![Atlassian](Atlassian logo)

User Guide for Crucible 2.10

Contents

Getting Started
Supported Platforms

This page shows the supported platforms for **Crucible 2.10.x** and its minor releases.

Key: ✅ = Supported; ❌ = Not Supported

Java Version		
JRE / JDK	✅ 1.6 ✅ 1.7	Crucible requires **Java Runtime** (JDK or JRE), version as noted. Pre-release/Early access versions of the Java Runtime are *not supported*. You can download a Java Runtime for Windows/Linux/Solaris. On Mac OS X, the JDK is bundled with the operating system. We highly recommend that you use the Oracle JVM (or use the default Mac OS X JVM), as other implementations have not been tested. Please note: • Once you have installed the JDK, 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.
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.
Linux	✅	
Apple Mac OS X	✅	
Databases		
MySQL	✅ MySQL Enterprise Server 5.x ✅ MySQL Community Server 5.x	FishEye supports a number of external databases, as listed. See the Crucible Database documentation for further details.

PostgreSQL	✅ 8.2, 8.3, 8.4	For MySQL:
Oracle	✅ 11g	• For 5.0, version must be 5.0.21 or later
SQL Server	✅ 2005, 2008, 2008 R2	• For 5.1, version must be 5.1.10 or later
HSQLDB ([3])	✅ 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.
Web Browsers		
Microsoft Internet Explorer	✅ 8.0, 9.0 ❌ 6.0 and 7.0 are not supported	
Mozilla Firefox	✅ Latest stable version supported ✅ 3.6, 4.0	
Safari	✅ Latest stable version supported ✅ 4, 5	
Chrome	✅ Latest stable version supported	
Version Control Systems		
Subversion (SVN)	✅ Server 1.1, 1.2, 1.3, 1.4, 1.5, 1.6. 1.7	The client uses SVNkit or native JavaHL.
CVS (and CVSNT)	✅ All versions	
Perforce (P4)	✅ Client version 2007.3 or later ✅ Server version 2005.1 or later	The Server must support the ztag tagged protocol.
Git	✅ 1.7.1.1 or later	
Mercurial (Hg)	✅ 1.5.1 or later	Mercurial 2.1 has a bug that makes it incompatible with Crucible. Please use Mercurial 2.1.1 or later.

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.

End of Support Announcements for Crucible

This page contains announcements of the end of support for various platforms and browsers when used with Crucible. This is summarised in the table below. Please see the sections following for the full announcements.

End of Support Matrix for Crucible

Platform	Crucible End of Support
MySQL 5.0	January 2012
PostgreSQL 8.0 and 8.1	January 2012
IBM ClearCase (all versions)	4 April 2012 (announcement)

The table above summarises information regarding the end of support announcements for upcoming Crucible releases. If a platform (version) has already reached its end of support date, it is not listed in the table.

> ℹ **Why is Atlassian ending support for these platforms?**
>
> Atlassian is committed to delivering improvements and bug fixes as fast as possible. We are also committed to providing world class support for all the platforms our customers run our software on. However, as the complexity of our applications grows, the cost of supporting multiple platforms increases exponentially. Each new feature has to be tested on several combinations of application servers, databases, web browsers, etc, with setup and ongoing maintenance of automated tests. Moving forward, we want to reduce the time spent there to increase Crucible development speed significantly.

On this page (most recent announcements first):

- Deprecated Database Support for Crucible (4 October 2011)
- Deprecated Web Browsers for Crucible (21 March 2011)
- Deprecated Java Platforms for Crucible (21 March 2011)
- Deprecated SCM Repository Support for Crucible (4 April 2011)

Deprecated Database Support for Crucible (4 October 2011)

This section announces the end of Atlassian support for certain databases for Crucible.

We will **stop supporting older versions of databases** as follows:

- For the next major version of Crucible, in January 2012, support for MySQL 5.0, PostgreSQL 8.0 and 8.1 will end.

Please refer to the Supported Platforms for more details regarding platform support for Crucible. If you have questions or concerns regarding these announcements, please email `eol-announcement at atlassian dot com`.

Database	Support End Date
MySQL 5.0	January 2012

| PostgreSQL 8.0 and 8.1 | January 2012 |

End of Support Notes for MySQL 5.0 and PostgreSQL 8.0 and 8.1:

- Atlassian intends to end of life support for MySQL 5.0, PostgreSQL 8.0 and 8.1 in January 2012. The release of Crucible after January 2012 will not support MySQL 5.0, PostgreSQL 8.0 or 8.1.
- As mentioned above, the releases of Crucible before January 2012 will contain support for MySQL 5.0 and PostgreSQL 8.0 and 8.1.

Deprecated Web Browsers for Crucible (21 March 2011)

This section announces the end of Atlassian support for certain web browsers for Crucible.

We will **stop supporting older versions of web browsers** as follows:

- From Crucible 2.6, due in May 2011, support for Internet Explorer 7 will end.

The details are below. Please refer to the Supported Platforms for more details regarding platform support for Crucible. If you have questions or concerns regarding this announcement, please email `eol-announcement at atlassian dot com`.

End of Life Announcement for Web Browser Support

Web Browsers	Support End Date
Internet Explorer 7	When Crucible 2.6 releases (target May 2011)

Internet Explorer 7 Notes:

- Crucible 2.5 is the last version to officially support Internet Explorer 7.
- Crucible 2.6 is currently targeted to release in May 2011 and will not be tested with Internet Explorer 7. After the Crucible 2.6 release, Atlassian will not provide fixes in older versions of Crucible for bugs affecting Internet Explorer 7.

Deprecated Java Platforms for Crucible (21 March 2011)

This section announces the end of Atlassian support for certain Java Platforms for Crucible.

We will **stop supporting the following Java Platforms**:

- From Crucible 2.6, due in May 2011, support for Java Platform 5 (JDK/JRE 1.5) will end.

We are ending support for Java Platform 5, in line with Sun's Java SE Support Road Map (i.e. "End of Service Life" for Java Platform 5 dated October 30, 2009). We are committed to helping our customers understand this decision and assist them in updating to Java Platform 6, our supported Java Platform.

The details are below. Please refer to the Supported Platforms for more details regarding platform support for Crucible. If you have questions or concerns regarding this announcement, please email `eol-announcement at atlassian dot com`.

End of Life Announcement for Java Platform Support

Java Platform	Support End Date

Java Platform 5 (JDK/JRE 1.5)	When Crucible 2.6 releases, due May 2011

Java Platform 5 End of Support Notes:

- Crucible 2.5 is the last version to officially support Java Platform 5 (JDK/JRE 1.5).
- Crucible 2.6 is currently targeted to release in May 2011 and will not be tested with Java Platform 5 (JDK/JRE 1.5). After the Crucible 2.6 release, Atlassian will not provide fixes in older versions of Crucibl for bugs affecting Java Platform 5 (JDK/JRE 1.5).

Deprecated SCM Repository Support for Crucible (4 April 2011)

This section announces the end of Atlassian support for certain SCM repositories for Crucible. End of support means that Atlassian will remove all functionality related to certain SCM repositories past the specified date. Releases before that date will contain the functionality that supports the SCM, however, Atlassian will fix only critical bugs that affect functionality for that SCM, and will not add any new features for that SCM. After the specified date, Atlassian will not support the functionality in any version of Crucible.

Please refer to the [Supported Platforms](#) for more details regarding platform support for Crucible. If you have questions or concerns regarding these announcements, please email `eol-announcement at atlassian dot com`.

SCM Repository	Support End Date
IBM ClearCase (all versions)	4 April 2012

IBM ClearCase End of Support Notes:

- Atlassian intends to end of life IBM ClearCase functionality on 4 April 2012. The release of Crucible afte 4 April 2012 will not contain any IBM ClearCase functionality.
- As mentioned above, the releases of Crucible before 4 April 2012 will contain support for IBM ClearCase However, we will only be fixing critical bugs related to IBM ClearCase and will not be adding any feature
- After 4 April 2012, Atlassian will not support IBM ClearCase functionality in any version of Crucible

End of Support Announcement for IBM ClearCase

Support in Crucible for [IBM ClearCase](#) ended on **April 4th 2012**. Crucible 2.8, and later versions, do not have support for ClearCase.

We have made these decisions to reduce the testing time required for each release and to help us to deliver market-driven features faster.

You can stay on older versions of Crucible to support your existing installations with ClearCase. However, Atlassian will not be providing any ClearCase-related support for any Crucible version after 4 April 2012, and **ha s removed all functionality** related to ClearCase from Crucible versions released after April 4th 2012. We are committed to helping our customers understand this decision and to assist you in migrating to a different SCM, if needed.

For more details about the announcement, please refer to this page: [End of Support Announcements for Crucible](#).

Installing Crucible on Windows

Hey! We're going to perform a clean install of Crucible on Windows. 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:***
>
> - Running Crucible as a Windows service
> - Upgrading from FishEye to Crucible
> - Installing Crucible on Linux and Mac
> - Starting to use Crucible
> - Supported Platforms

2. Check your version of Java

In a command prompt, run this:

```
java -version
```

The version of Java should be **1.6.0** or higher. If you intend to Running Crucible as a Windows service, using the Java Service Wrapper, you should use 32-bit Java (even on a 64-bit machine), and the JDK rather than the JRE (so as to take advantage of the `-server` parameter).

▼ **If you don't see Java 1.6.0 or higher, then get Java...**

> Download and install the Java Platform JDK from Oracle's website.
>
> ⚠ *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 **1.6.0** or higher.

3. 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 that does *not* contain spaces. We recommend that JAVA_HOME should point to the Java executable in your 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 2. 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 as the **Variable value**. Don't use a trailing backslash. We recommend that JAVA_HOME should point to the Java executable in your PATH.
4. Now, in a new command prompt, try running `'%JAVA_HOME%\bin\java -version'`. You should see the same version of Java as you saw above.

4. 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.

5. Now it's time to get Crucible

Download Crucible from the Atlassian download site.

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 `<FishEye home directory>` in these instructions, for compatibility with the FishEye install instructions.
- 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.

⚠ You *should not* locate your Crucible instance directory inside the `<FishEye home directory>` — these should be entirely separate locations. If you do put the instance directory in the `<FishEye 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.

Create your Crucible instance directory, and then tell Crucible where you created it by setting a `FISHEYE_INST` environment variable, 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 "FISHEYE_INST" as the **Variable name**, and the absolute path to your new Crucible instance directory as the **Variable value**. Don't use a trailing backslash.
4. Now copy the newly extracted `<FishEye home directory>\config.xml` file to the root of your new FishEye instance directory.

ℹ️ Note that if Crucible is run as a Windows service using the Java Service Wrapper, Crucible-specific environment variables such as `FISHEYE_INST` are ignored – these must be set in the wrapper.conf file. See Ru nning Crucible as a Windows service .

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.

7. Start Crucible!

In a command prompt, change directory to `<FishEye home directory>` and run this:

```
bin\start.bat
```

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 organises 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 command prompt, change directory to `<FishEye home directory>` and run this:

```
bin\stop.bat
```

Running Crucible as a Windows service

Crucible can be run as a service under Microsoft Windows using a Java Service Wrapper.

The service wrapper provides the following benefits:

- Allows Crucible, which is a Java application, to be run as a Windows Service.
- No need for a user to be logged on to the system at all times, or for a command prompt to be open and running on the desktop to be able to run Crucible.
- The ability to run Crucible in the background as a service, for improved convenience, system performance and security.
- Crucible is launched automatically on system startup and does not require that a user be logged in.
- Users are not able to stop, start, or otherwise tamper with Crucible unless they are an administrator.
- Provides advanced failover, error recovery, and analysis features to make sure that Crucible has the maximum possible uptime.

⚠ Please note that:

- This page should be read in conjunction with Installing Crucible on Windows.
- You should use 32-bit Java to run the service wrapper provided via the link in the install instructions below, even on a 64-bit machine.
- You should use the Java JDK, rather than the JRE, to take advantage of the `-server` parameter, provided in the Wrapper configuration of wrapper.zip, which enables the Java HotSpot(TM) Server VM. See the note below for details.

> ***On this page:***
>
> - Installing the Java Service Wrapper
> - Setting Crucible environment variables for Windows Services
> - Troubleshooting
>   - Extracting files from wrapper.zip
>   - Warning when using 64-bit Java JDK
>   - Wrapper configuration and "-server" parameter

> ***Related pages:***
>
> - Installing Crucible on Windows

Installing the Java Service Wrapper

To install the Java Service Wrapper on Windows:

1. Download wrapper.zip from here.
2. Unzip the wrapper zip file into your `<FishEye home directory>` (that is, the directory into which Crucible was originally installed). Note, the resulting folder structure should be `<FishEye home directory>\wrapper`, `<FishEye home directory>\wrapper\bin`, etc and NOT `<FishEye home directory>\wrapper\wrapper`, `<FishEye home directory>\wrapper\wrapper\bin`. The location of the wrapper directory is important.
3. Tell the wrapper where to find the Java JDK by editing the `<FishEye home directory>\wrapper\conf\wrapper.conf` file, replacing this:

```
# Java Application
wrapper.java.command=java
```

with the following, and comment out the option you don't wish to use:

```
# Java Application

# Option 1: If you have JAVA_HOME defined in your Windows system
environment variables, then you can use:
wrapper.java.command=%JAVA_HOME%/bin/java

# Option 2: If you have multiple JDKs installed, and you don't want to use
a Windows environment variable to specify which one to use, provide the
absolute path to where the JDK is installed (e.g.
C:/Java/jdk1.7.0_05/bin/java):
wrapper.java.command=C:/<path to Java location>/bin/java
```

To get confirmation in the wrapper log that the wrapper is using the correct Java JDK, add the following lines to the `wrapper.conf` file:

```
# Tell the Wrapper to log the full generated Java command line.
wrapper.java.command.loglevel=INF
```

You can find the logs at `<FishEye home directory>\var\log\wrapper.log`.

4. Set the FISHEYE_INST environment variable (and other Crucible-specific environment variables) in the `<FishEye home directory>\wrapper\conf\wrapper.conf` file, following the instructions below.
5. Install Crucible as a service as follows:
   a. Open an Administrator command prompt by searching for 'Command prompt' in the Windows Start menu, right-clicking on **Command Prompt** and then choosing **Run as administrator**.
   b. Change directory to `<FishEye home directory>\wrapper` and run `bin\Fisheye-Install -NTService.bat`. If you run into any problems starting the wrapper, you'll find its logs in `<FishE ye home directory>\var\log\wrapper.log`.
6. Start the Crucible service (which has the name 'Fisheye') from the Windows Control Panel; you can search in the Start menu for 'services', and in the list of services, right-click on the 'Fisheye' item and choose **Start**. You can also stop the Crucible service in this way.

ⓘ Please note that:

- If you make changes to the wrapper.conf file, having already started the service, you need to stop and then restart the service for it to make use of the changed configuration.
- If in future you move the FishEye home directory, you will need to uninstall (using `Fisheye-Uninstall -NTService.bat`) and then reinstall the Crucible service.

Setting Crucible environment variables for Windows Services

Please note, that if you run Crucible as a Windows service, any Crucible-specific environment variables must be set in your `<FishEye home directory>\wrapper\conf\wrapper.conf` file.

If you run into any problems starting the wrapper, you'll find its logs in `<FishEye home`

```
directory>\var\log\wrapper.log.
```

If there are other Java parameters you wish to add, then you will need to add them under the additional parameters section, e.g.

```
# JDK Additional Parameters for jmx
wrapper.java.additional.4=-Dcom.sun.management.jmxremote
wrapper.java.additional.5=-Dcom.sun.management.jmxremote.port=4242
wrapper.java.additional.6=-Dcom.sun.management.jmxremote.authenticate=fa
lse
wrapper.java.additional.7=-Dcom.sun.management.jmxremote.ssl=false
wrapper.java.additional.8=-Dcom.sun.management.jmxremote.authenticate=fa
lse
wrapper.java.additional.9=-Dcom.sun.management.jmxremote.password.file=.
/wrapper/jmxremote.password
wrapper.java.additional.10=-Dwrapper.mbean.name="wrapper:type=Java
Service Wrapper Control"
```

To add the FISHEYE_INST environment variable, the Java MaxPermSize parameter, or the -Xrs options, use the following:

```
wrapper.java.additional.11=-Dfisheye.inst="c:/path/to/FISHEYE_INST"
wrapper.java.additional.12=-XX:MaxPermSize=128m
wrapper.java.additional.13=-Xrs
```

Note that the the -Xrs options should be used when running Crucible as a service under Windows to prevent the JVM closing when an interactive user logs out.

Your memory settings can also be found in this file:

```
# Initial Java Heap Size (in MB)
wrapper.java.initmemory=256

# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=1024
```

Increase these values if you have a large repository or expect to use more memory (init of 256, and a max of 1024 are the default values).

In Fisheye/Crucible 1.6.4 and higher, you can check the JVM input arguments by clicking **System info**, under 'System Settings' in the admin area.

Troubleshooting

Extracting files from wrapper.zip

Some customers have reported trouble running the wrapper. These can be avoided by:

- Uncompressing wrapper.zip with Winzip or WinRar rather than using the **Extract All** command in the Windows right-click contextual menu.
- If the wrapper.zip filename appears green instead of black in Windows Explorer, decrypt it, prior to

unzipping its contents, by right-clicking on the file, choose **Properties**, click the **Advanced** button, then clear the **Encrypt contents to secure data** checkbox.

Warning when using 64-bit Java JDK

When using a 64-bit Java JDK with the wrapper obtained via the link in the install instructions above, you may see the following in the wrapper.log file:

> *WARNING - Unable to load the Wrapper's native library 'wrapper.dll'. The file is located on the path at the following location but could not be loaded: C:\installs\service\fisheye28\wrapper\lib\wrapper.dll.*
>
> *Please verify that the file is readable by the current user and that the file has not been corrupted in any way. System signals will not be handled correctly.*

This is caused by using a 64-bit JDK (even on a 64-bit machine). Changing to a 32-bit version of the JDK will prevent this warning. Community Edition versions of the 64-bit Windows Java Service Wrapper are not currently available.

