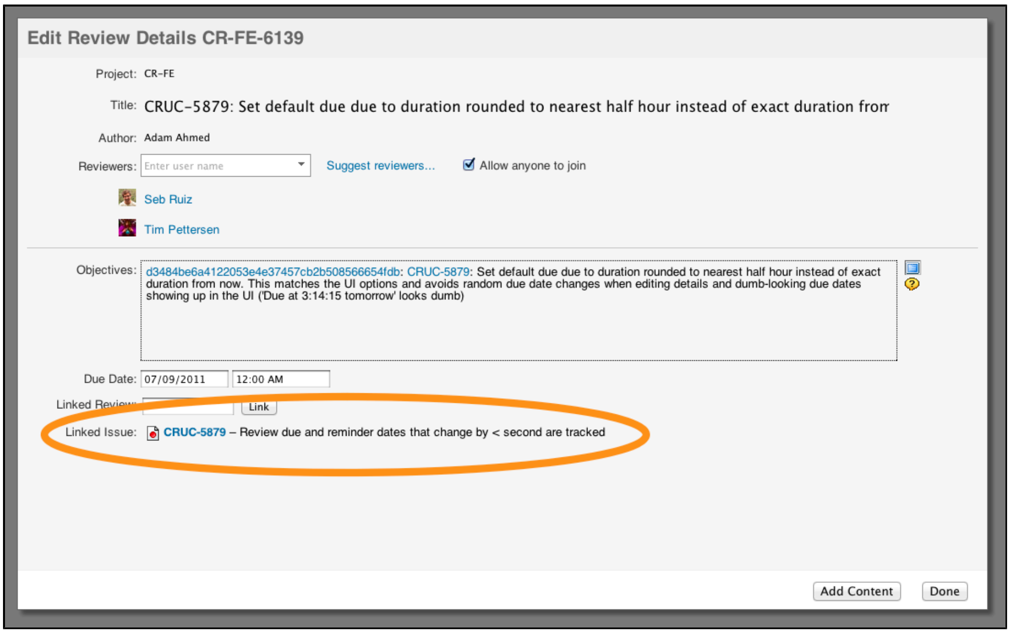
See all the Crucible reviews related to a JIRA Software issue

In a JIRA Software issue, the Development panel shows the number of reviews that are linked to the issue. Click the **reviews** link to see details of those reviews.

Link your Crucible review to a JIRA Software issue

When creating, or editing, your review, Crucible will suggest a JIRA Software issue that can be linked to the review, if a JIRA Software issue key is found in the review title. You can:

- click the suggested JIRA Software issue key, to link it to the review
- delete the suggested JIRA Software issue and specify a different issue key and click **Link** to save it.



Edit Review Details CR-FE-6139

Project: CR-FE

Title: CRUC-5879: Set default due due to duration rounded to nearest half hour instead of exact duration fron

Author: Adam Ahmed

Reviewers: Enter user name

☑ Allow anyone to join

Seb Ruiz

Tim Pettersen

Objectives: d3484be6a4122053e4e37457cb2b508566654fdb: CRUC-5879: Set default due due to duration rounded to nearest half hour instead of exact duration from now. This matches the UI options and avoids random due date changes when editing details and dumb-looking due dates showing up in the UI ('Due at 3:14:15 tomorrow' looks dumb)

Due Date: 07/09/2011 12:00 AM

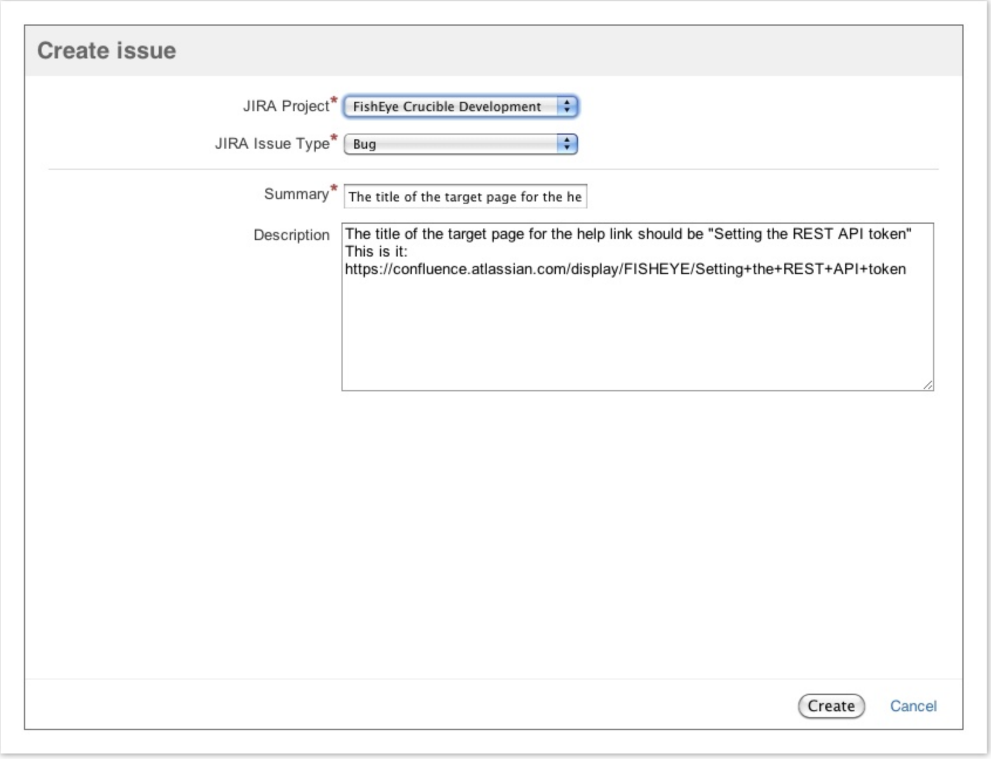
Linked Review:

Linked Issue: CRUC-5879 – Review due and reminder dates that change by < second are tracked

Create a JIRA Software issue from a review comment

When viewing any review comment (genaral, file, inline), you can click **Create Issue** in the comment to create a JIRA Software issue. Crucible suggests the JIRA Software instance, project and issue type, but you can modify these. This requires JIRA 5.0, or later, and is disabled if Crucible is integrated with an earlier version of JIRA Software.

See Creating JIRA issues from the review for more details.



Create issue

JIRA Project* FishEye Crucible Development

JIRA Issue Type* Bug

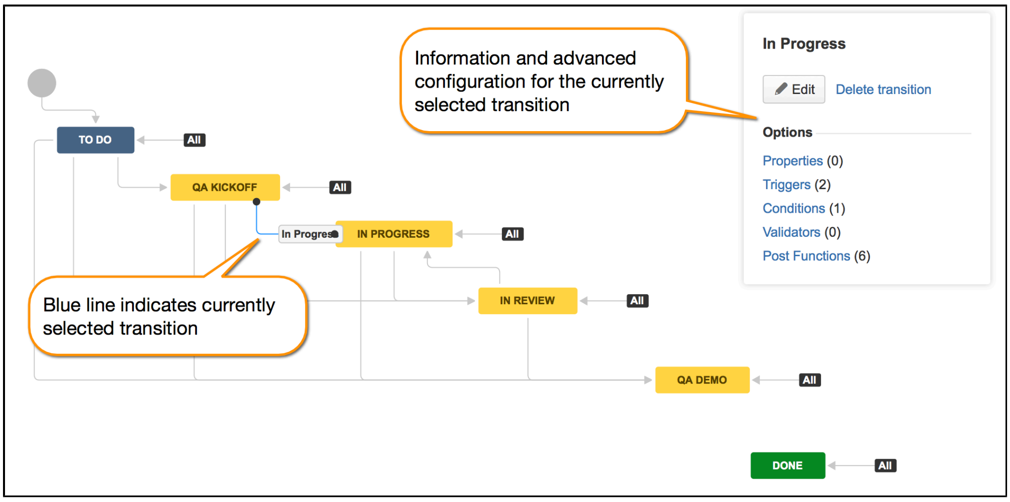
Summary* The title of the target page for the he

Description The title of the target page for the help link should be "Setting the REST API token"
This is it:
https://confluence.atlassian.com/display/FISHEYE/Setting+the+REST+API+token

Transition JIRA Software issues automatically

Your JIRA Software workflow can now respond to events in your linked development tools, for when a review is started, your JIRA Software workflow can be configured to automatically transition the related issue.

Configure this from transitions within the JIRA Software workflow editor. (Available with JIRA 6.3.3 and later.)

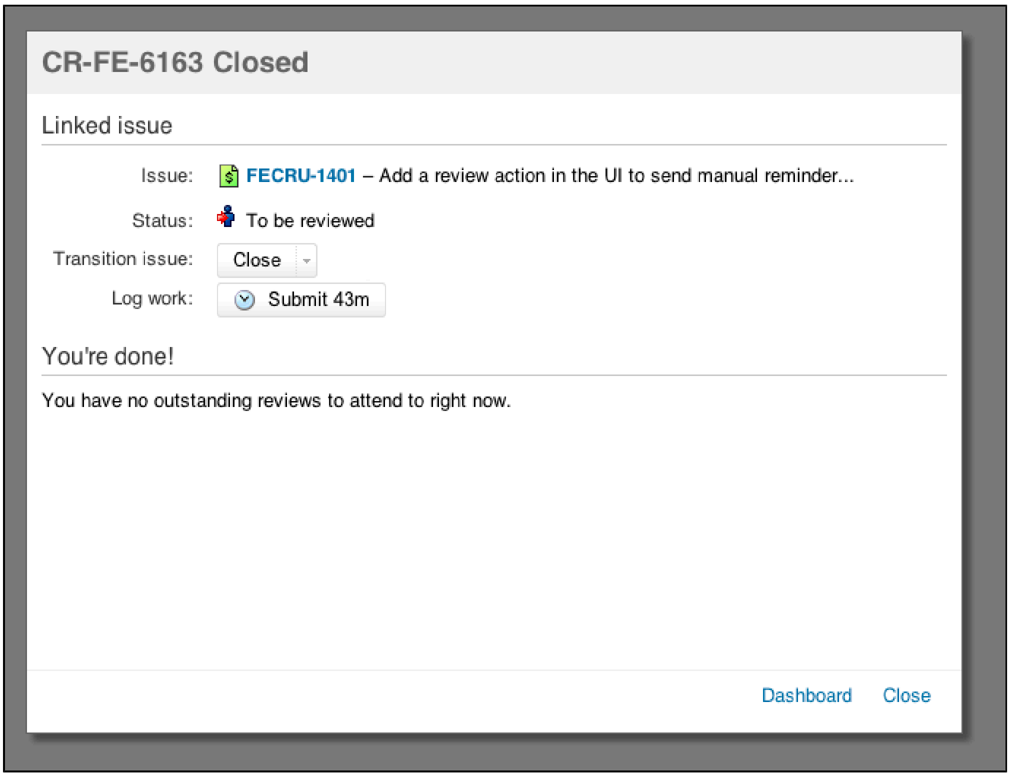


The events available in Crucible are:

- Review started
- Submitted for approval
- Review rejected
- Review abandoned
- Review closed
- Review summarized

Transition JIRA Software issues from within Crucible manually

For Crucible reviews that have linked JIRA Software issues, you can advance the JIRA Software workflow for the issue from within Crucible. You can do this at any time by clicking the linked issue, or when you close the review:



CR-FE-6163 Closed

Linked issue

Issue: FECRU-1401 – Add a review action in the UI to send manual reminder...

Status: To be reviewed

Transition issue: Close

Log work: Submit 43m

You're done!

You have no outstanding reviews to attend to right now.

See Transitioning JIRA issues for more details.

See issues from multiple instances of JIRA Software

Crucible can link to more than one JIRA Software server at a time, so different teams can work with their own projects in different JIRA Software instances, or a single team can link to issues across multiple JIRA Software servers.

Creating JIRA issues from the review

From any review comment (general, file, inline) in Crucible, you can create a JIRA Software issue directly from the comment. This requires that Crucible is integrated with JIRA, version 5.0 or later, and is disabled if you have an earlier version of JIRA Software.