Wrapper configuration and "-server" parameter

Please note that the wrapper configuration provided above uses the `-server` parameter to enable the [Java HotSpot(TM) Server VM](). This feature is only available if you use the JDK. If you use the JRE you will likely get the following error in your logs:

> *INFO | jvm 1 | 2010/12/20 18:19:28 | Error: missing `server' JVM at `C:\Program Files\Java\jre6\bin\server\jvm.dll'.*

A common issue is that customers remove the `-server` parameter from the `wrapper.conf` file. Please note that if you do this, the wrapper script will ignore any of the following JVM parameters in the file unless you change the sequence to be in order, starting from `wrapper.java.additional.1`. This is an issue with the Wrapper application.

In this situation it's best to install and run Fisheye/Crucible with the JDK to get all the advantages of the `-server` functionality. You also need to force the wrapper to use the JDK by specifying the path to the Java JDK in the `wrapper.conf` file, as described in the installation instructions above.

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. Check your version of Java

In a terminal, run this:

```
java -version
```

The version of Java should be **1.6.0** or higher.

▼ **If you don't see Java 1.6.0 or higher, then get Java...**

> Download and install the Java Platform JDK from Oracle's website.
>
> Now try running `'java -version'` again to check the installation. The version of Java should be **1.6.0** or higher.

3. 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_HO ME" >> /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` file: ``` JAVA_HOME="path/to/JAVA_HOME" export JAVA_HOME ``` where `path/to/JAVA_HOME` may be like: `/System/Library/Frameworks/JavaVM.framework/Versions/CurrentJDK/Home/` Refresh your ~/.profile in the terminal and confirm that `JAVA_HOME` is set: ``` source ~/.profile $JAVA_HOME/bin/java -version ``` You should see a version of Java that is **1.6.0** or higher, like this: ``` java version "1.6.0_24" ```

4. Create a dedicated Crucible user (recommended)

For production installations, we recommend that you create a new dedicated 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.

5. Now it's time to get Crucible

Download Crucible from the Atlassian download site.

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 `<FishEye home directory>` in these instructions, for consistency with the FishEye documentation.
- 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 it 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 `<FishEye home directory>\config.xml` file to the root of your new FishEye instance directory.

⚠️ You *should not* locate your Crucible instance directory inside the `<FishEye home directory>` — they should be entirely separate locations. If you do put the instance directory in the `<FishEye 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 `<FishEye 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 `<FishEye 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:

- Connecting to JIRA in the Setup Wizard
- Troubleshooting
- Notes

Related pages:

- 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-administrat ors`.

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

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 '**Appl ication 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 '**A pplication 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 '**Applic ation URL**' matches the base URL of your Stash server.
   c. Click the '**Delete**' link next to the application link that you want to delete.
   d. A confirmation screen will appear. Click the '**Confirm**' button 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 '**Appl ication 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 '**A pplication 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 '**Applic ation URL**' matches the base URL of your Stash server.

    c. Click the '**Delete**' link next to the application link that you want to delete.

    d. A confirmation screen will appear. Click the '**Confirm**' button 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 the '**Next**' button.

    d. Enter the following information:

- '**Create a link back to this server**' – Tick this check box 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 the '**Next**' button.

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

- '**The servers have the same set of users**' – Tick this check box, because the users are the same in both applications.
- '**These servers fully trust each other**' – Tick this check 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 the '**Create**' button to create the application link.

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 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 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 to JIRA for user management.

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 will guide you though the basics of using Crucible. By the end of it you should know how to:

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

Assumptions

This page assumes 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.

Add a repository

In this section we're going to add a repository to Crucible.

Click on **Add Existing...** in the **Repositories** listing of the Administration area.
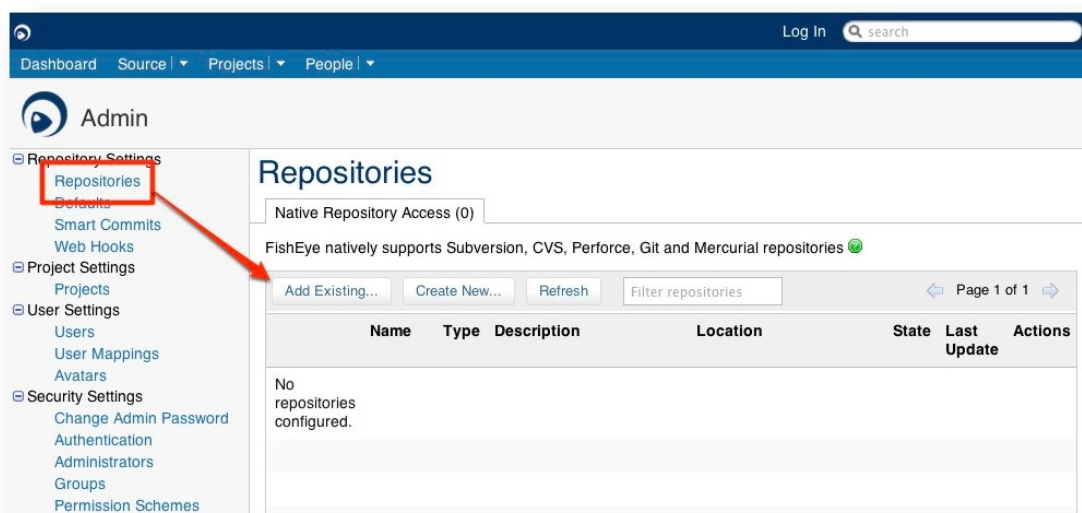
Choose the repository type and fill in the name and description.

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 should be turned on and if the repository should not be indexed right away then click **Add** to finish the process.
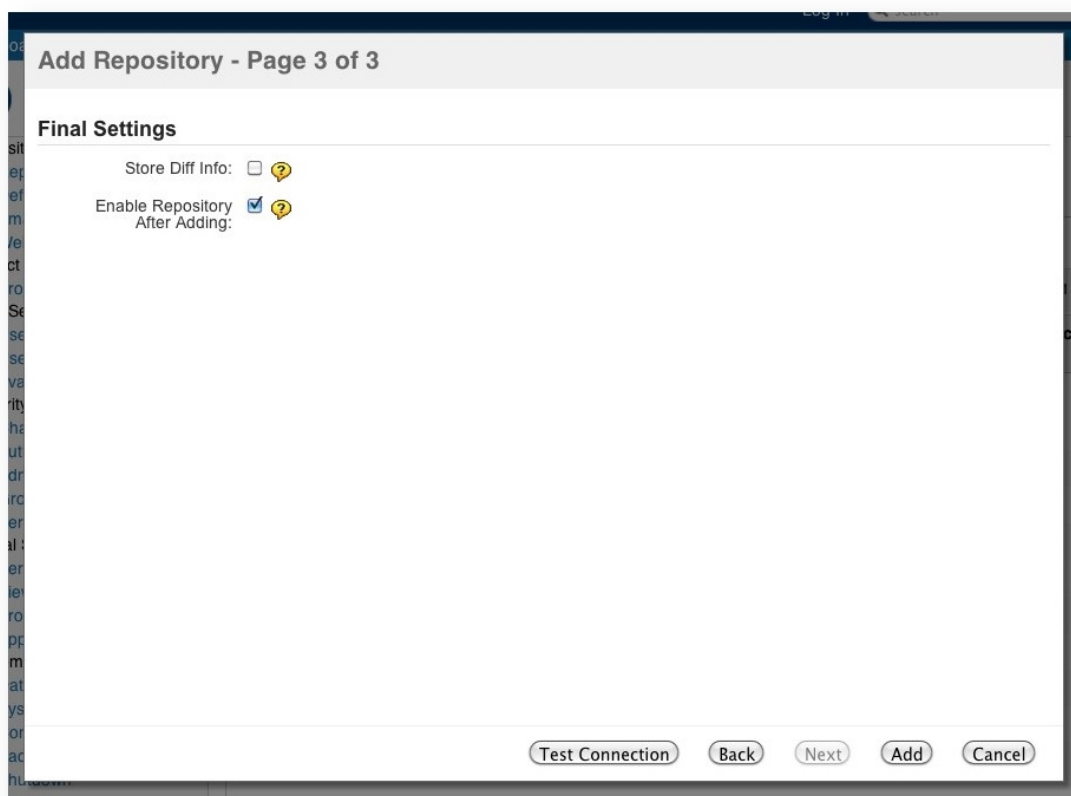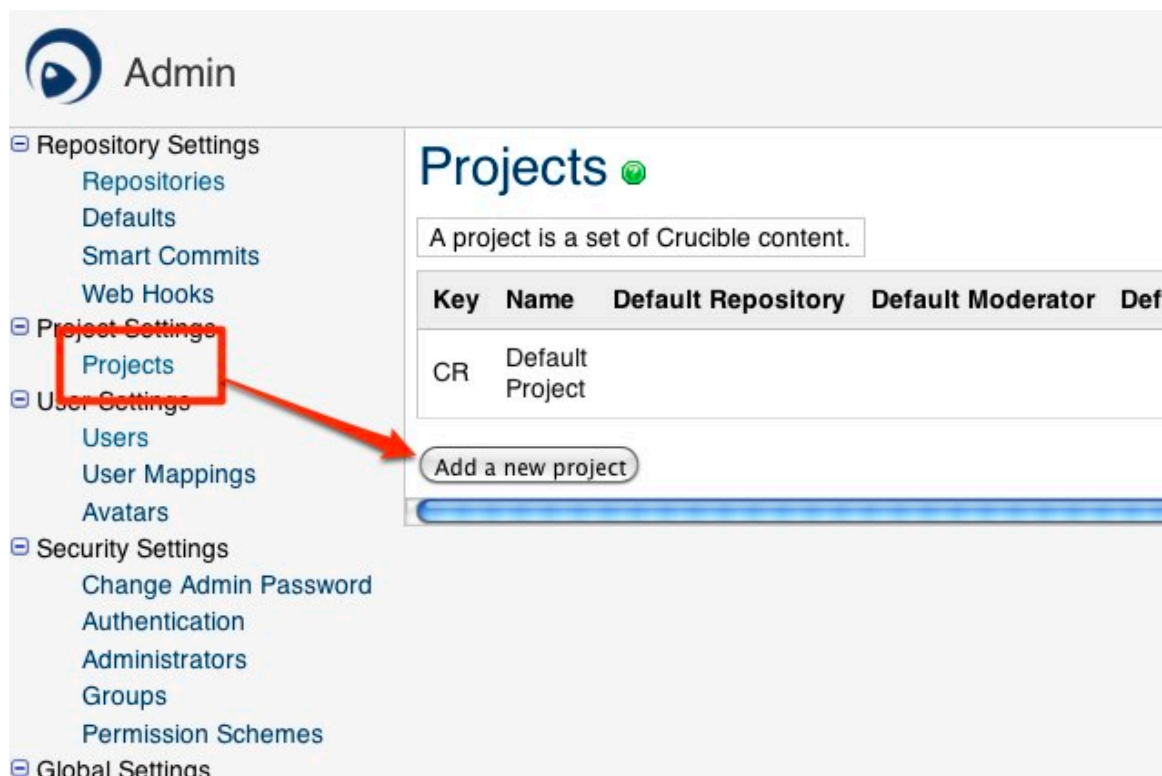
Your repository is now created in **Crucible** and the indexing should have started.
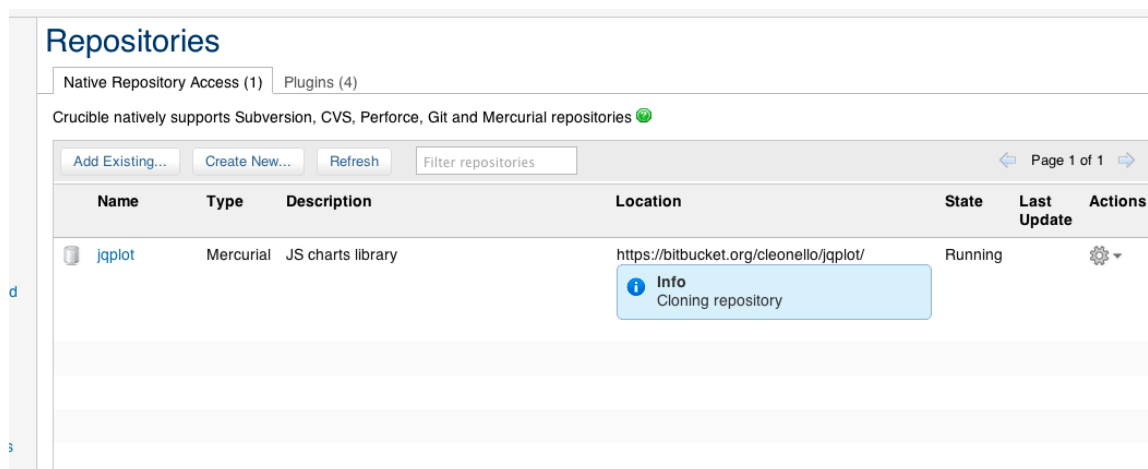
Create a project in Crucible

Crucible comes with a default project with the key CR but you will most likely have to create your own projects that will contain your reviews. This is achieved in a couple of steps.

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

Fill in the form with the default settings for the project and hit **Save**.
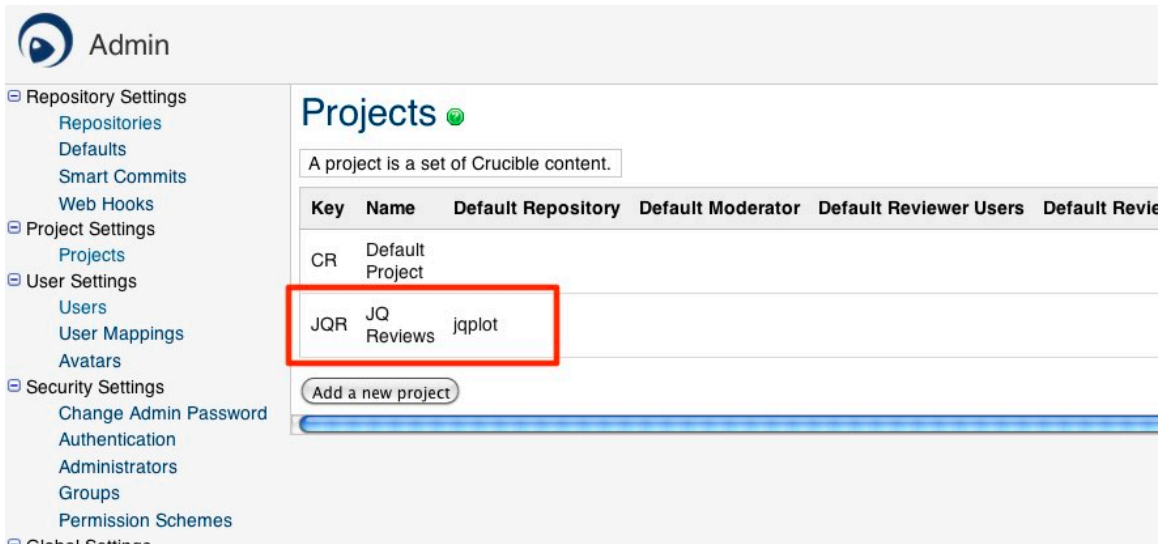
You should then be taken back to the **Projects** listing where you can see your new project has been created.
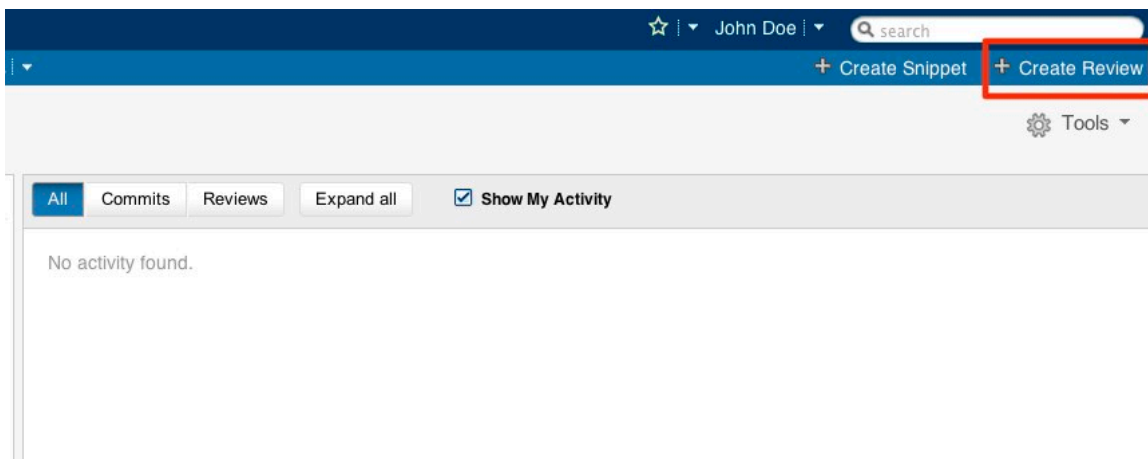
Create a review

Now that you have your own project it is possible to create reviews in it.

> ⚠ You need to be logged in to create reviews.