Inline issue creation allows:

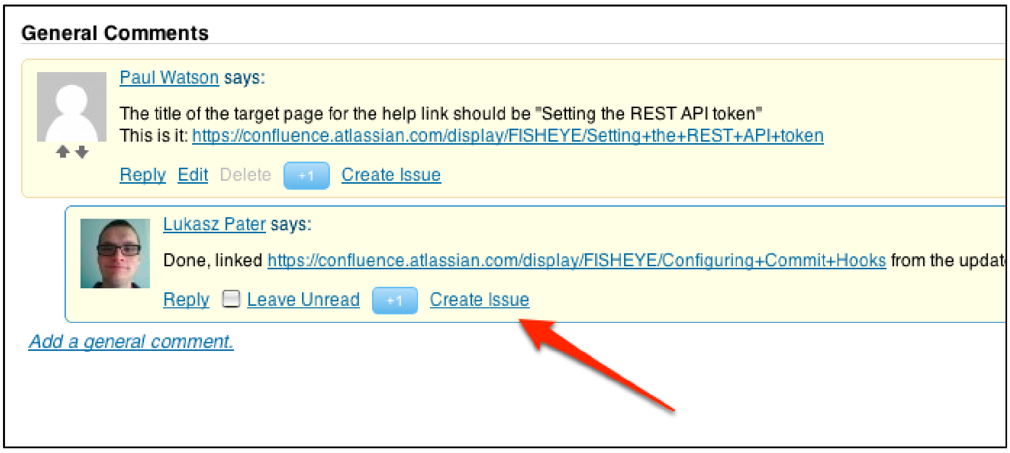
- Tracking of the status of the comment
- A faster way to pull out incidental suggestions raised in reviews as JIRA Software issues
- A quick link back to the comment from the JIRA Software issue, using Remote Issue Links.

You might find this useful when:

1. Tracking the status of a review:
   a. The 'Issues Raised from Comments' section in the review shows the open/closed status of related issues.
   b. Raising related issues enforces dealing with subtasks before the review can be closed.
2. Closing off a review:
   a. You can create JIRA Software issues, unrelated to the current review, to track matters to be dealt with later.

Creating a JIRA Software issue

To create a JIRA Software issue from a review, click **Create Issue** in an existing comment. Note that you need the 'Comment' permission in Crucible to see the **Create Issue** link.



General Comments

Paul Watson says:

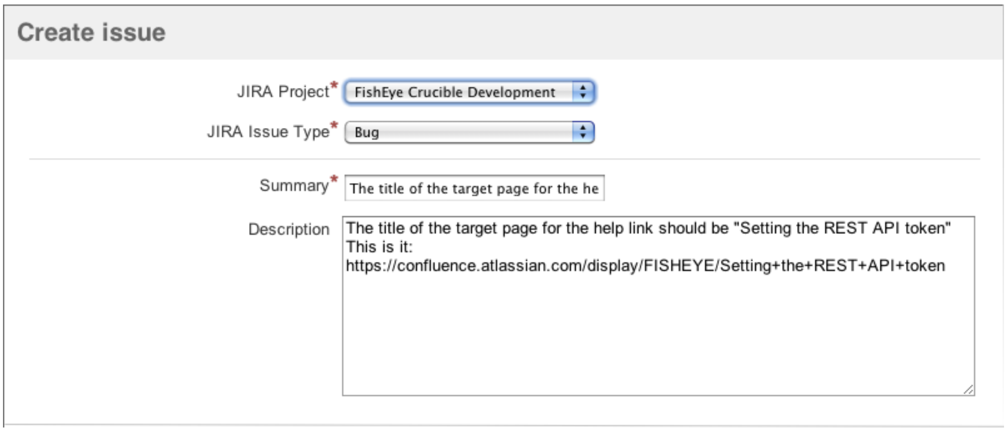
The title of the target page for the help link should be "Setting the REST API token"
This is it: https://confluence.atlassian.com/display/FISHEYE/Setting+the+REST+API+token

Lukasz Pater says:

Done, linked https://confluence.atlassian.com/display/FISHEYE/Configuring+Commit+Hooks from the updat

Crucible suggests a JIRA Software instance, project and issue type, but you can choose from the available options. You can choose **Sub-task** from the **JIRA Issue Type** list if a JIRA issue is already linked to the review.

Crucible only displays required fields for the issue type; these can be configured in JIRA Software by your administrator.



Create issue

JIRA Project* FishEye Crucible Development

JIRA Issue Type* Bug

Summary* The title of the target page for the he

Description The title of the target page for the help link should be "Setting the REST API token"
This is it:
https://confluence.atlassian.com/display/FISHEYE/Setting+the+REST+API+token

Once the issue is created. the comment displays a link to the issue in JIRA Software, and in JIRA Software, the issue displays a link back to the comment in Crucible. The 'Issues raised from comments' section of the review displays links to the JIRA Software issues.

See also Creating a review from JIRA.

Transitioning JIRA issues

When Crucible is linked to JIRA Software, you can advance the workflow for a JIRA Software issue directly from within Crucible.

You can transition a JIRA Software issue in two ways:

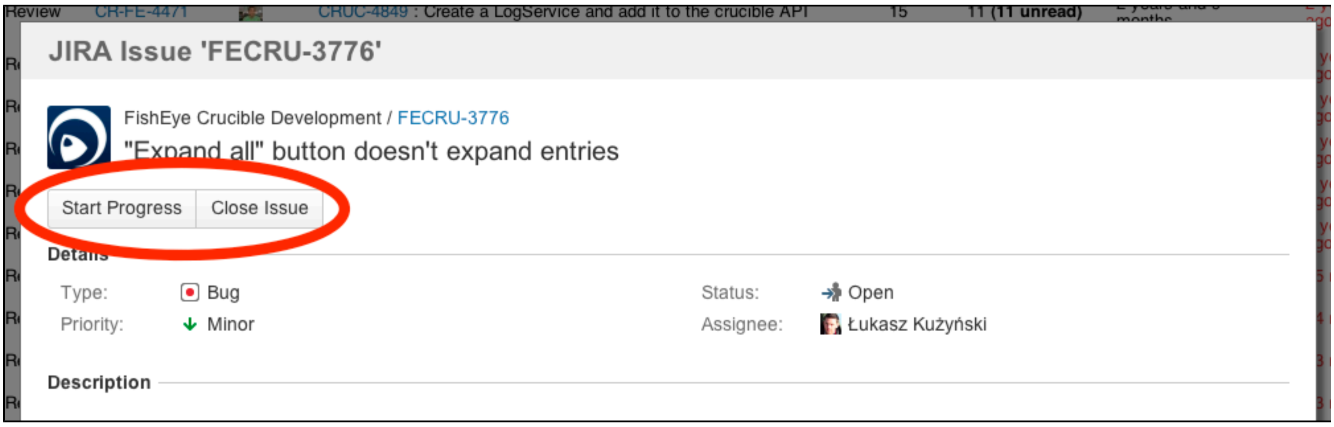
- Transitioning any JIRA Software issue at any time
- Transitioning a linked JIRA Software issue when closing the review