From the **Dashboard** click on **Create review** to open the review creation form.
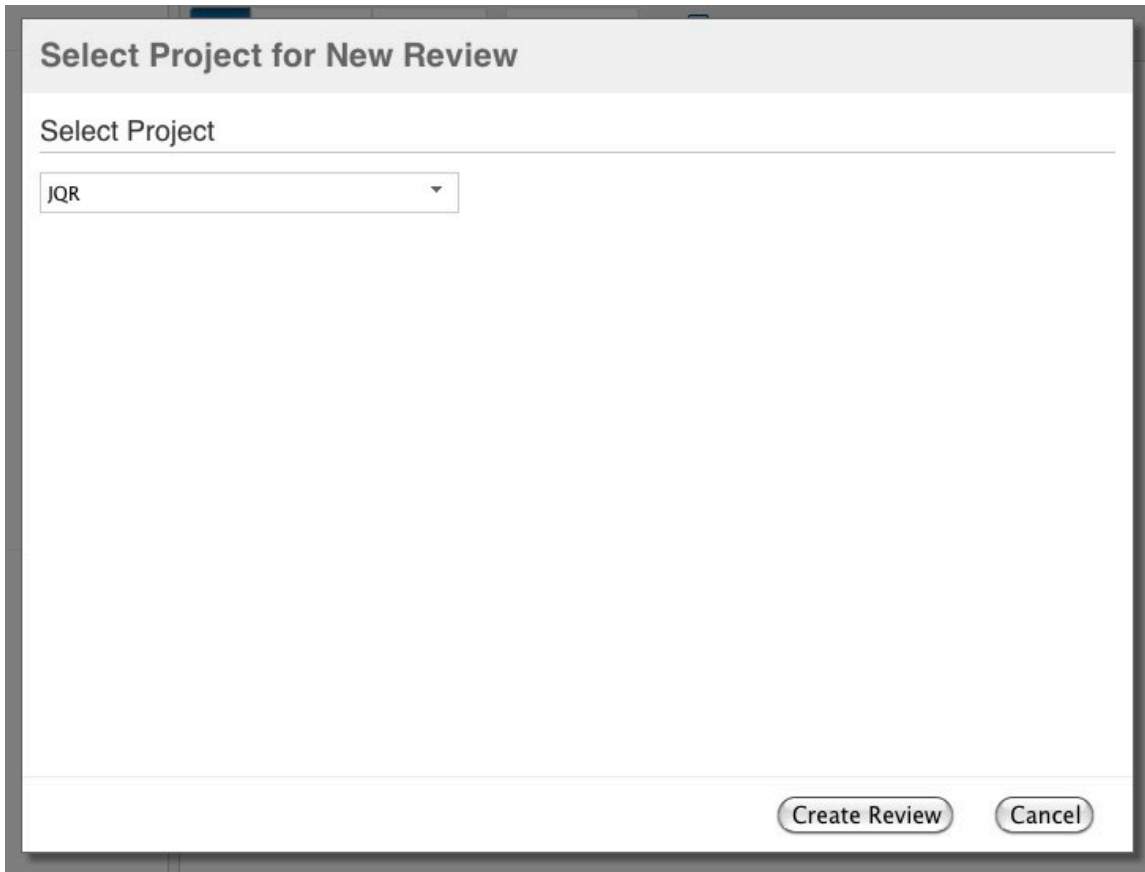
Choose the project in which you want to create the review.

In the next screen click on **Browse Changesets** to see the list of changesets available to create the review.

Select the changesets that you would like to be reviewed and then click on **Edit Details** to add reviewers.

Once you have chosen your reviewers and updated the review information you can click on **Start Review** to begin the review process.
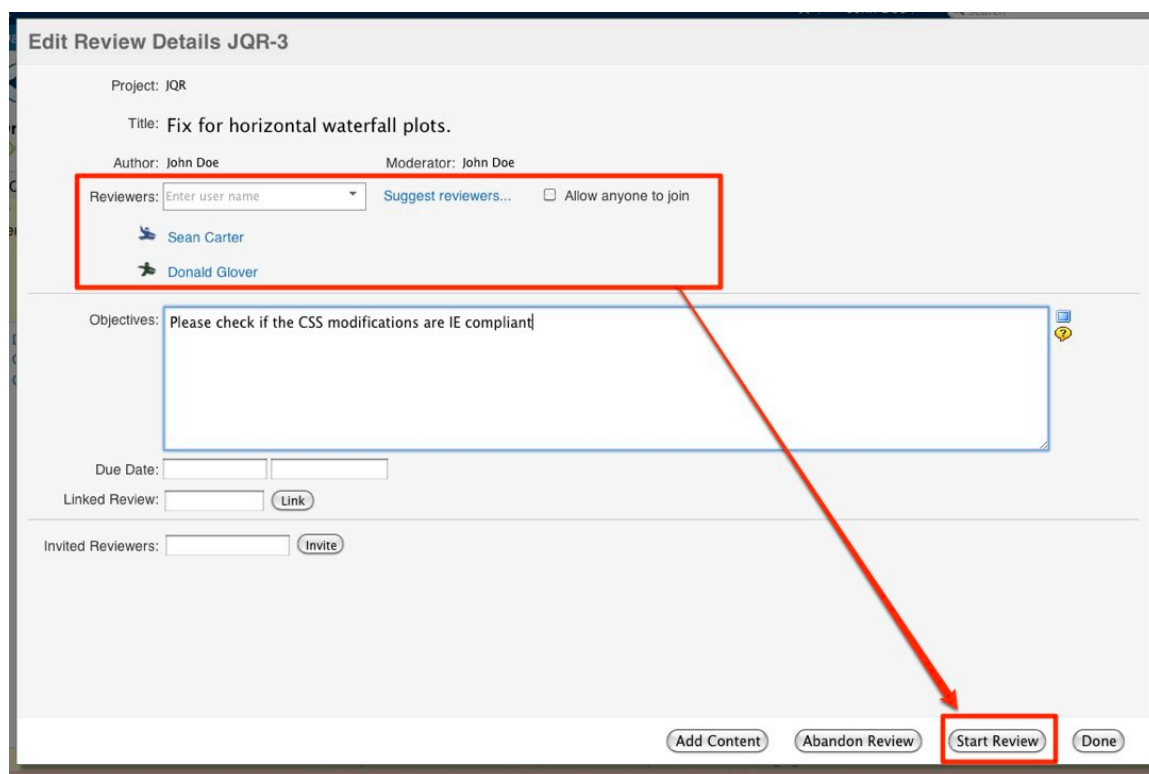
The review is now created and the reviewers will have been notified that a review is pending.

In order to close a review where you are the moderator you need to click on **Summarize** in the top right and then close the review from the modal window.

Using Crucible

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

- The Crucible workflow
- Using the Crucible Screens
    - Browsing All Reviews
    - Browsing Source Files

The Crucible workflow

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

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

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

- 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.

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

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

On this page:

Diagram: Workflow for One-to-One Reviews

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.

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

1. The author starts the review

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

Screenshot: Opening a review from the FishEye Source view
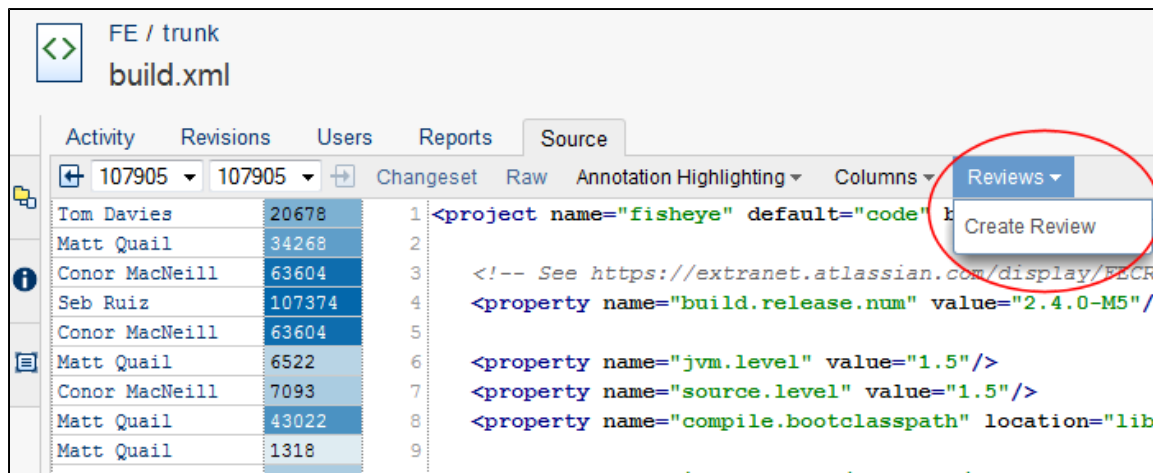
From the FishEye Source view, the author chooses **Reviews** > **Create Review**. If there are multiple projects, the Select Project dialogue opens.

Screenshot: The Select Project dialogue
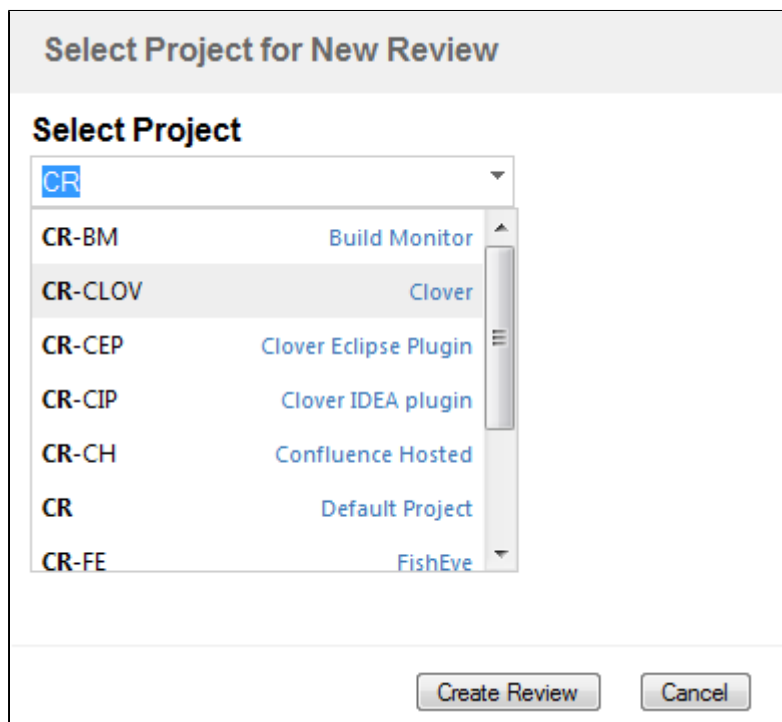
In the Select Project dialogue, you are prompted to choose a project for this review from the drop-down list. Once the selection is made, the author clicks **Create Review**. The Edit Review Details dialogue opens, where the author can create and issue the review.

Screenshot: Creating a review in the Edit Review Details dialogue

In the Edit Review dialogue, the author enters information needed for the review. This includes entering a title and description for the review, selecting reviewers, a due date and the key for a related JIRA issue (if any). The project and author are pre-selected.

The author can also add more content to the review, if they wish, by clicking **Add Content**. See Selecting the Files for the Review.

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

Screenshot: A new Crucible review

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

Screenshot: A Crucible review notification email
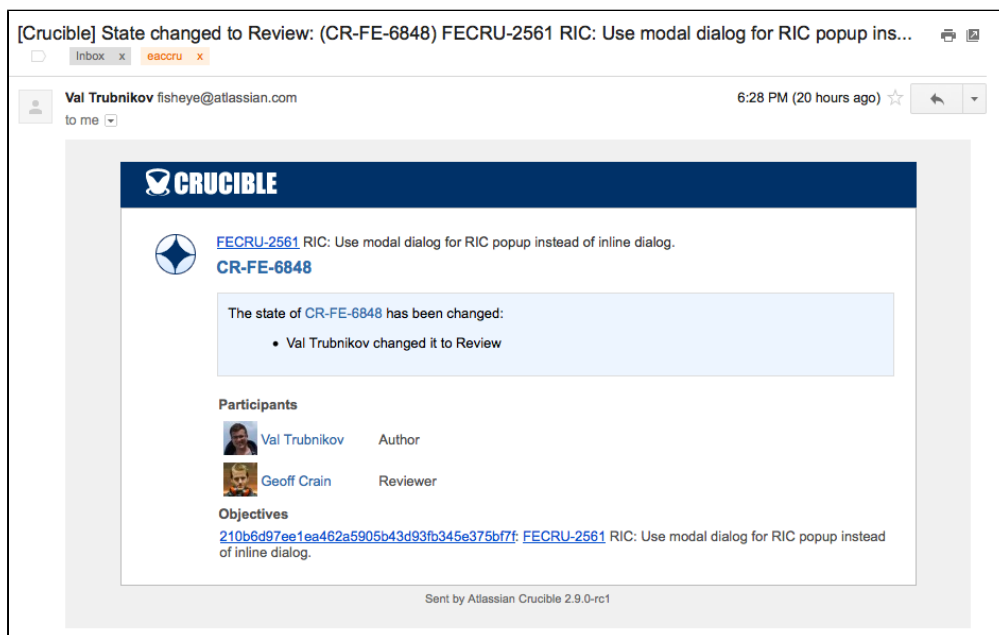
When the reviewer clicks the link in the notification email, the Crucible Review screen opens.

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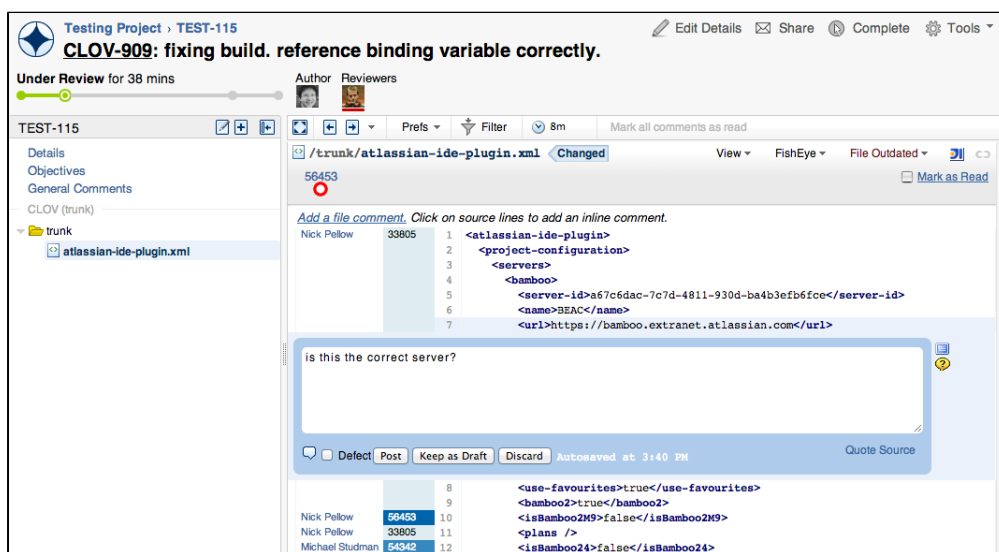
On the Crucible Review screen, the code changes under review are displayed. The reviewer clicks file names to expand the code for in-line reviewing. As the reviewer reads the changes, they can simply click on any line to enter a comment there (multiple lines can be selected by clicking and dragging).

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

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

Screenshot: Comment threads in Crucible
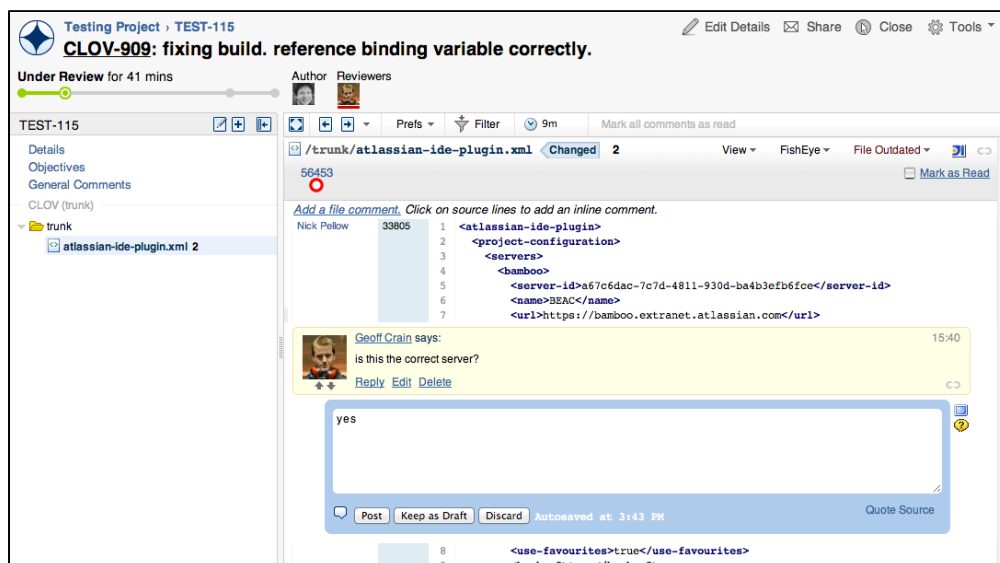
4. The author closes the review

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 will then add any final comments, then click Close when finished.

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

Screenshot: Viewing a closed review
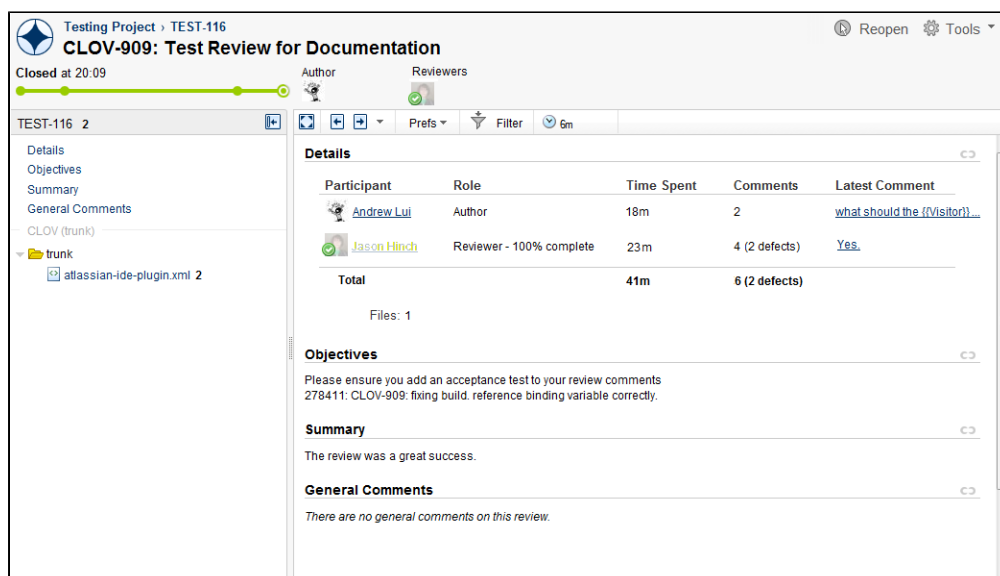
If the author ever needs to resume work on the closed review, they can simply click **Reopen** when viewing this

screen. Doing this will return the review's status to "open".