Related pages:

- JIRA integration in Crucible
- Creating JIRA issues from the review
- Linking Crucible to JIRA

Transitioning any JIRA Software issue at any time

You can easily transition a JIRA Software issue at any time from within Crucible. Click on a JIRA Software issue link anywhere in Crucible to see a dialog with the available workflow steps:



JIRA Issue 'FECRU-3776'

FishEye Crucible Development / FECRU-3776

"Expand all" button doesn't expand entries

Details

Type: Bug

Priority: Minor

Status: Open

Assignee: Łukasz Kużyński

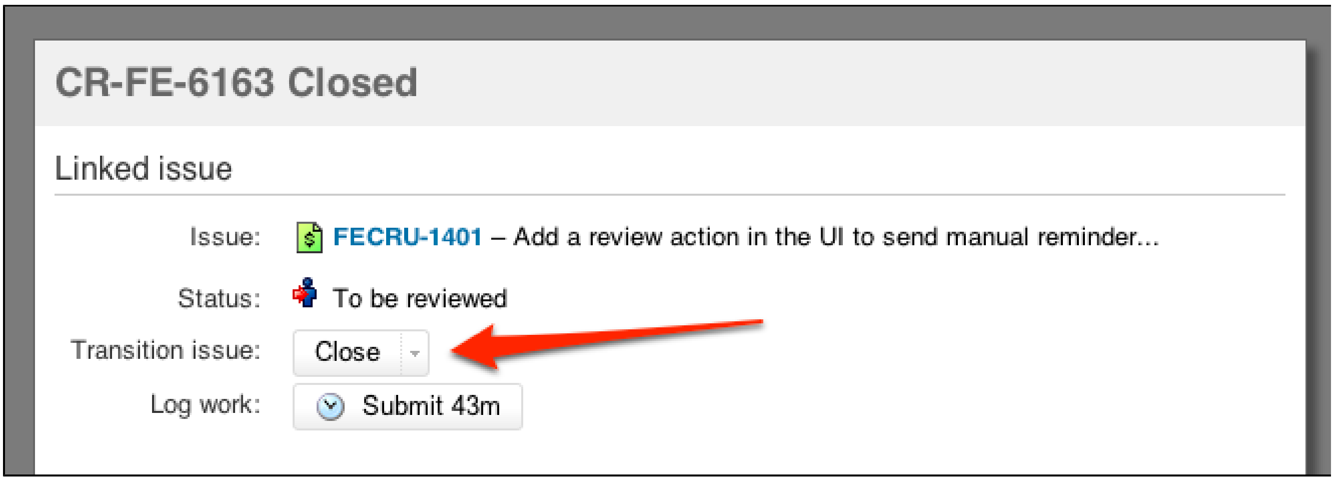
Description

Click on a step in the dialog, and complete any displayed fields as required. If there are custom required fields that are unsupported by Crucible, just click **Edit this field in JIRA** to transition the issue directly in JIRA Software.

Transitioning a linked JIRA Software issue when closing the review

When closing a Crucible review you may also want to close a JIRA Software issue that is linked to that review.

In the Review Summary screen, click **Close** near the top right. In the 'Closed' dialog, the available workflow transitions for the linked JIRA Software issue appear in the **Transition issue** dropdown:



CR-FE-6163 Closed

Linked issue

Issue: FECRU-1401 – Add a review action in the UI to send manual reminder...

Status: To be reviewed

Transition issue: Close

Log work: Submit 43m

Choose a step from the dropdown, and click **Close**.

Notes

- Only the transitions accessible by the user are displayed.
- The list of available transitions only appears if the user has visibility to any available workflow transitions.
- Crucible administrators can turn off JIRA Software issue transitioning by disabling the Crucible Issue Transitioning Plugin. See Managing add-ons.

ⓘ This feature does not support editing fields on the issue, only setting the *resolution* field if required by the transition.

Crucible FAQ

Crucible FAQ

Answers to frequently asked questions about configuring and using Crucible.

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

- FishEye documentation
- FishEye FAQs

- Top Evaluator Questions
  - Can Crucible add support for new repositories?
  - Can I purchase Crucible on it's own?
  - Can I trial Crucible without FishEye?
  - How can I do reviews from the file system?
  - How does Crucible help enforce compliance and auditability?
  - How do I convince my team of the benefits of code review?
  - How do I do pre-commit reviews?
  - How do I raise defects in JIRA?
  - How do I review patch diffs?
  - What user permissions and review security is available?

Do you still have a question, or need help with Crucible? Please create a support request.

Crucible Resources

Resources for Evaluators

- Free Trial
- Feature Tour

Resources for Administrators

- Crucible Knowledge Base
- Crucible FAQ
- Guide to Installing an Atlassian Integrated Suite
- The big list of Atlassian gadgets

Downloadable Documentation

- Crucible documentation in PDF, HTML or XML formats

Plugins

- Crucible Developer Documentation
- Add-ons for Crucible

Support

- Atlassian Support
- Support Policies

Forums

- Crucible Forum
- Crucible Developers Forum

Mailing Lists

- Visit http://my.atlassian.com to sign up for mailing lists relating to Atlassian products, such as technical alerts, product announcements and developer updates.

Feature Requests

- Issue Tracker and Feature Requests for Crucible

General FAQs

Crucible General FAQs

- Can I deploy Crucible or FishEye as a WAR? — Unfortunately FishEye and Crucible cannot be deployed as a WAR.
- How do I force reviews to include SVN property changes?
- How to Automate Daily Crucible Backups — Configuring Crucible backups is easy.

Can I deploy Crucible or FishEye as a WAR?