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

Using the Crucible Screens

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

On this page:

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

Tour of the Crucible Interface

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

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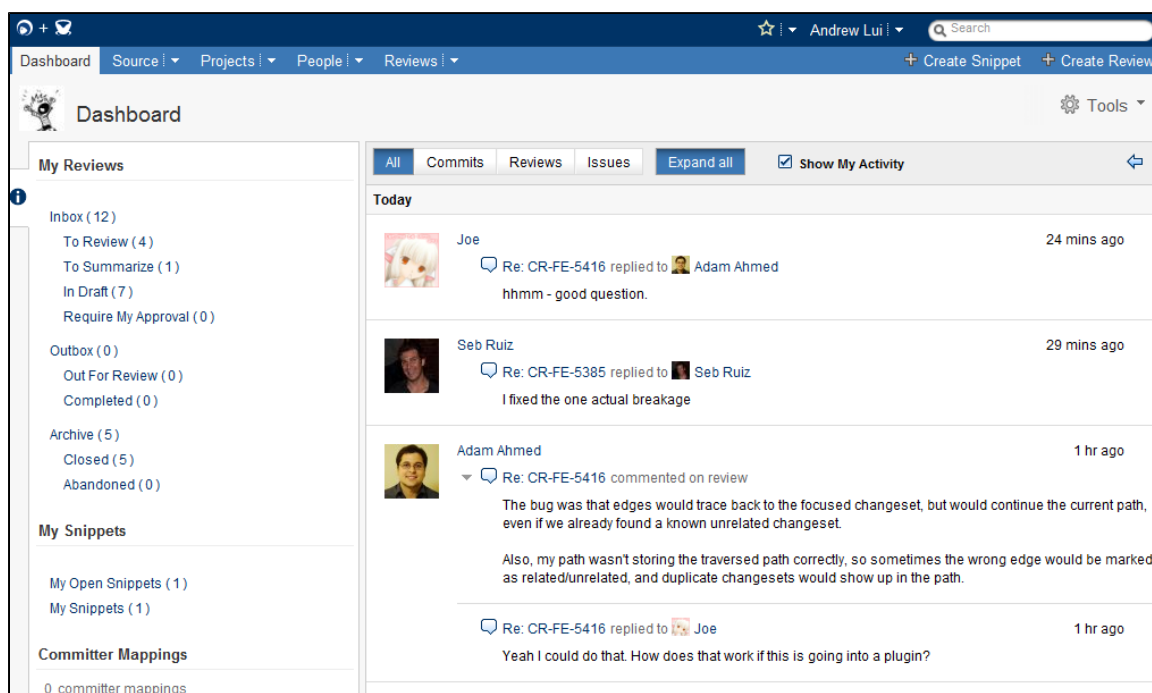
The table below explain the top-level tabs in the Crucible User Interface. Click on the name of a tab for more information.

Element name	Function	Appears
Dashboard Tab	Displays reviews and system activity related to you.	All screens.
Source Tab	Displays contents of connected source repositories.	Only when FishEye is used with Crucible.

Source Dropdown Menu	Displays a list of links to repositories recently visited by the user	Only when FishEye is used with Crucible for logged-in users.
Projects Tab	Displays reviews and content from specific projects.	All screens.
Projects Dropdown Menu	Displays a list of links to projects recently visited by the user	All screens for logged-in users.
People Tab	Displays metrics on the users of the Crucible instance.	All screens.
People Dropdown Menu	Displays a list of links to user pages recently visited by the user	All screens for logged-in users.
Reviews Tab	Allows you to search and report on reviews.	All screens.
People Dropdown Menu	Displays a list of links to reviews recently visited by the user as well as links to the Crucible Inbox and Outbox	All screens for logged-in users.

Left Navigation Sidebar

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

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

Related Links

Browsing 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

The instructions on this page describe how to browse all reviews on the 'Reviews' screen, by people or for projects that you have selected as favourites. This includes reviews that you are involved with. You can also generate reports on review blockers for all people.

The 'Reviews' tab is essentially a shortcut to viewing your reviews (or a custom filtered list, if you have set up a filter). You can also view all reviews filtered by different statuses, e.g. Open, Closed.

To browse all reviews:

1. Click the arrow next to the '**Reviews**' tab at the top of the screen, and click '**All Reviews**' in the dropdown menu. The 'Reviews' page will be displayed, showing all open reviews ('All Open Reviews' will be highlighted in the sidebar) unless you have previously used a custom filter. See the screenshot below.

2. Browse your reviews, by clicking the links under the 'My Reviews' and 'My Snippets' sections in the sidebar.
3. Browse reviews for all people by clicking the links under the '**Everyone's Reviews**' and '**Everyone's Snippets**' sections in the sidebar, as follows:

    ℹ️ *Note, clicking any of the links below will redirect you to the '**Reviews'** tab on the* [Dashboard](). *You can click the '**Reviews'** tab at the top of the screen, if you need to return to this (e.g. to access the 'Reports' tab).*

    *   '**Everyone's Reviews**' — Click '**See reviews by state...**' to expand the list of review categories in the sidebar. You can then click any of the links to view the relevant list of reviews:
        *   '**All Open Reviews**' — Click to view all open reviews, i.e. reviews that have not been [summarised and closed]() yet.
        *   '**All Closed Reviews**' — Click to view all closed reviews, i.e. reviews that have been [summarised and closed]().
        *   '**All Reviews**' — Click to view all reviews, including open reviews, closed reviews and draft reviews.
    *   '**Everyone's Snippets**' — Click '**See snippets by state...**' to expand the list of snippet categories in the sidebar. You can then click any of the links to view the relevant list of snippet reviews, similar to the links described above. See [Browsing All Reviews]()for more information.
        *   '**All Open Snippets**' — Click to show all open [snippets]().
        *   '**All Snippets**' — Click to show all [snippets](), i.e. open and closed snippets.

Screenshot: Browsing all Reviews on the 'Reviews' screen

Notes

Related Topics

[Using the Dashboard]()
[Viewing Reports]()

Browsing Source Files

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

To browse source files:

1. Click the '**Source**' tab. 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 desired 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. Click the file that
   you want to view. The file will be displayed in the main panel. See the 'Viewing a file' screenshot below.
4. You can view various information about the file, as outlined in the table below:

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

ⓘ To download files, firstly click through the desired file. From there, you will see a control bar directly

above the code content which contains the '**FishEye**' item. Clicking this leads to a drop-down menu where '**Download Raw File**' is available. You can use this to download the file in context only.

Screenshot: Viewing all repositories

Screenshot: Viewing a repository
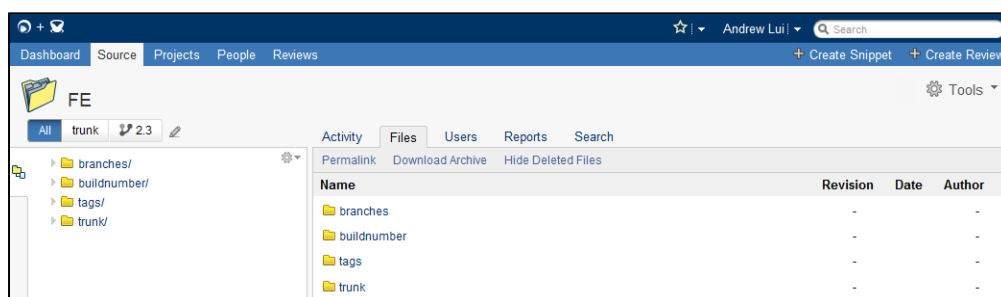
Screenshot: Viewing a file
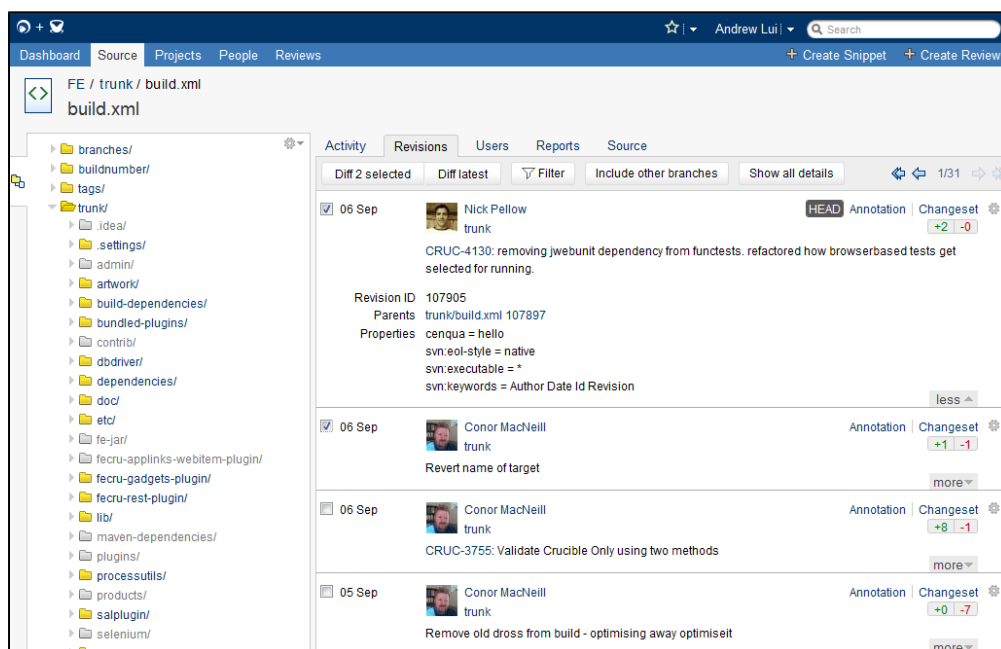
Using the Dashboard

The Dashboard is the first screen that you see when you log into FishEye/Crucible. It is the home page for the instance and can be accessed by clicking the FishEye/Crucible icon , or by clicking **Dashboard** at the top.

The Dashboard sidebar displays your reviews, and activity is displayed in the main pane.

Screenshot: Viewing the Dashboard
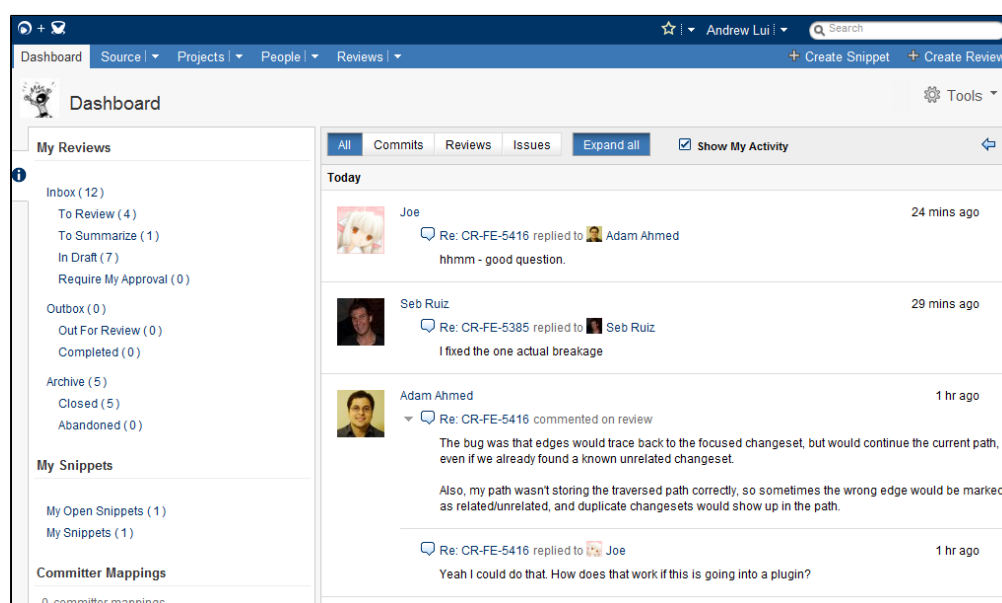
Using the 'My Reviews' sidebar

The 'My Reviews' sidebar contains information about reviews/snippets that you are involved with, e.g. how many reviews require your approval. Read more about the sidebar in Browsing All Reviews.

- Click on any of the links in the sidebar to navigate to that information in the 'Reviews' tab of the Dashboard.
- Hover your mouse over the sidebar and click the collapse icon (▲) to hide any of the sections. Click the expand icon (▼) to expand any collapsed sections.
- Click the 'i' icon (ⓘ) to hide/show the entire sidebar.

Inbox	Click to show all reviews in 'To Review', 'To Summarize', 'In Draft' and 'Require My Approval' states: • **To Review** — Click to show all reviews where you are a reviewer and haven't yet completed your review work. • **To Summarize** — Click to show all reviews where you are a moderator and haven't yet summarised and closed the review. • **In Draft** — Click to show all reviews that you have created but have not yet been moved to the 'Approval' state or the 'Require Approval' state. • **Require My Approval** — Click to show all reviews where you are a moderator and need to approve the review.

Outbox	Click to show all reviews in 'Out for Review' and 'Completed' <u>states</u>"
	• **Out for Review** — Click to show all reviews that you are a participant of, that have review work that is yet to be completed by other reviewers.
	• **Completed** — Click to show all reviews that you are a participant of, and have been <u>completed</u>.
Archive	Click to show all reviews in 'Closed' and 'Abandoned' <u>states</u>:
	• **Closed** — Click to show all reviews that you are a participant of, that have been <u>summarised and closed</u>.
	• **Abandoned** — Click to show all reviews that you are a participant of, that have been abandoned. You may wish to delete these reviews.
My Open Snippets	Click to show all open <u>snippets</u> created by you.
My Snippets	Click to show all <u>snippets</u> created by you.

Using the Activity Stream

The Dashboard has an activity stream that displays the following information:

- **reviews activity** — This includes the addition of review comments, opening and closing reviews, etc.
- **source activity** — This includes files being committed to a repository.
- **issues activity** — If you have <u>linked a JIRA server with your Crucible project</u>, the activity stream will also include updates to linked JIRA issues.

Your activity stream will only display information from projects, reviews, people, repositories, etc, that you have selected as favourites as well as your own activity. For more information on favourites, see <u>Using Favourites</u>.

The following instructions describe how to view the activity stream and filter it to display specific information.

- **Commits** tab — See <u>Browsing Source Activity only</u> below.
- **Reviews** tab — See <u>Browsing Reviews Activity only</u> below.
  Issues tab — This tab will only display if you have connected a JIRA instance to your FishEye/Crucible instance. See <u>Browsing Issues Activity only</u> below.
- **Show My Activity** checkbox — Untick this checkbox to filter out your activity from the activity stream displayed. This toggle can be used in any of the tabs for the activity stream.

Browsing Source Activity only

Source activity includes files commits to repositories that you have selected as <u>favourites</u>.

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

Browsing Source Activity on the Dashboard

Browsing Reviews Activity only

Reviews activity includes updates to reviews in all projects that you have selected as favourites. 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).

Browsing Reviews Activity on the Dashboard

Browsing Issues Activity only

Issues activity includes updates to issues in JIRA projects that are associated with your favourite Crucible projects. For more information about integrating JIRA with Crucible, see JIRA integration in Crucible.

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

Browsing Issues Activity on the Dashboard

Watching an Activity Stream

You can "watch" an activity stream in FishEye/Crucible. Watching the activity stream allows you to receive emails when updates occur in the activity stream. You can view all of your watches and configure the frequency of your watch emails in your user profile. See Changing your User Profile for more information.

Note, the option to add a watch will only be available if the administrator has enabled watches for the repository.

To watch an activity stream:

1. Navigate to the activity stream that you want to watch.
2. Click the **Tools** menu and click **Watch**. The page will reload and a watch will be set up for the activity stream (the watch icon will be coloured, not grey).
    - If you want to remove the watch, navigate the activity stream, click **Tools** then **Watch**. The watch will be removed (the watch icon will be coloured, not grey).
      You can also remove watches via your user profile.

Notes

Related Topics

Browsing All Reviews
Using Favourites
Changing your User Profile

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

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.

> **On this page:**
>
> - [Using the Quick Search]
> - [Filtering Reviews]
> - [Searching for Review Comments]
> - [Notes]

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 criteria in the search box in the Crucible header (Quick Nav). Crucible offers a number of parameters and functions that you can use to refine your expected results, see Refining your Quick Search Criteria below.
2. "Quick Nav" results will appear in a dropdown, as you type. "Quick Nav" will attempt to match against the review name, project and user.
   - If you want to use a quick nav result, use the up- and down-arrows on your keyboard and press enter or use your mouse to select the item.
   - If the quick nav 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.
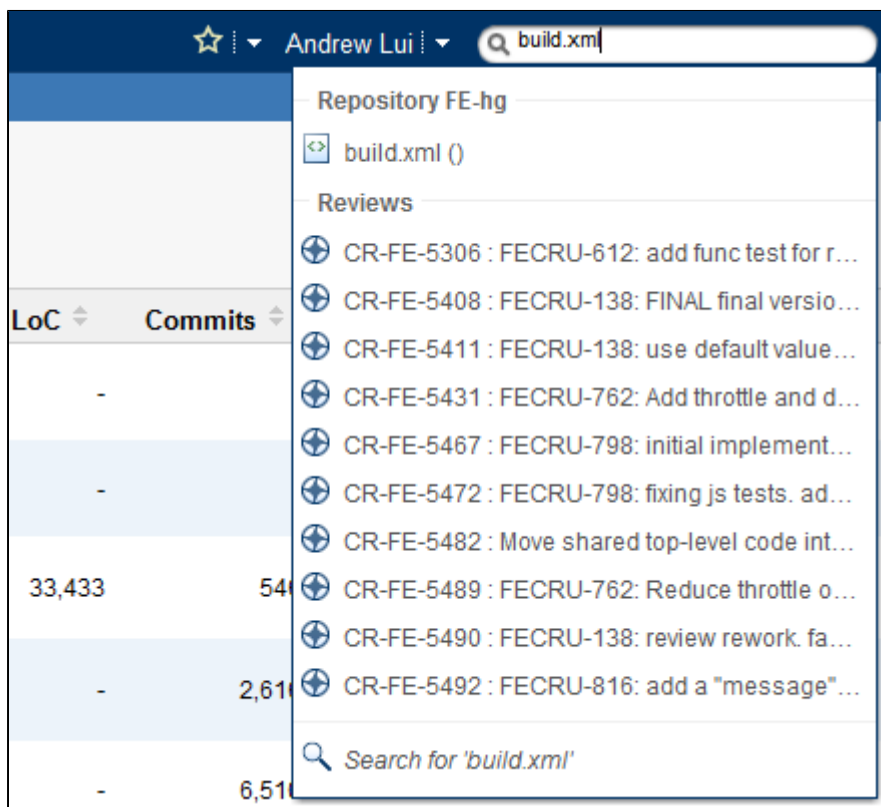
Screenshot above: Quick Search displaying "quick nav" matches

Refining your Quick Search Criteria

The Crucible Quick Search has a number of powerful tools that you can use to 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-28 189`.