Unfortunately FishEye and Crucible cannot be deployed as a WAR. FishEye has some special needs for performance reasons that are not easily supported on third-party containers. While this is an often requested feature, there are no immediate plans to provide a WAR version of FishEye or FishEye+Crucible. However the upcoming separate edition of Crucible (i.e. without FishEye) may at some stage be available as a WAR.

How do I force reviews to include SVN property changes?

Subversion (SVN) allows you to store arbitrary name/value pairs, called 'SVN properties', in association with files and directories. You can use these properties to store metadata, and Subversion also uses them, for example to store where code is branched from.

These name/value pairs can only be changed as part of a changeset or commit in Subversion. As such, you will have changesets with purely changes to SVN properties, or changesets with a mix of textual changes and changes to SVN properties.

Crucible 2.6 introduced review creation without metadata to prevent SVN properties from being included in reviews. This functionality was on by default.

For Crucible 3.0 and later, this functionality is disabled by default, and affects the following actions:

- Creating a review — The **Create Review** link is disabled in the activity stream, dashboard and changeset page for any changeset where all the file revisions only differ by SVN properties.
- Adding content to a review — SVN property-only changesets cannot be added to the review from the 'Add Content' dialog. Changesets with a mix of SVN property changes and textual changes can be added to the review, however only the file revisions with textual changes are added. File revisions with both textual changes and SVN property changes are always added.
  Note that you can explicitly add a particular file revision to a review regardless of whether it is a textual change or metadata-only change.

The prime motivation behind metadata checking is to prevent the creation of reviews with a large number of files which have no effective content changes. Sometimes this sort of problem is seen when there is some wholesale SVN property change, for example setting the line endings on all files (`svn:eol-style`). In that respect, this is a performance setting as it avoids the creation of large reviews. The properties themselves are still stored and managed in FishEye. There is no major performance impact on including properties in reviews.

Enabling this functionality can be done by starting up Crucible with the following system property:

```
-Dcrucible.detect.metadata.revision.changes=true
```

Crucible 3.0+ Changes

In Crucible 3.0 and later the default value of this setting was changed from `true` to `false`. Due to the phased nature of FishEye indexing, changesets are available for review much earlier in their overall processing. In particular changesets are available for review prior to having their line count and diff information processed. In this state all changes look somewhat like metadata only changes. Reviews created in this state with metadata detection enabled will cause the changes to be excluded from the review.

How to Automate Daily Crucible Backups

Configuring Crucible backups is easy.To set daily Crucible backups, open the administration page, click **Backup** (under 'System' on the left), and simply follow the instructions set out on the Backing up and restoring Crucible data page.

Licensing FAQ

Crucible Licensing FAQ

- What happens if I decide to stop using FishEye with Crucible — Crucible can be run as an application alone, without FishEye. However, if you decide to stop using FishEye with Crucible, you will lose certain functionality and will need to make configuration changes.
- Do I need a FishEye license to run Crucible? — FishEye and Crucible are separate products. They can be run separately, and they can also be run together.
- Updating your Crucible license

What happens if I decide to stop using FishEye with Crucible

Crucible can be run as an application alone, without FishEye. However, if you decide to stop using FishEye with Crucible, you will lose certain functionality and will need to make configuration changes.

On this page:

How do I run Crucible without FishEye?

- **Have a valid Crucible license but not a FishEye license**
  To run Crucible without FishEye you need to have a **valid Crucible license but not a FishEye license** or if you want to disable FishEye enter `Disabled` in the license field. Crucible will actually use a "light" mode of FishEye that comes pre-bundled and does not need to be installed separately. For more information on Crucible with light FishEye, see How is Crucible without FishEye different from using Crucible with FishEye? below.

- **No need to reconnect your repositories**
  Any repositories that you have currently defined in FishEye will not be visible in Crucible after removing FishEye (no more `Source` tab). The repositories however, will continue to update as usual without any intervention. You can add additional repositories as described in the FishEye documentation. Note, all repositories supported in FishEye are supported in light FishEye.
  ℹ *Legacy "lightSCM" plugins, like the Crucible Subversion SCM plugin, will still work. However, the functionality will be limited compared to using Crucible with light FishEye. See the Can I still use lightSCM plugins with Crucible? section below for more information.*

- **You need to reindex your repositories after removing a FishEye license**
  When you remove a FishEye license, and operate with only a Crucible license, you need to reindex those repositories that were originally indexed under a FishEye license. You do not need to remove those repositories, you just need to run a re-index to create an index which is compatible with your new license setup.

How is Crucible without FishEye different from using Crucible with FishEye?

The following changes in functionality will occur if you use Crucible without FishEye (i.e. use Crucible with "light" FishEye).

Conducting Reviews

- When using Iterative reviews in Crucible, you will not be prompted when a new version of a file is available.

Viewing Repositories/Files

- Files and changesets displayed in activity streams (e.g. the dashboard activity stream) will not render as links to the relevant files/changesets.
- You will not be able to see your content roots and repositories associated with projects.
- You will no longer be able to see repository lists and browse repositories using the '**Source**' tab.

Charts

- You will not be able to view charts or code metrics.

Do I need a FishEye license to run Crucible?

FishEye and Crucible are separate products. They can be run separately, and they can also be run together.

We recommend that you run Crucible together with FishEye. If you choose to run Crucible alone without FishEye, you will have access to your repositories via the "light" FishEye implementation bundled with Crucible. However, a number of FishEye's advanced features will not be available to you, including pre-caching repository content (for improved performance), the ability to search and browse through repositories and FishEye's activity graphs.

For more information, please read the following FAQ: What happens if I decide to stop using FishEye with Crucible

Updating your Crucible license

When you upgrade or renew your Crucible license, you will receive a new license key – you'll need to update your Crucible server with the new license.