Filtering Quick Search Results

Once you have a set of search results on the Quick Search page, you can filter them to a subset of the original results. The filter controls are in the left panel of the Quick Search page in the 'Source' section.

Filter	Description
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.

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. Navigate to the '**Reviews**' tab.
2. Click the list of reviews that you want to start with, in the reviews sidebar of the 'Review Dashboard', e.g. 'All Open Reviews'.

3. Click the '**Custom Filter**' in the reviews sidebar.
4. Update the filters with your search criteria (see table below) and click '**Apply Filter**' to filter the reviews.

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.
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. Navigate to the '**Reviews**' tab.
2. Enter your search criteria 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 contains the specified text.
Review PermaId	Find comments made on the specified review.
After	Find comments made after after a particular date.
Before	Find comments made after before a particular date.

Comment Author	Find comments made by a particular user.
Search Type	You can:

- Tick the '**Defects**' check-box to find comments that are flagged as Defects.
- Tick the '**Comments**' check-box to find comments that are not flagged as Defects.
- Tick neither check-box (or both of them) to find all . ||

Review State	Find comments on reviews that are in a particular state. See *Review State Filter* (above).
Ranking	Find defects have been given a particular ranking (e.g. 'Major', 'Minor').
Requires Re-Review	Find defects that have been marked as requiring re-review (or not).
Classification	Find defects that have been given a particular classification (e.g. 'Missing', 'Ambiguous').

Screenshot above: Search Comment Filter Options

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.
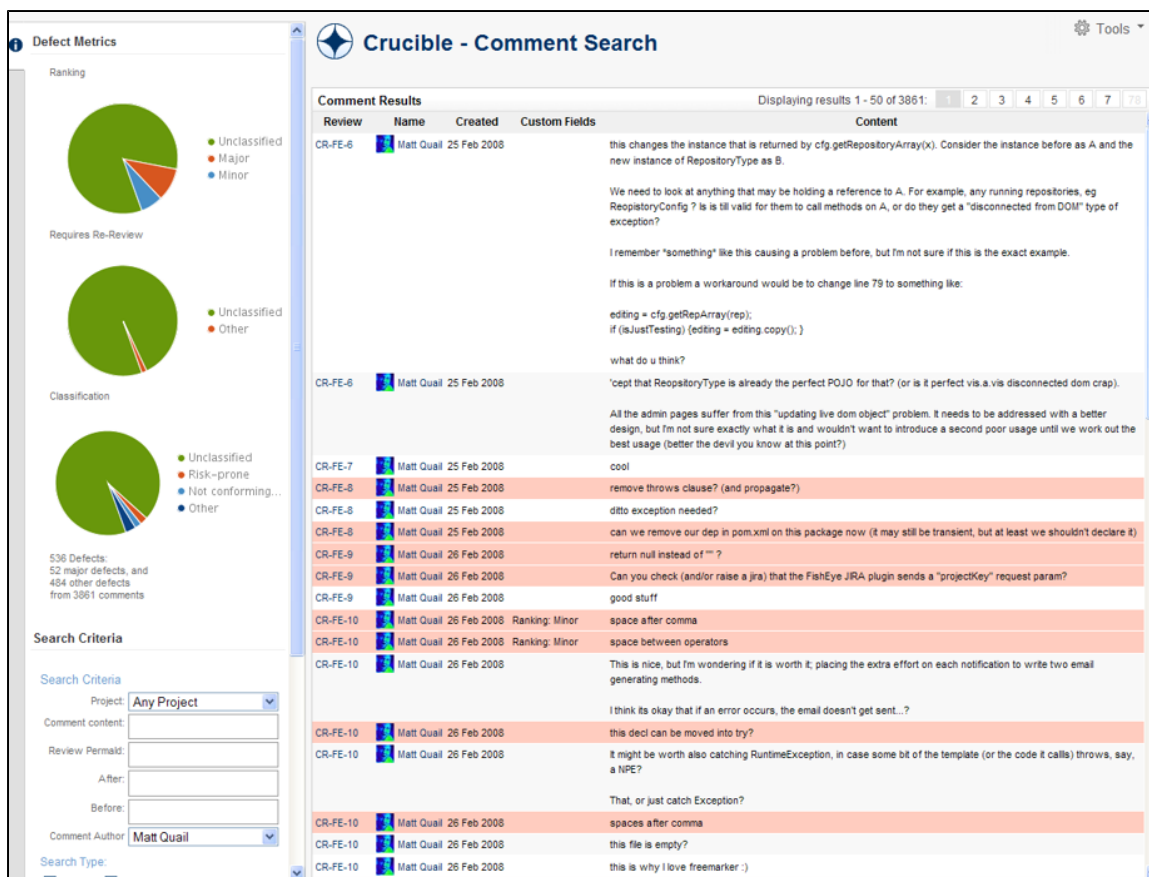
Screenshot above: Comment Search Results

Notes

Related Topics

Searching FishEye *(FishEye documentation)*

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)
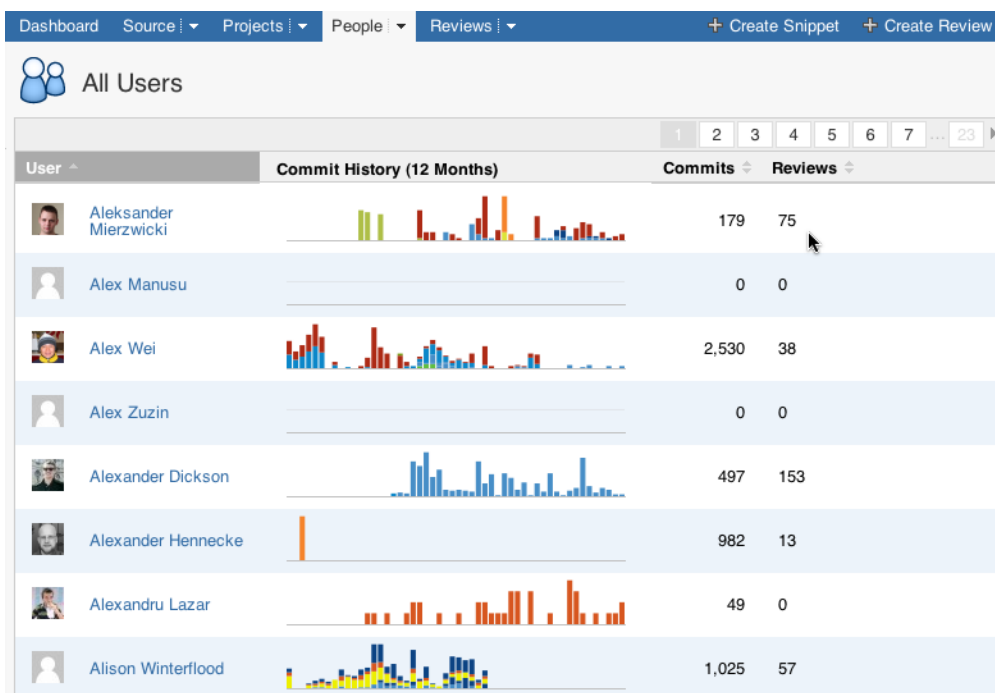
Viewing a Person's Activity Screen

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 favourites.

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
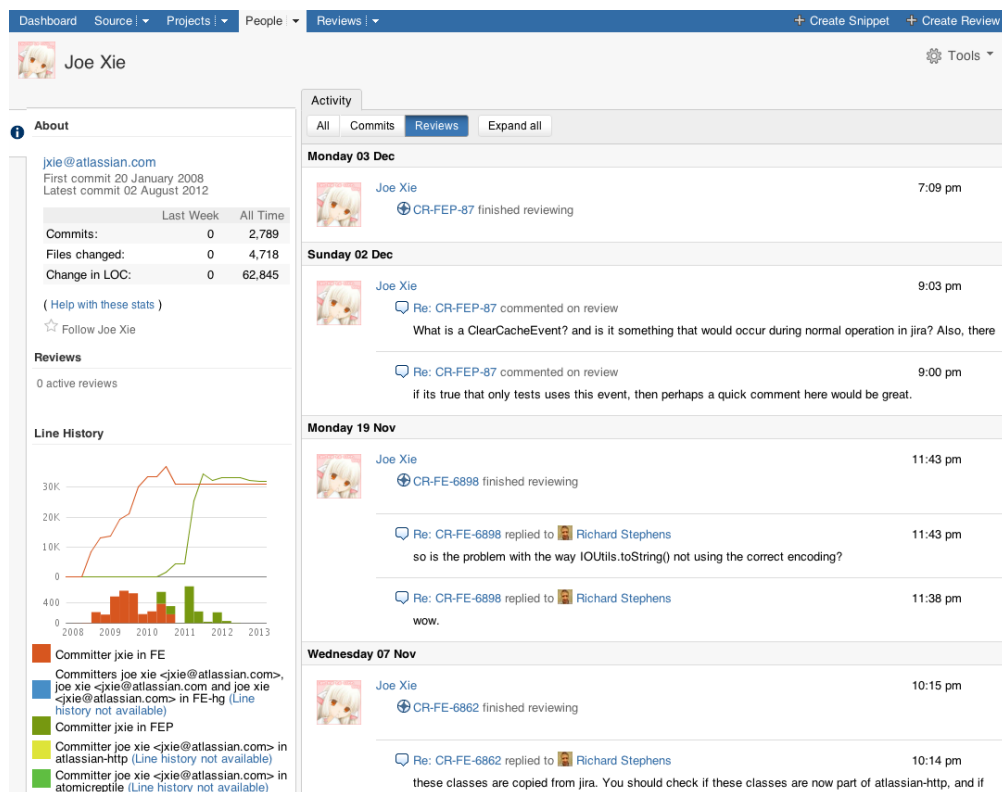
> ℹ️ Some users may not appear to have the correct number of Files Changed or LOC, despite regularly committing. In this situation, if they have committed to a directory which is not covered by the regexes in your symbolic definition (i.e. they have committed to a directory that is neither trunk, branches 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

	• **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.

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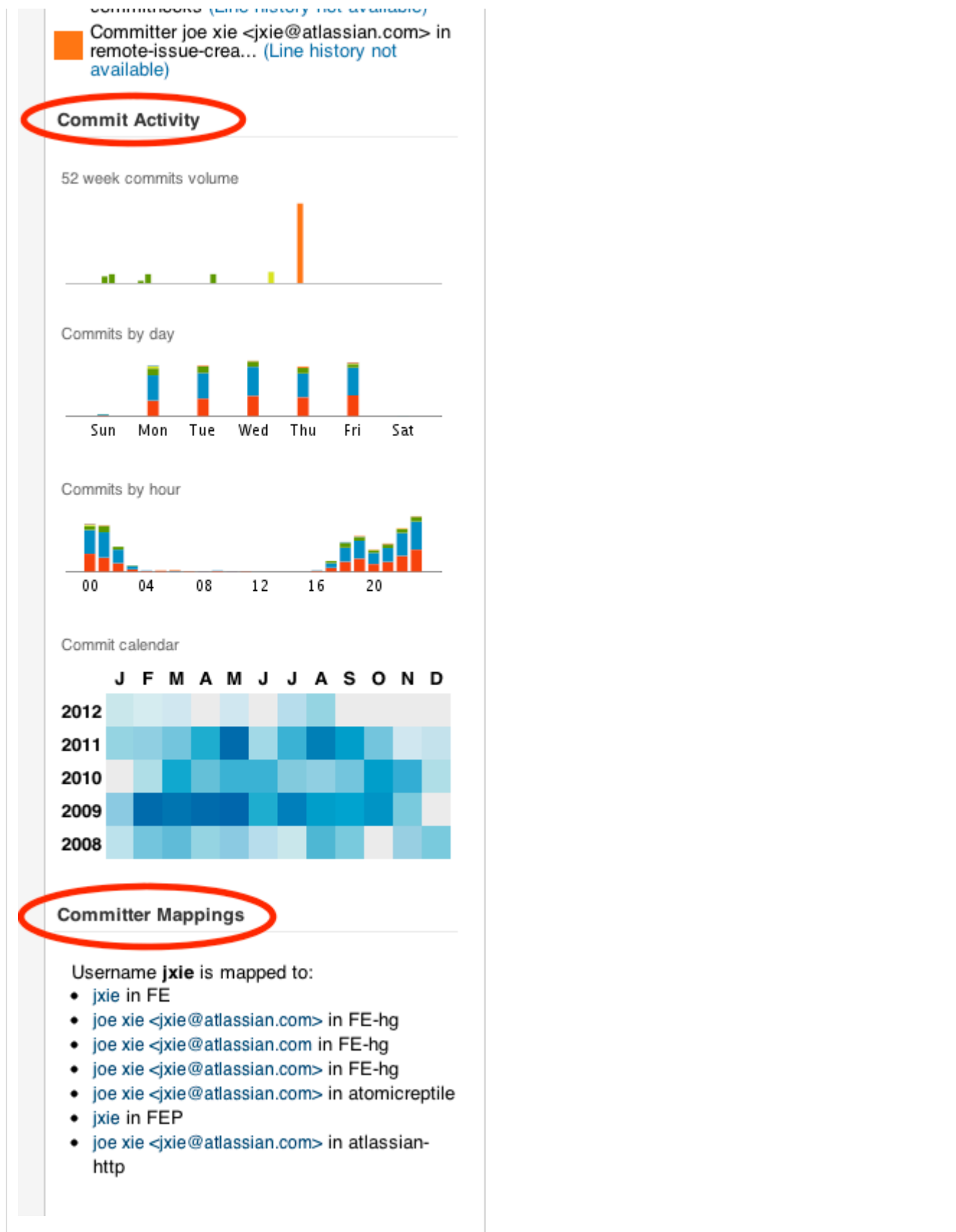
- **Line History**
  The Line History section shows a graph with the number of lines committed to the repository, charted over time.

- **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 joe xie <jxie@atlassian.com> in
remote-issue-crea... (Line history not
available)

Commit Activity

52 week commits volume

Commits by day

Sun   Mon   Tue   Wed   Thu   Fri   Sat

Commits by hour

00   04   08   12   16   20

Commit calendar

Committer Mappings
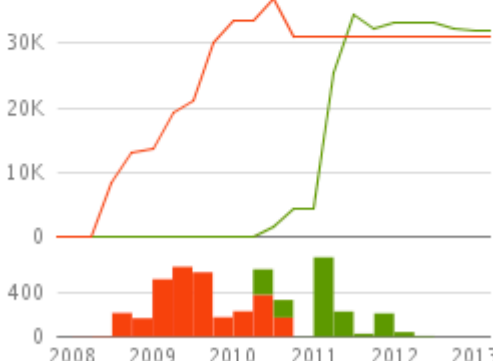
Username **jxie** is mapped to:
- jxie in FE
- joe xie <jxie@atlassian.com> in FE-hg
- joe xie <jxie@atlassian.com in FE-hg
- joe xie <jxie@atlassian.com> in FE-hg
- joe xie <jxie@atlassian.com> in atomicreptile
- jxie in FEP
- joe xie <jxie@atlassian.com> in atlassian-
  http

Browsing Projects

To browse the content in a project, click the Projects tab at the top of the page. The 'Projects' view opens (see 'The Crucible Projects Index' screenshot below).
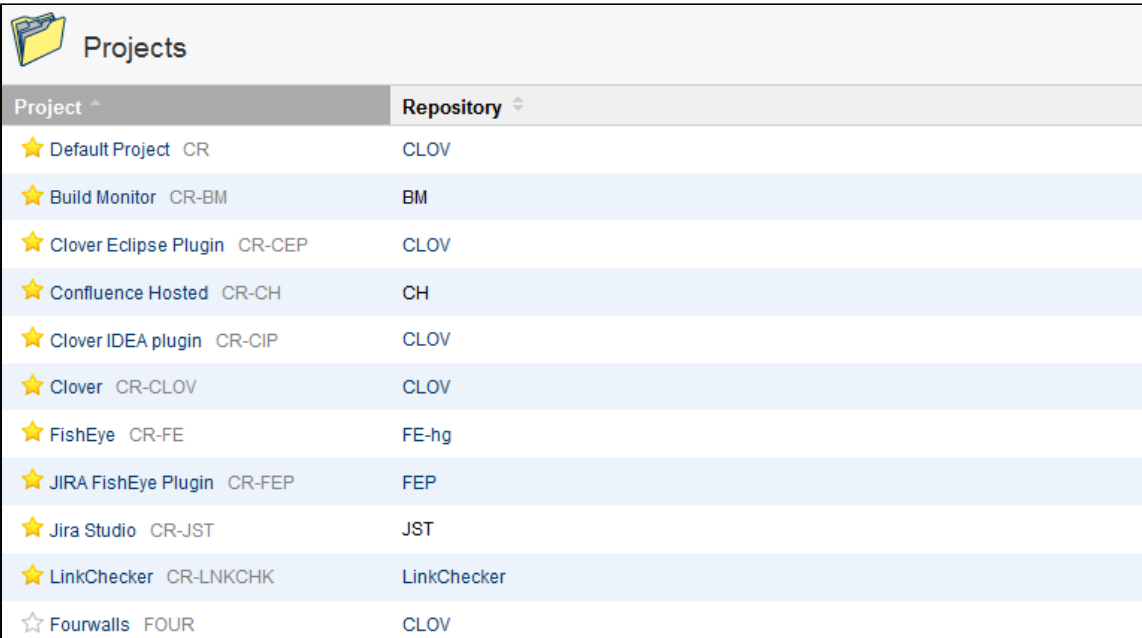
A list of projects will be shown if there is more than one. Click the name of the desired project to open it. The '**Pr oject Activity**' page opens (see 'The Crucible Project View' screenshot below). In the left navigation bar, charts showing overall project statistics are displayed.