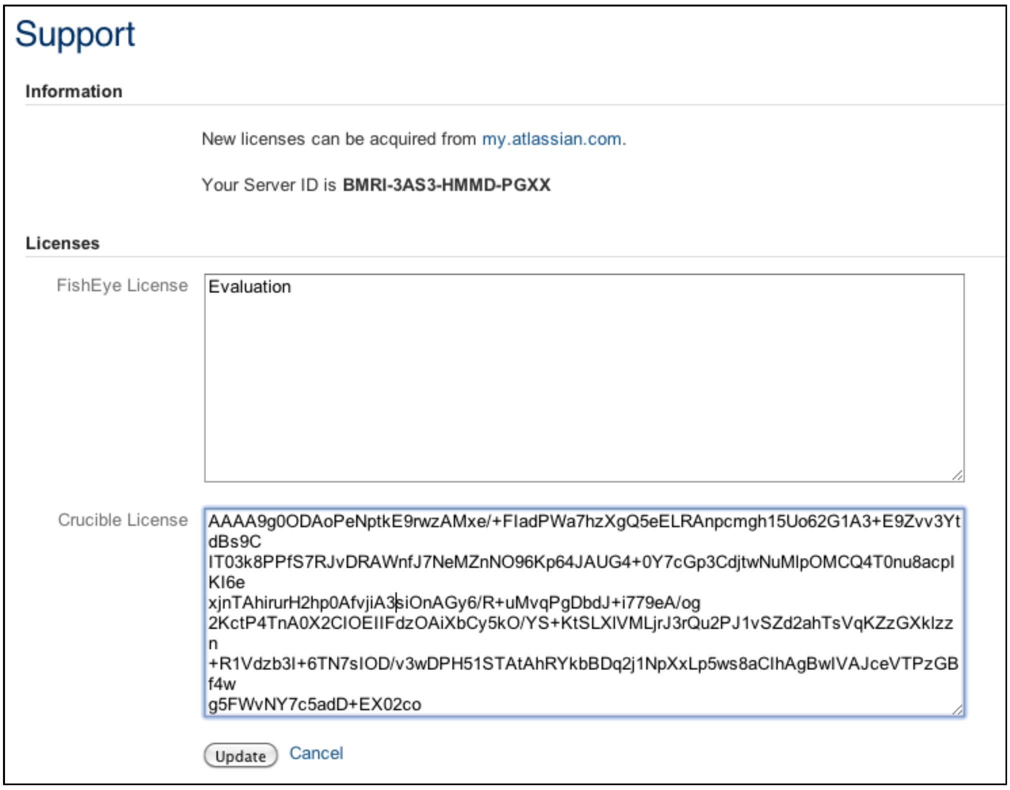
Note that you can access your current license, or obtain a new license, by going to my.atlassian.com.

Related pages:

- Crucible and FishEye
- What happens if I decide to stop using FishEye with Crucible

To update your Crucible license key:

1. Log in to Crucible Admin area.
2. Click **System Info** (under 'System Settings').
3. Click **Edit License** and paste your new license key into the appropriate text box.
4. Paste your new license into this box. Obtain a new license by clicking my.atlassian.com in the 'Information' section.
5. Click **Update**.



Support

Information

New licenses can be acquired from my.atlassian.com.

Your Server ID is **BMRI-3AS3-HMMD-PGXX**

Licenses

FishEye License: Evaluation

Crucible License:

```
AAAA9g0ODAoPeNptkE9rwzAMxe/+FladPWa7hzXgQ5eELRAnpcmgh15Uo62G1A3+E9Zvv3Yt
dBs9C
IT03k8PPfS7RJvDRAWnfJ7NeMZnNO96Kp64JAUG4+0Y7cGp3CdjtwNuMlpOMCQ4T0nu8acpl
Kl6e
xjnTAhirurH2hp0AfvjiA3siOnAGy6/R+uMvqPgDbdJ+i779eA/og
2KctP4TnA0X2ClOEIIFdzOAiXbCy5kO/YS+KtSLXlVMLjrJ3rQu2PJ1vSZd2ahTsVqKZzGXklzz
n
+R1Vdzb3l+6TN7sIOD/v3wDPH51STAtAhRYkbBDq2j1NpXxLp5ws8aClhAgBwlVAJceVTPzGB
f4w
g5FWvNY7c5adD+EX02co
```

Support Policies

Welcome to the support policies index page. Here, you'll find information about how Atlassian Support can help you and how to get in touch with our helpful support engineers. Please choose the relevant page below to find out more.

- Bug Fixing Policy
- New Features Policy
- Security Bugfix Policy

To request support from Atlassian, please raise a support issue in our online support system. To do this, visit support.atlassian.com, log in (creating an account if need be) and create an issue under Crucible. Our friendly support engineers will get right back to you with an answer.

Bug Fixing Policy

Summary

- Our Support team will help with workarounds and bug reporting
- We'll generally fix critical bugs in the next maintenance release
- We schedule non-critical bugs according to a variety of considerations

Report a bug

Building an add-on

Are you developing an add-on for an Atlassian product or using one of our APIs? Report any related bugs here.

Bug reports

Atlassian Support is eager and happy to help verify bugs—we take pride in it! Create an issue in our support system, providing as much information as you can about how to replicate the problem you're experiencing. We'll replicate the bug to verify, then lodge the report for you. We'll also try to construct workarounds if possible.

Search existing bug reports

Use our issue tracker to search for existing bugs, and watch the ones that are important to you. When you watch an issue, we'll send you an e-mail notification when the issue's updated.

How we approach bug fixing

Maintenance (bug fix) releases come out more frequently than major releases, and attempt to target the most critical bugs affecting our customers. The notation for a maintenance release is the final number in the version (the 1 in 6.0.1, for example).

If a bug is critical (production application down or major malfunction causing business revenue loss or high numbers of staff unable to perform their normal functions) we'll fix it in the next maintenance release, provided that:

- The fix is technically feasible (it doesn't require a major architectural change)
- It doesn't impact the quality or integrity of a product

For non-critical bugs, the developer assigned to fixing bugs prioritises the bug according to these factors:

- How many of our supported configurations are affected by the problem
- Whether there is an effective workaround or patch
- How difficult the issue is to fix
- Whether many bugs in one area can be fixed at one time

Developers responsible for fixing bugs also monitor comments on existing and new bugs, so you can comment to provide feedback if you need to. We give high priority to security issues.

When considering the priority of a non-critical bug, we try to determine a *value* score for a bug. The score takes into account the severity of the bug from our customers' perspective, how prevalent the bug is, and whether new features on our roadmap may render the bug obsolete. Our developers combine the value score with a *complexity* score (how difficult the bug is) when selecting issues to work on.

Further reading

See Atlassian Support Offerings for more support-related information.

New Features Policy

Summary

- We encourage and display customer comments and votes openly in our issue tracking system, http://jira.a

tlassian.com.
- We do not publish roadmaps.
- Product Managers review our most popular voted issues on a regular basis.
- We schedule features based on a variety of factors.
- Our Atlassian Bug Fixing Policy is distinct from this process.
- Atlassian provides consistent updates on the top 20 issues.

How to track what features are being implemented

When a new feature or improvement is scheduled, the 'fix-for' version will be indicated in the JIRA issue. This happens for the upcoming release only. We maintain roadmaps for more distant releases internally, but because these roadmaps are often pre-empted by changing customer demands, we do not publish them.

How Atlassian chooses what to implement

In every major release we *aim* to implement highly requested features, but it is not the only determining factor. Other factors include:

- **Customer contact**: We get the chance to meet customers and hear their successes and challenges at Atlassian Summit, Atlassian Unite, developer conferences, and road shows.
- **Customer interviews**: All product managers at Atlassian do customer interviews. Our interviews are not simply to capture a list of features, but to understand our customers' goals and plans.
- **Community forums**: There are large volumes of posts on answers, of votes and comments on jira.atlassian.com, and of conversations on community forums like groups on LinkedIn.
- **Customer Support**: Our support team provides clear insights into the issues that are challenging for customers, and which are generating the most calls to support
- **Atlassian Experts**: Our Experts provide insights into real-world customer deployments, especially for customers at scale.
- **Evaluator Feedback**: When someone new tries our products, we want to know what they liked and disliked and often reach out to them for more detail.
- **In product feedback**: The JIRA Issue Collectors that we embed our products for evaluators and our Early Access Program give us a constant pulse on how users are experiencing our product.
- **Usage data:** Are customers using the features we have developed?
- **Product strategy:** Our long-term strategic vision for the product.
- Please read our post on Atlassian Answers for a more detailed explanation.

How to contribute to feature development

Influencing Atlassian's release cycle
We encourage our customers to vote on issues that have been raised in our public JIRA instance, http://jira.atlassian.com. Please find out if your request already exists - if it does, vote for it. If you do not find it you may wish to create a new one.

Extending Atlassian products
Atlassian products have powerful and flexible extension APIs. If you would like to see a particular feature implemented, it may be possible to develop the feature as a plugin. Documentation regarding the plugin APIs is available. Advice on extending either product may be available on the user mailing-lists, or at Atlassian Answers.

If you require significant customisations, you may wish to get in touch with our partners. They specialise in extending Atlassian products and can do this work for you. If you are interested, please contact us.

Further reading

See Atlassian Support Offerings for more support-related information.

Security Bugfix Policy

See Security @ Atlassian for more information on our security bugfix policy.

Troubleshooting

Crucible Troubleshooting

Crucible Troubleshooting

The most common cause of FishEye/Crucible issues is an incorrect symbolic setup (trunk/branch/tag) for Subversion repositories. If you are using Subversion and your initial index is taking forever, double-check that your symbolic setup matches your repository.

FishEye runs with the default Java heap of 64 megabytes. This is sometimes problematic for FishEye, especially for Subversion repositories during the initial scan. You can give FishEye's JVM more memory by setting the FISHEYE_OPTS environment variable.

Starting Crucible with the command line options `--debug --debug-perf` will print a lot of information to Crucible's logs. This can give you an insight into what is happening and possibly where you are stuck. Attach these logs along with your `config.xml` to an Atlassian support ticket, to speed up your support request.

JIRA Integration Issues

Users are mapped to their own accounts when using Trusted Applications.

If you (or the general account used for JIRA access, if not using Trusted Applications) do not have the permissions to carry out the JIRA actions linked from Crucible, an error will occur. Depending on the error returned from JIRA, Crucible may not display the error correctly or display it at all, simply reporting that "An error has occurred". To investigate what the error was, you can access the Crucible debug log, named `fisheye-debug.log.YYYY-MM-DD` under the `dist.inst/var/log` folder of your Crucible installation. In the debug log, look for the date and time when your error took place. Here, you will be able to follow the links and see what error the JIRA instance was producing by clicking through to JIRA.

If you are using JIRA 4.0 you will not be able to create subtasks in versions of Crucible prior to 2.0.5. If you are affected by this bug, please upgrade to at least 2.0.6 (2.0.5 is affected by another bug CRUC-2471).

Problems with very long comments and MySQL migration

Affects Version

This issue was introduced in Crucible 2.0 and fixed in Crucible 2.1.

Issue Symptoms

There is a known issue with Crucible 2.0.x and very long comments when migrating your database to MySQL. In some circumstances, this might result in truncation of very long comments, causing data loss.

Depending on your MySQL configuration, you may see an error message like this while migrating to MySQL, causing the migration to fail:

```
2009-07-16 16:56:12,390 ERROR [ThreadPool1] fisheye.app
com.cenqua.crucible.actions.admin.database.DBEditHelper-doGet -
Database migration failed:
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
java.sql.BatchUpdateException: Data truncation: Data too long for column
'cru_message' at row 1
```

ℹ You may not see the message if you are running MySQL with default settings.

For more information, see the JIRA issue.

Workaround

If your data contains very long comments or review descriptions (longer than 21,845 multibyte unicode characters), consider avoiding use of MySQL until you can upgrade the product. Alternatively, use PostgreSQL or the default (built-in) HSQLDB database.

This issue is now resolved. This issue was introduced in Crucible 2.0 and fixed in Crucible 2.1.

Requesting Support

If you require assistance in resolving the problem, please raise a support request under the Crucible project.

Contributing to the Crucible documentation

Would you like to share your Crucible hints, tips and techniques with us and with other Crucible users? We welcome your contributions.

On this page:

Updating the documentation

Have you found a mistake in the documentation, or do you have a small addition that would be so easy to add yourself rather than asking us to do it? You can update the documentation page directly

Getting permission to update the documentation

Please submit the **Atlassian Contributor License Agreement.**

Our style guide

Please read our short guidelines for authors.