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

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

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

Screenshot: The Crucible Projects Index

Screenshot: The Crucible Project View

Viewing Project Statistics

This page explains the layout of the Project Summary page.

On this page:

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

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

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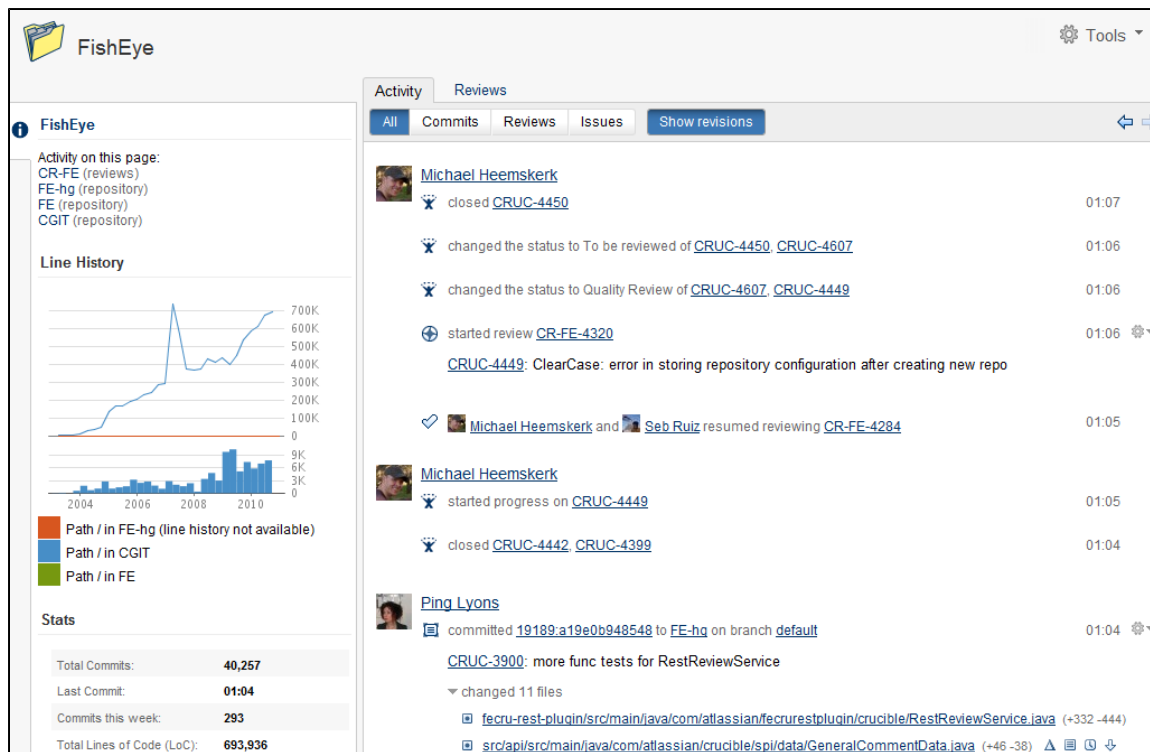
In the right hand pane, you can see an activity stream relating to this project. In the left hand pane, you can see various statistics charts relating to the project in context. These appear in a reduced size until you click them, when they will expand to show more information.

Project Name Panel

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

Project Line History Panel

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

Screenshot: The Project Line History Panel
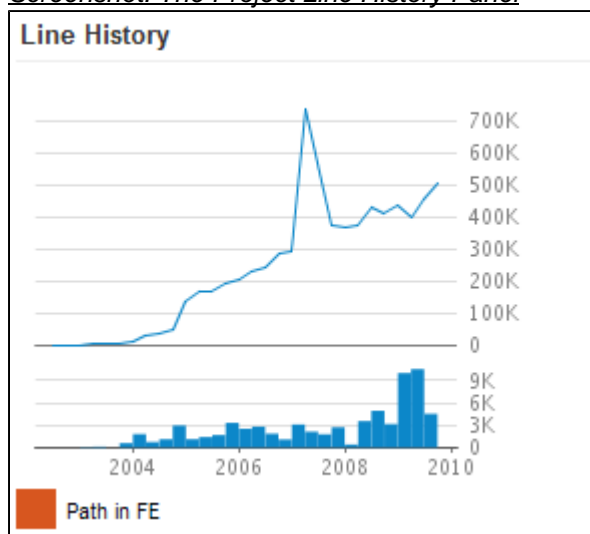
Project Stats Panel
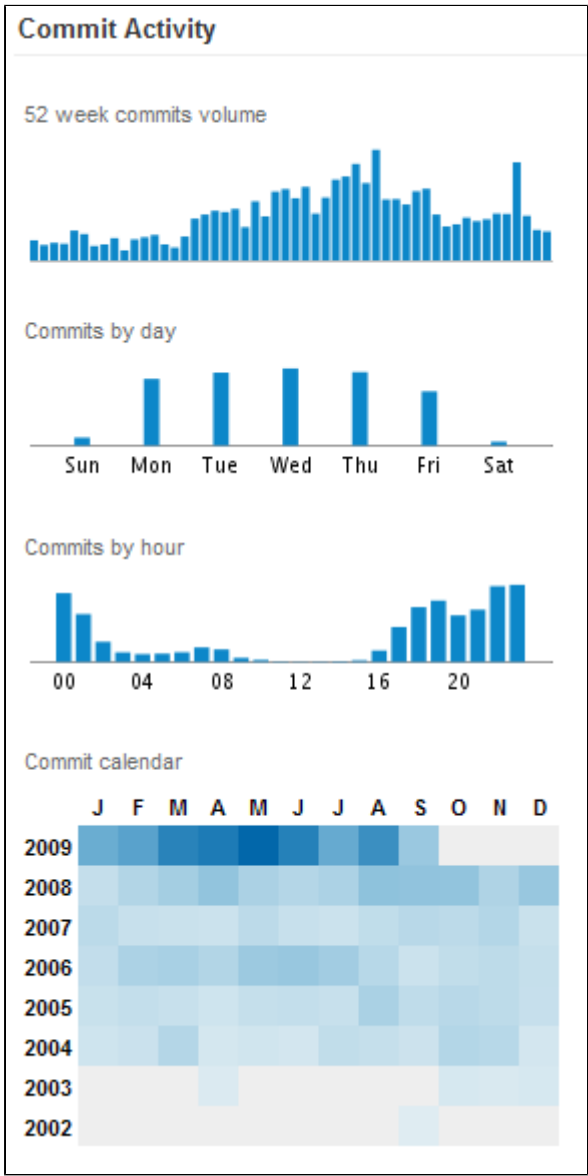
This panel contains a chart showing numerical data for commits, files changed and lines change, graphed over time.

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 colours to indicate more commits) graphed by month, over years that the repository has been running.

Viewing Reports

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

FishEye Reports (on this page):

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

Viewing the 'Review Blockers' report

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

To view the 'Review Blockers' report:

1. Click the dropdown arrow next to the '**Reviews**' tab at the top of the page and select '**Reports**' from the dropdown menu.
2. Click the '**Review Blockers**' link under the '**Reports**' sub-tab. The 'Review Blockers' report will be displayed.
   - Click a **user's name** to go to their 'Activity' screen.
   - Click a number in the '**To Complete**' or '**To Summarize**' column to go to the list of reviews waiting to be completed/summarised by the user.

Screenshot: 'Review Blockers' Report

Review Blockers Report

User	To Complete	To Summarize
Matthew Watson	16	-
Anna Buttfield	4	6
Craig Sharkie	8	2
Felix Schmitz	4	4
Tim Pettersen	2	6
Michael Studman	4	3
Giancarlo Lionetti	3	3
Joe Xie'"	6	-
Andrew Lui	4	1
Conor MacNeill	4	-
Brendan Humphreys	2	1
Don Brown	2	1
Edwin Wong	1	2
Geoff Crain	3	-
Rosie Jameson	-	3
Andrew Myers	-	2
Edwin Dawson	-	2
Jake Causby	1	1
Marek Went	1	1
Matt Quail	2	-
Pierre-Etienne Poirot	-	2
Ross Rowe	-	2
Seb Ruiz	2	-
Slawek Ginter	1	1

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**' from the dropdown menu.
2. Click the '**JIRA Blockers**' link under the '**Reports**' sub-tab.
3. Enter the details of your JIRA server and project, and click the '**Go**' button. The 'JIRA Blockers' report will be displayed with 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

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 in your repository has been reviewed, which files and when. You can also access the reviews.

ℹ️ 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 Statistics Panel
    - Using the Changesets Coverage Panel

Screenshot: The Review Coverage Report
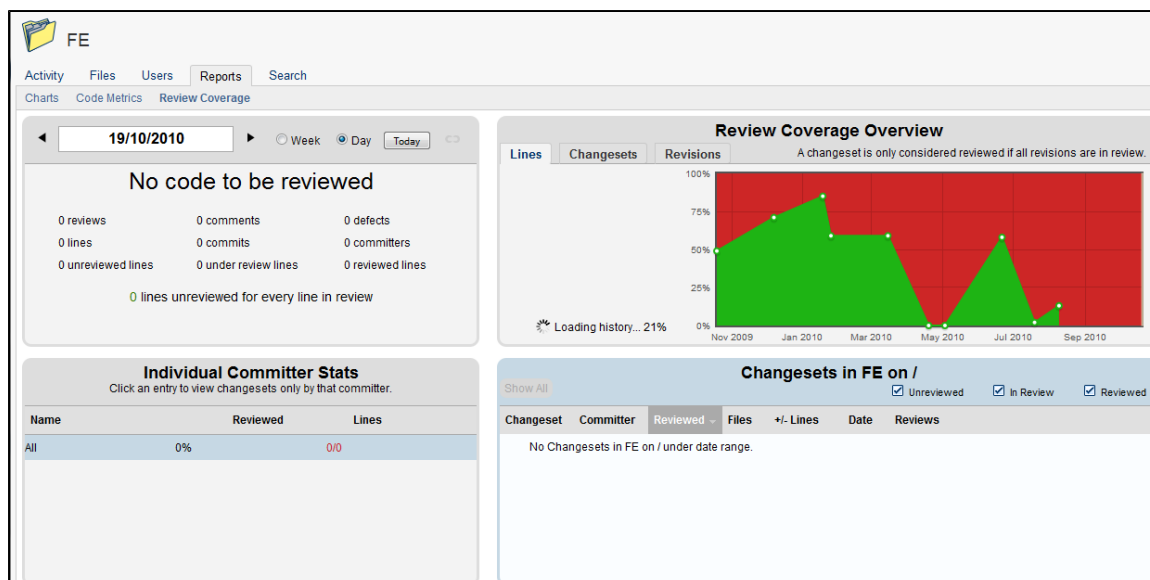
Opening the Review Coverage Report

To open the Review Coverage Report:

1. Click the '**Source**' tab.
2. Select your repository. The repository you've chosen will set the scope for the Coverage Coverage report.
3. If desired, navigate down the tree to the desired path you want to view coverage on.
4. Click the Reports tab in the secondary toolbar.
5. Select '**Review Coverage Report**' from the list of reports in the upper panel.

ℹ️ You can view coverage of any path by navigate 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 shows some choice metrics from your Crucible instance. The following information from your repository is arrayed:

- 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

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
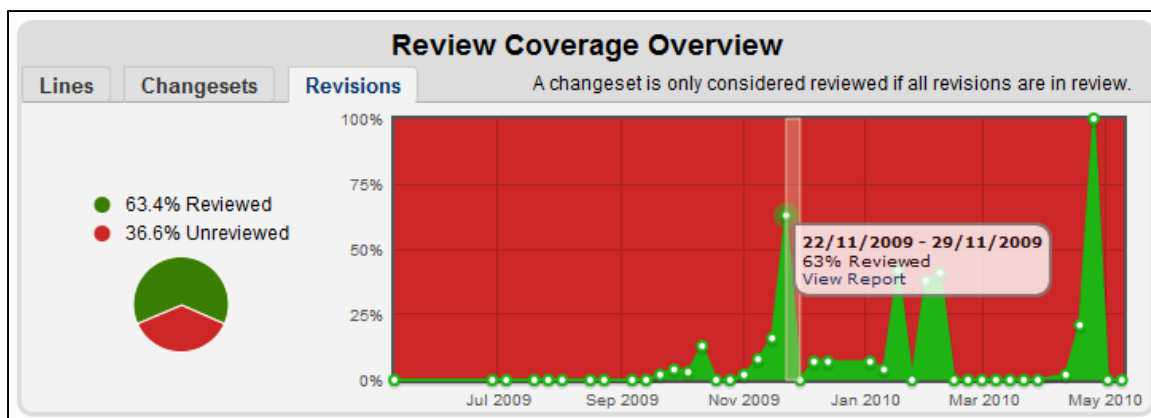
Using the Individual Committer Statistics Panel

The Individual Committer Statistics window 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 Coverage Panel

The Changesets Coverage 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 colour coding to denote review coverage (listed in the table below).

Colour Key

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

Screenshot: Changesets Coverage panel in the Review Coverage Report

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.

ℹ️ 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
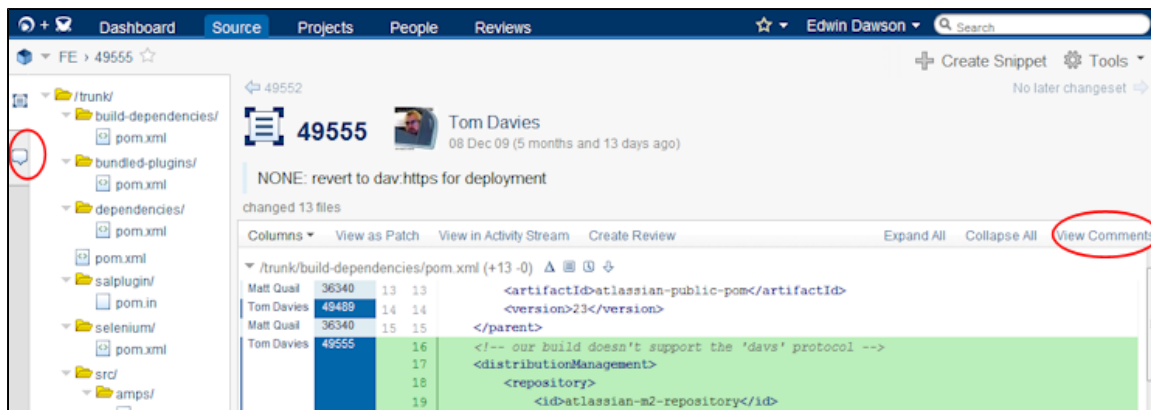
Screenshot: Composing a Changeset Comment
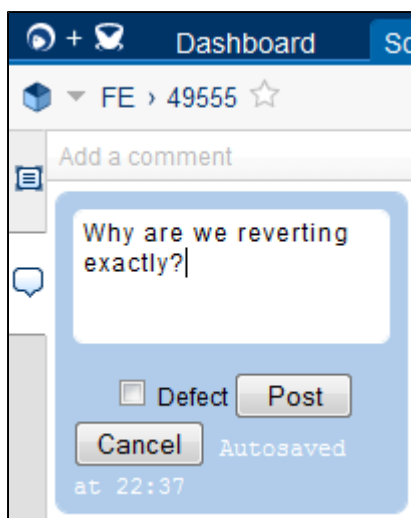
Turning changeset discussions on and off

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.

Changing your User Profile

See Changing your User Profile in the FishEye documentation for instructions on how to change your user profile.

Roles and Status Classifications

This page explains the roles & status classifications in Crucible.

- Roles in Crucible

Roles in Crucible

Author

The page author does not exist.

Creator/Moderator

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

User

A *user* is a person using Crucible.

Status Classifications in Crucible

Draft

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

Under Review

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

Summarized

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

Closed

Closed reviews are complete.

Abandoned

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

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

Conducting a Review

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

- Creating a Review
- Selecting the Files for the Review
- Adding Reviewers
- Issuing a review
- Performing the review
- Summarising and Closing the Review
- Transitioning JIRA issues
- Moving a Review to Another Project
- Deleting a review

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

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

Creating a Review

This page explains how to create a Crucible review.

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

- Creating a Patch Review
- Creating a Review within Crucible
- Creating a review from FishEye
- Creating a review from JIRA
- Creating a Review from a URL
- Creating a Snippet Review

Whichever way you choose, the overall process looks like this:
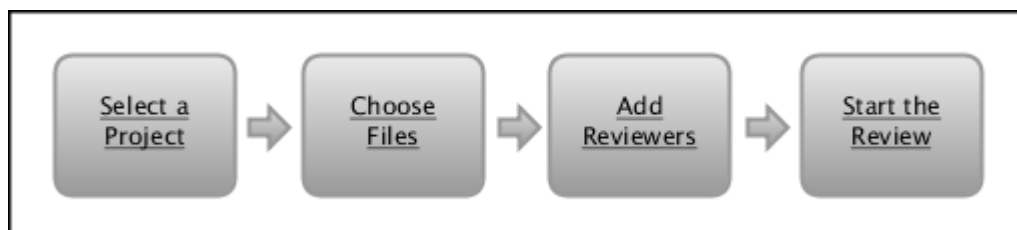
ⓘ Note that only people with the 'Create' permission can create a review.

Review creation considerations

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 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.