How we manage community updates

Here is a quick guide to how we manage community contributions to our documentation and the copyright that applies to the documentation:

- **Monitoring by technical writers.** The Atlassian technical writers monitor the updates to the documentation spaces, using RSS feeds and watching the spaces. If someone makes an update that needs some attention from us, we will make the necessary changes.
- **Wiki permissions.** We use wiki permissions to determine who can edit the documentation spaces. We ask people to sign the Atlassian Contributor License Agreement (ACLA) and submit it to us. That allows us to verify that the applicant is a real person. Then we give them permission to update the documentation.
- **Copyright.** The Atlassian documentation is published under a Creative Commons CC BY license. Specifically, we use a Creative Commons Attribution 2.5 Australia License. This means that anyone can copy, distribute and adapt our documentation provided they acknowledge the source of the documentation. The CC BY license is shown in the footer of every page, so that anyone who contributes to our documentation knows that their contribution falls under the same copyright.

Contributing documentation in other languages

Have you written a guide to Crucible in a language other than English, or translated one of our guides? Let us know, and we will link to your guide from our documentation.

RELATED TOPICS

Author Guidelines
Atlassian Contributor License Agreement

Glossary

Code review terminology can be confusing as there are many different words for the concepts, roles and process. Crucible has adopted the following terms (click for definitions):

approve

authors in Crucible

code review

comment

creator

defect

moderator

participant

permission scheme

permissions in Crucible

projects in Crucible

review duration

reviewer

role

state

statement of objective

users in Crucible

approve

Issuing a review to the reviewers is known as *approving* the review.

authors in Crucible

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

Note: to map your repository username to your FishEye/Crucible username, see Changing your User Profile.

code review

Without prejudice to 'code inspection', 'peer review' or a myriad of other terms, Crucible uses the phrase *code review* for simplicity.

See Getting Started.

comment

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

See Commenting on reviews.

creator

The *creator* is the person who creates the review. In most cases this person will also act as moderator.

defect

A *defect* is a comment flagged as something that requires addressing and includes optional defect classifications.

See Flagging Defects and Customizing the defect classifications.

moderator

The *moderator* is the person responsible for creating the review, approving the review, determining when reviewing is finished, summarizing the outcomes and closing the review. By default, the moderator is the creator. See also author, the person whose changes to the code are to be reviewed.

participant

Crucible uses the terms creator, author, moderator, and reviewer to describe the *roles* of review participants.

permission scheme

A *permission scheme* assigns particular permissions to any or all of the following:

- Particular Users.
- Particular Groups.
- All logged-in users.
- Anonymous Users
- People in particular Review Roles, such as:
    - Author
    - reviewer
    - creator
    - moderator

The scheme's permissions will apply to all reviews belonging to the project(s) with which the scheme is associated.

You can create as many permission schemes as you wish. Each permission scheme can be associated with many projects or just one project, allowing you to tailor appropriate permissions for individual projects as required.

See Creating a permission scheme.

permissions in Crucible

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
Abandon	Ability to abandon (i.e. cancel) a review.	Creator Author Moderator
Approve	Ability to approve a review (i.e. issue it to the reviewers).	Creator Author Moderator

Close	Ability to close a review once it has been summarized.	Creator Author Reviewer Moderator
Comment	Ability to add or remove a comment to or from a review.	Creator Author Reviewer Moderator
Complete	Ability of a reviewer to change their individual review status to Complete.	Reviewer
Create	Ability to create a review.	All logged-in users
Delete	Ability to delete a review.	Creator Author Moderator
Edit Review Details	Ability to edit a review's details and change the set of revisions being reviewed.	Creator Author Reviewer Moderator
Re-Open	Ability to re-open a closed or abandoned review.	Creator Author Reviewer Moderator
Recover	Ability to resurrect an abandoned (i.e. canceled) review.	Creator Author Reviewer Moderator
Reject	Ability to reject a review submitted for approval (i.e. prevent it from being issued to reviewers).	Creator Author Moderator
Submit	Ability to submit a review for approval (i.e. request that the review be issued to the reviewers).	Creator Author Moderator
Summarize	Ability to summarize a review. (Normally this would be done after all reviewers have completed their review.)	
Uncomplete	Ability of a reviewer to change their individual review status from Complete to Uncomplete.	Reviewer
View	Ability to view a review. (People without this permission will not know that the review exists.)	Anonymous users All logged-in users Creator Author Reviewer Moderator

projects in Crucible

A Crucible *project* provides a way to group and manage related reviews – typically reviews that are all involved with the same software project. A Crucible project allows you to

- define default moderators, authors and reviewers for the reviews in that project.
- define which people are eligible to be reviewers for the reviews in that project.
- use permission schemes to restrict who can perform particular actions (e.g. 'Create Review') in that project.

Every Crucible review belongs to a project. Each project has a *name* (e.g. ACME Development) and a *key* (e.g. ACME). The project key becomes the first part of that project's *review keys*, e.g. ACME-101, ACME-102, etc:

By default, Crucible contains one project. This default project has the key 'CR' and the name 'Default Project'. See Creating a project.

review duration

The *review duration* is the period of time for which a review will run.

See Editing a project.

reviewer

A *reviewer* is a person assigned to review the change. Reviewers can make comments and indicate when they have completed their review. The moderator and author are implicitly considered to be participants of the review, but are not reviewers.

A Crucible admin can specify *default* reviewers, who are added automatically to all reviews in a project, and can also restrict participation in a project's reviews to just *allowed* users. See Editing a project.

role

See participant.

state

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

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

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

A review may also be in the following states:

Abandoned	This happens when a review is deleted.
Rejected	Any reviews that a moderator has rejected.
Needs Fixing	This means that the review state is not understood by Crucible, and indicates a programming or data issue. The review moderator can move the review into a known state if this happens.

statement of objective

A *statement of objective* is an optional text description of the review and any specific areas the reviewers should focus on.

users in Crucible

A *user* is a person using Crucible.

Collecting analytics in Crucible

We are continuously working to make Crucible better. Data about how you use Crucible helps us do that. We have updated our Privacy Policy so that we may collect usage data automatically, unless you disable collection. The data we collect includes information about the systems on which your installation of Crucible is operating, the features you use in Crucible, and your use of common IT terminology within the product. For more details, see our Privacy Policy, in particular the 'Analytics Information from Downloadable Products' section.

See also our End User Agreement.

How to change data collection settings?

You can opt in to, or out of, data collection at any time. A Crucible admin can change the data collection settings by going to **Analytics** (under 'Global Settings') in the Crucible admin area.

How is data collected?

We use the Atlassian Analytics plugin to collect event data in Crucible. Analytics logs are stored locally and then periodically uploaded to a secure location.