This suggests that reviews should be created with care to get the best value from them:

- Avoid overloading the review. Reviews should be focussed on a few necessary files.
- Avoid overcrowding. 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.

Creating a Patch Review

This page includes instructions on uploading patch files from your repository, how to load them into Crucible to be reviewed and use Crucible's Patch Anchoring to retrieve more lines of context from the original file.

On this page:

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

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

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

Using Crucible Patch Anchoring to Automatically Add Full Context

Crucible's Patch Anchoring feature allows you to add a regular patch to a review (showing only a few lines of context. 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 file so that you can view all of the lines of code (greatly enhancing the review process).

To use patch anchoring:

1. Create a new review. From the '**Tools**' menu in Crucible, select '**Create Review**'.
2. Click '**Pre-Commit - Upload a patch file to be reviewed**'. The '**patch upload**' dialog appears. Click '**Browse**', locate your file, then click '**Upload**'. Crucible will now search for matches in the files in its database. Crucible will analyse 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.

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

Screenshot: Crucible Patch Anchoring

1. You can click '**Edit**' to change the anchoring, by choosing a new match or removing the anchor. You can change the anchoring later, after the review is live.
2. Start the review. When viewing the diffs, you will be able to choose more than three lines of context from the '**View**' menu.

Screenshot: Editing Patch Anchoring Settings

*Screenshot: Viewing more than three lines with Patch Anchoring*

Creating a Patch File From Your IDE

Creating a Patch File in IntelliJ IDEA 7.0

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

Select a parent folder, sub-folder or file that you have altered in the Project tool window. Select '**Version Control** ' > '**Create Patch**'. The following window appears:
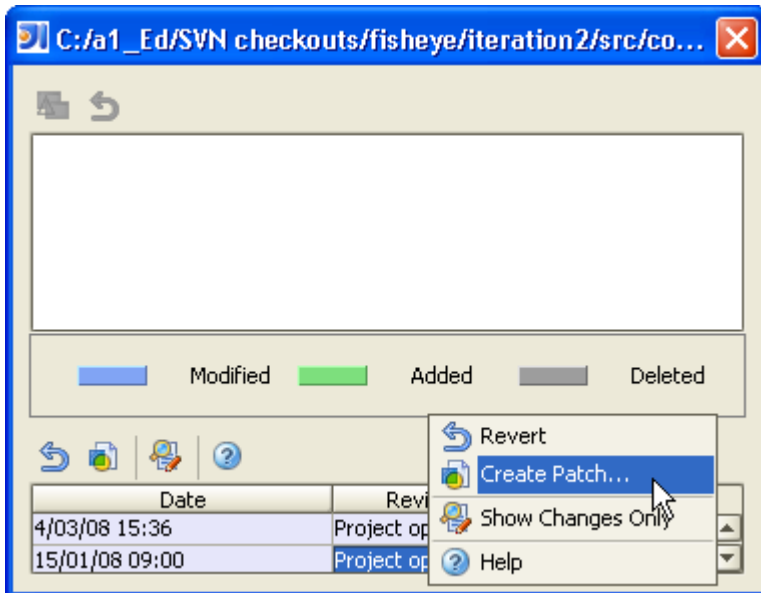
Screenshot: The IDEA Create Patch window
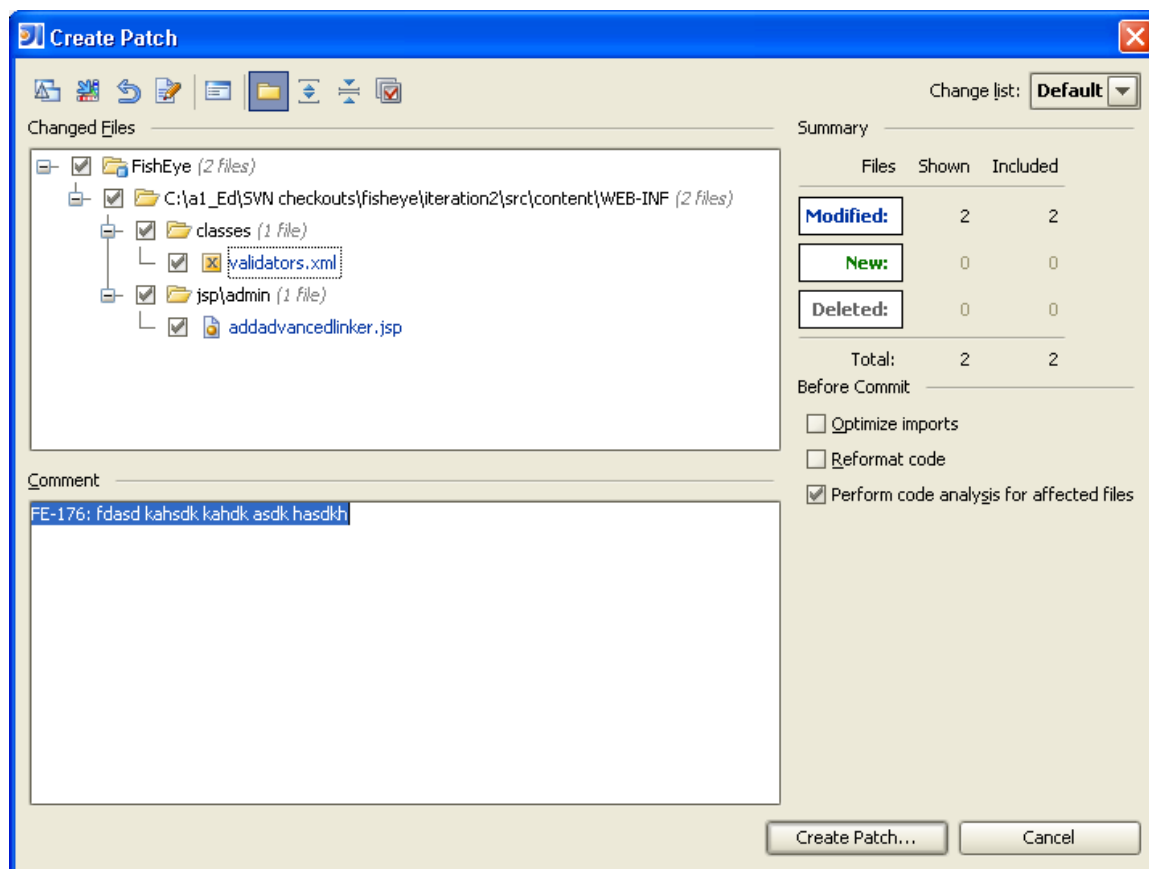
Click '**Create Patch**'. Choose a location to save the patch file and click '**Ok**'.

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

If you have not configured version control in IDEA, you may not have the '**Create Patch**' option available. If so, use the following steps to create a patch file in IDEA:
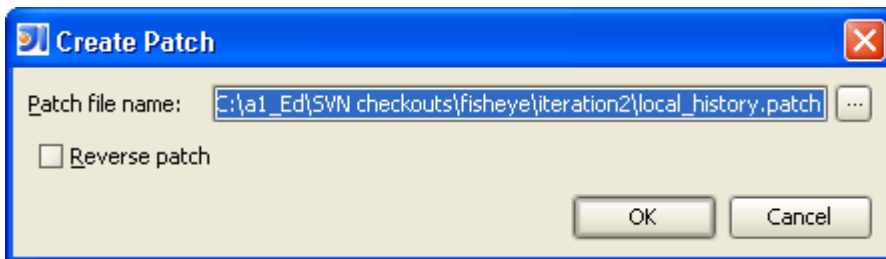
1. Select a parent folder, sub-folder or file that you have altered in the Project tool window, right-click it and choose '**Local History**' > '**Show History**'.

Screenshot: The IDEA Show History dialog

2. In the Local History view, right-click the revision number, and choose '**Create Patch**'.
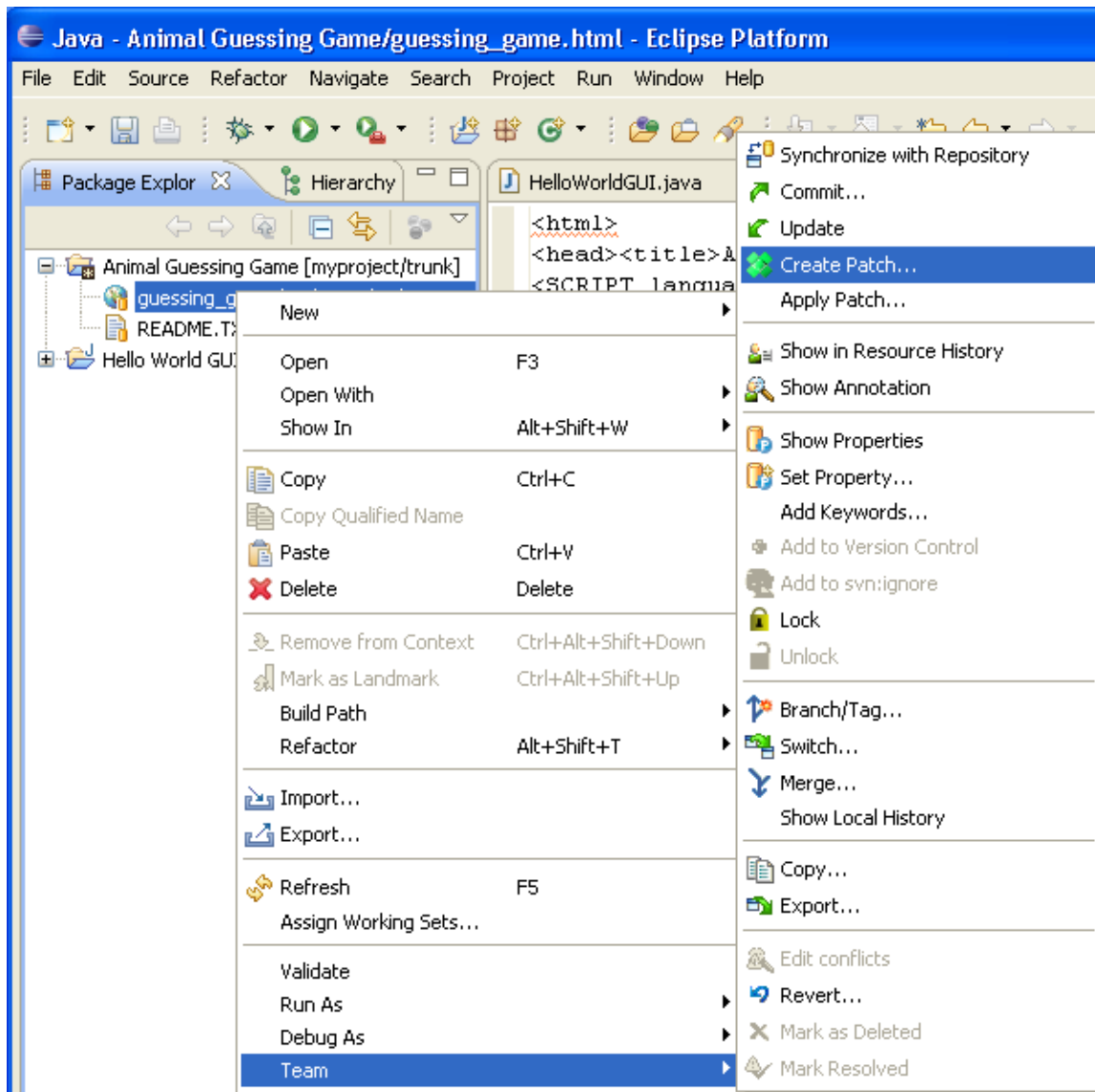
Screenshot: The IDEA Create Patch dialog

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

 Creating a Patch File in Eclipse 3.3.1.1

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

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

Screenshot: Instigating a Patch in Eclipse

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

Screenshot: The Eclipse Create Patch dialog
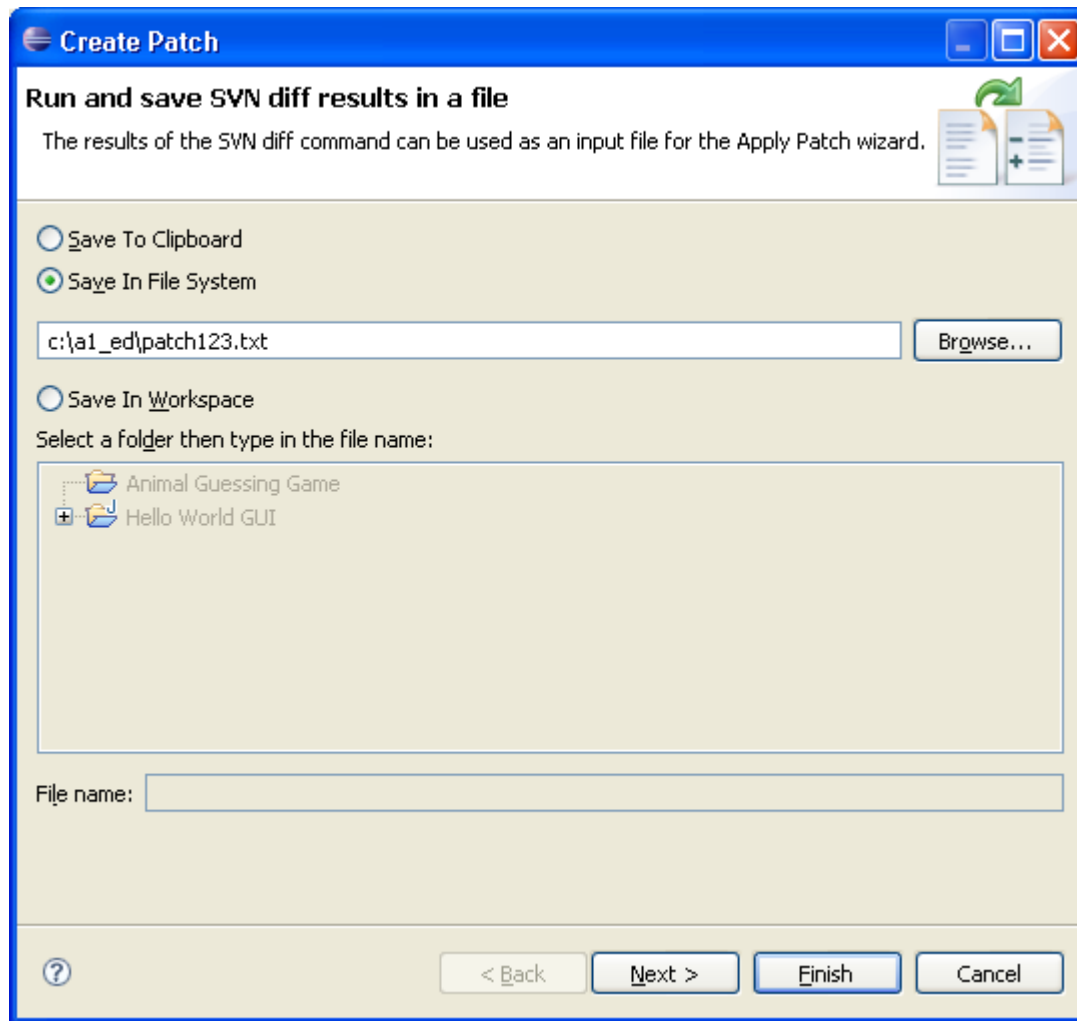
Creating a Basic Patch File From The Repository Command Line

CVS Patch Creation On The Command Line

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

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

Note that patch files created with this command will only include around three lines of code, before and after each change. It will include revision information by default, so if you have FishEye, you may be able to anchor the patch to get full context.

Subversion Patch Creation Via The Command Line

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

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

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

Note that patch files created with this command will only include around three lines of code, before and after each change. It will include revision information by default, so if you have FishEye, you may be able to anchor

the patch to get full context.

Perforce Patch Creation Via The Command Line

Use `p4 diff -dcu` to generate a patch for changed files. For example:

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

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).

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

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

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

Note that patch files created with this command will only include around three lines of code, before and after each change. It will include revision information by default, so if you have FishEye, you may be able to anchor the patch to get full context.

Mercurial Patch Creation Via the Command Line

Use `hg diff` to generate a patch for changed files. For example:

```
hg diff > patch.txt
```

This will include 3 lines of context. It will also include revision information, so if you have FishEye, you may be able to anchor the patch to get full context.

Note: 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 Patch Creation Via the Command Line

Use `git diff` to generate a patch for changed files. For example:

```
git diff > patch.txt
```

This will include 3 lines of context. It will also include revision information, so if you have FishEye, you may be able to anchor the patch to get full context.

Creating Patches That Include All Lines of Code

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

Creating a Patch in CVS With All Lines of code

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

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

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

Creating a Patch in Subversion With All Lines of Code

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

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

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

Creating a Patch in Perforce With All Lines of Code

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

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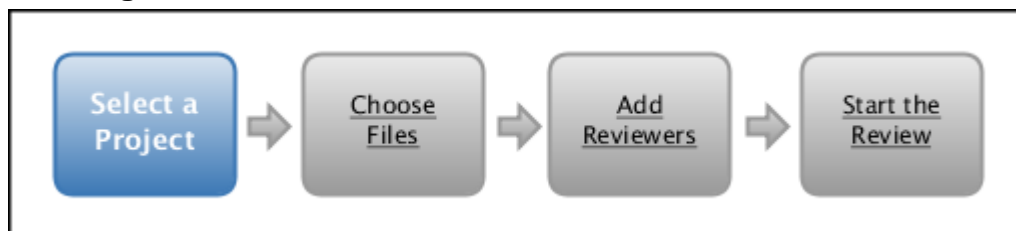
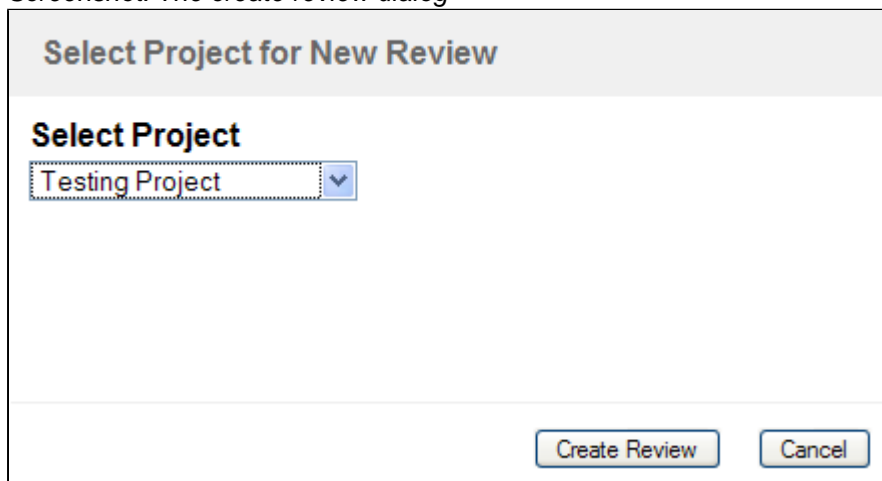This page explains how to create a review from the Crucible interface.

On this page:

- Create a new review
- Choose where your review files will come from
- Add content to the review
- Edit the review details
- Adding an entire directory's contents to a Crucible review

Create a new review

Within Crucible, create a review by opening the **Tools** menu at the top right of the Reviews screen, then clicking **Create Review**. You will be prompted to select the Project for the review (if you have multiple projects). Choose a project and click '**Create Review**'.

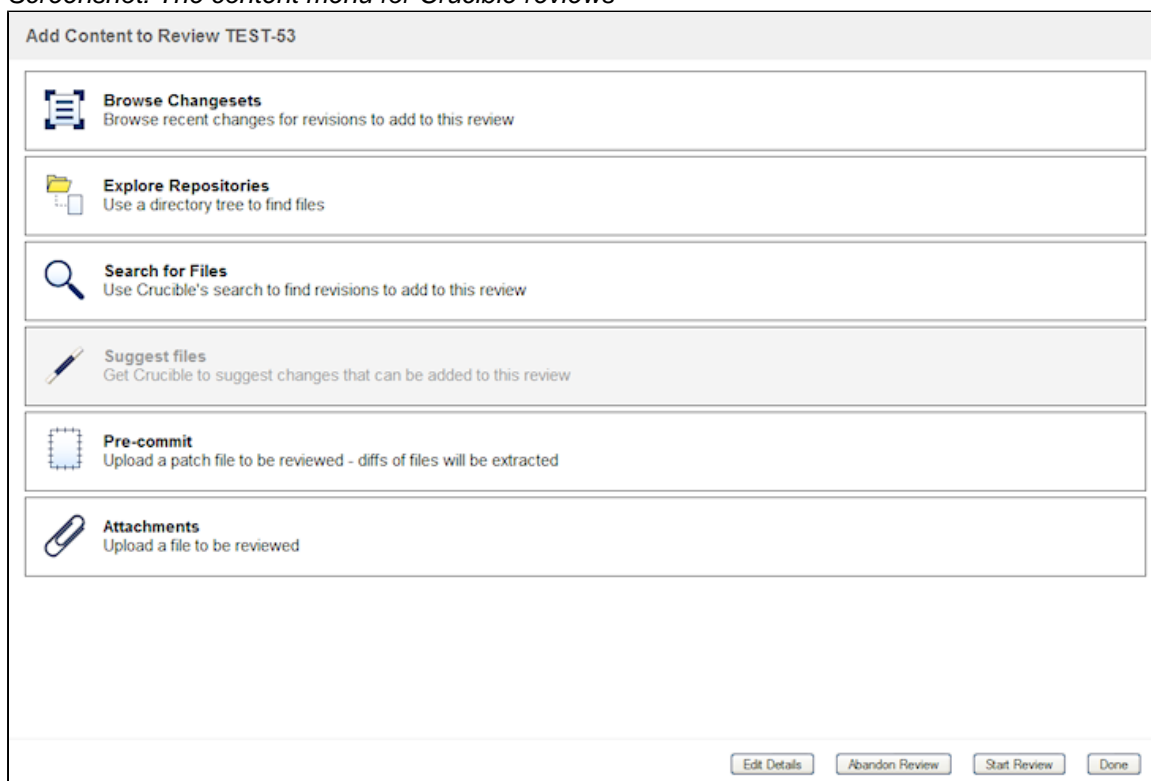*Screenshot: The create review dialog*

Select Project for New Review

Select Project

Testing Project

Create Review   Cancel

Choose where your review files will come from

The '**Add Content to Review**' screen appears, where you will now be prompted to choose where your review files will come from. Choose one of the options by clicking. See Selecting the Files for the Review for details.

Screenshot: The content menu for Crucible reviews

Add Content to Review TEST-53

Browse Changesets
Browse recent changes for revisions to add to this 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

Suggest files
Get Crucible to suggest changes that can be added 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

Edit Details   Abandon Review   Start Review   Done

Add content to the review

Once you select where your review files are coming from, you are prompted to select the files to be reviewed. Check the boxes next to any files you want to add.

Screenshot: Adding content to your review

Edit the review details

Once you have selected the files, click **Done** to go to the **Edit Review Details** screen, as shown below.

Screenshot: Editing review details

On the **Edit Review Details** screen, you can choose a title, reviewers, objectives, due date, linked reviews and issues. Once you're finished, click '**Done**'.

Screenshot: Editing review details

The review will open in a preview form. Here, you can check all the details and click to edit any that aren't correct. Once you click **Start Review**, the review is live.

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", or similar logic.

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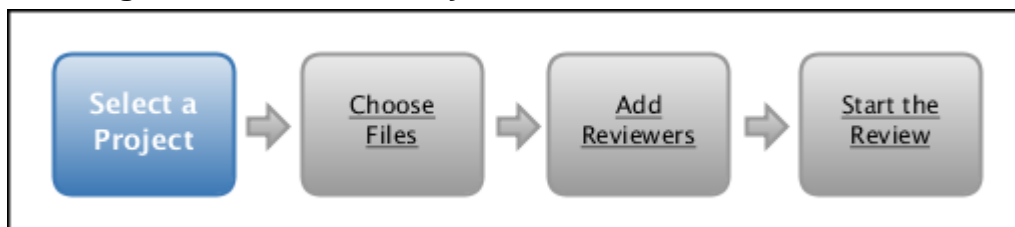
This page explains how to create a Crucible review from FishEye.

> **On this page:**
>
> 1. Open the FishEye Source view
> 2. Start the review
> 3. Choose a project

1. Open the FishEye Source view

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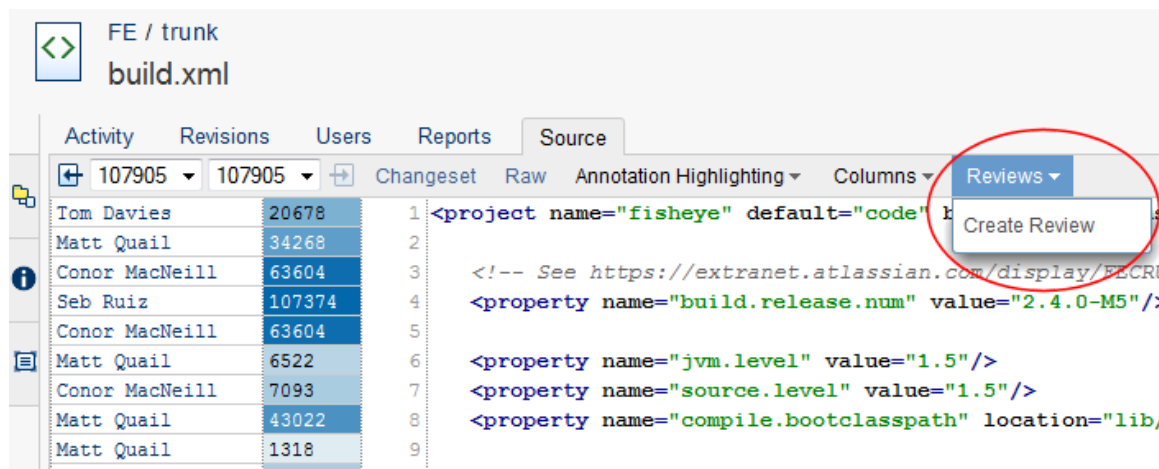
To navigate to the Source view for a file:

1. On the **Source** tab, click the name for a repository.
2. Click the name of the file to be reviewed.
3. Click the **Source** sub-tab.
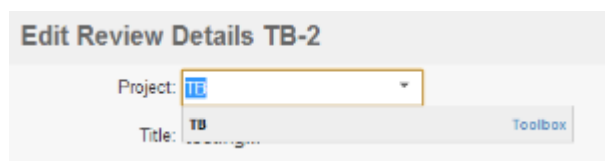
2. Start the review

To start the review:

- From the FishEye Source view, choose **Reviews** > **Create Review**.

Screenshot: Opening a review from the FishEye Source view

3. Choose a project

In the Edit Review Details dialog, choose the project for this review, and add reviewers. Now click **Start Review**.

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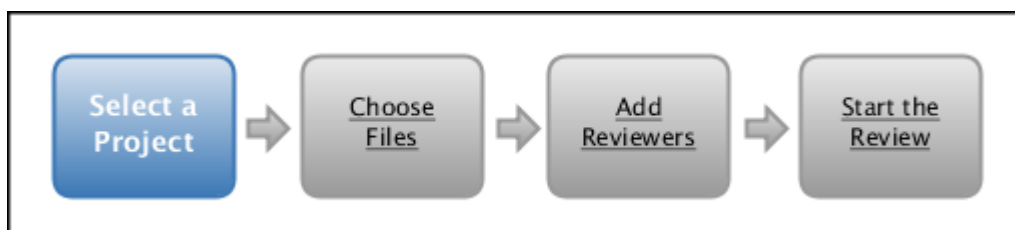
This page describes how to create a Crucible review directly from an issue in JIRA, the Atlassian issue-tracking application.

JIRA must be integrated with Crucible before you can do this. See JIRA integration in Crucible for information on how to set this up.

To create a review from within JIRA:

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 the '+' icon to the right of a changeset to create a new review for that changeset.
   b. Click **Create crucible review** to create a review for all the changesets in the list.
4. If a similar review already exists, you can add the changesets to that; otherwise click **Create New Review**
   .
5. In edit mode for a review:

- 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** when you are ready.

The next step is to add reviewers.

Screenshot: Adding a review from within JIRA
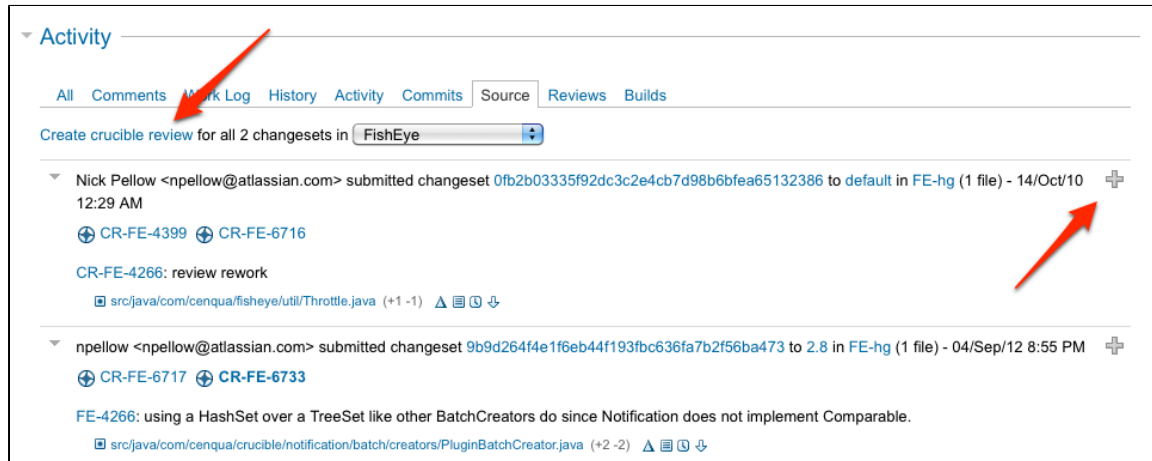
Screenshot: Review edit mode in Crucible
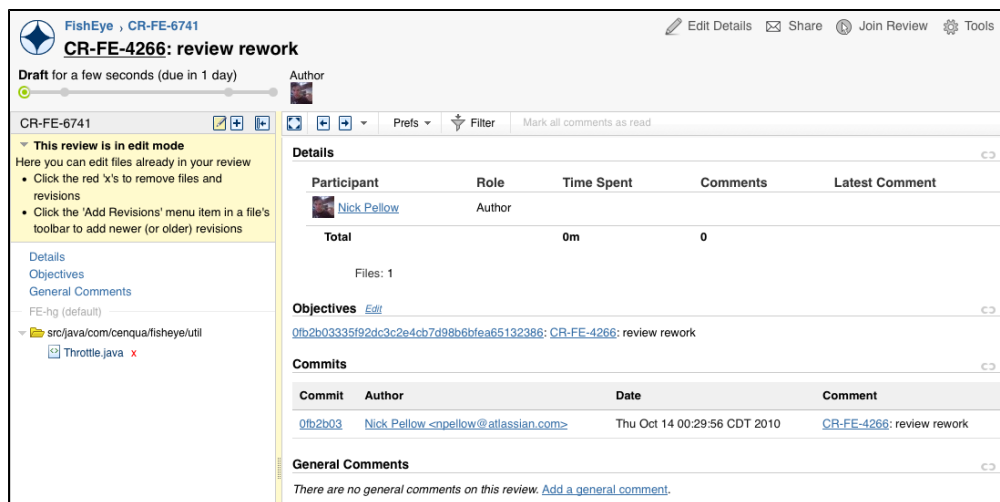
Creating a Review from a URL
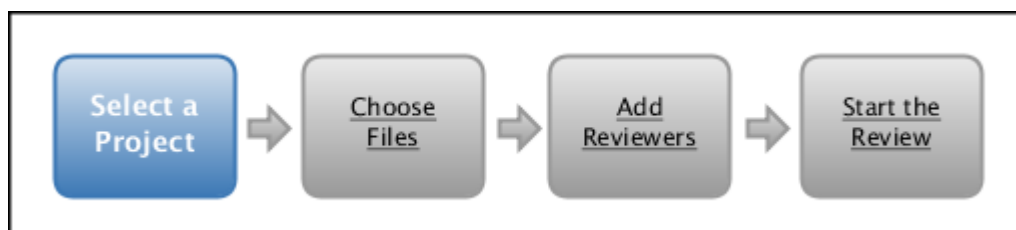
You can set up a URL which you can then click to create a Crucible review.

The format of your URL is as follows:

```
http://localhost:8060/cru/create?csid=%2F%2F&repo=a%2F1234&csid=%2F%2Frepob%2F758
3
```

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** 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 fro 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.

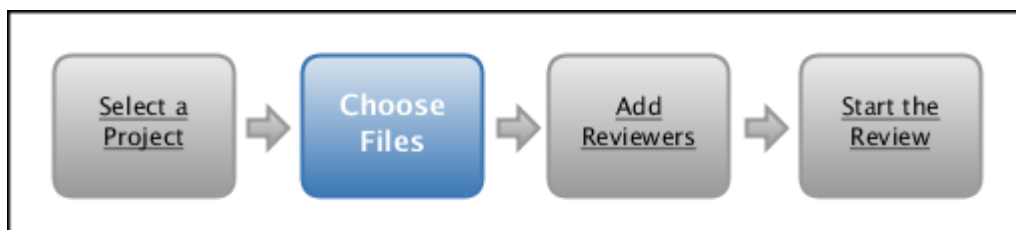*Screenshots: Creating a Snippet Review (click to view larger images)*
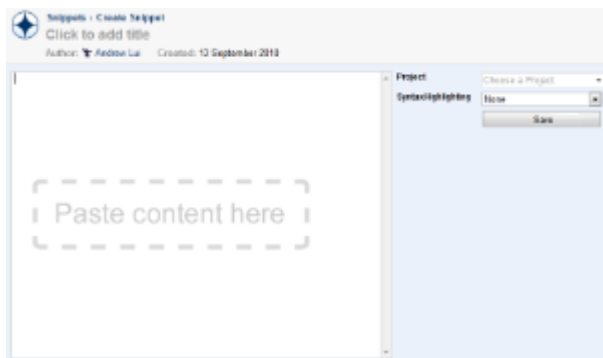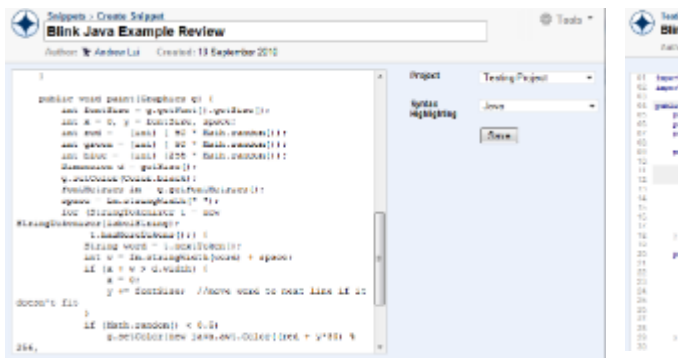
Step 1	Step 2

Selecting the Files for the Review

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

On this page:

- Selecting Changesets for Review
- Selecting Files for Review by Exploring Repositories
- Selecting Files for Review using the Crucible Search
- Using the Suggestions Feature When Adding Files to a Review
- Adding Pre-commit Patch Files to a Review
- Adding Attachments to a Review

To add content to a review:

1. Log in to FishEye/Crucible and either;
    - Create a review, as described on Creating a Review, or
    - Open an existing review, which you are the creator or moderator of, and click the Add Content (➕) button.
2. The 'Add Content to Review' dialogue will be displayed (see screenshot below). Select the method you want to use, to find the content for your review:
    - '**Browse Changesets**' — Allows you to choose changesets from a Source Code Management (SCM) repository to add to a review. See Selecting Changesets for Review.
    - '**Explore Repositories**' — Allows you to browse for files from a Source Code Management (SCM) repository to add to a review. This option only appears when FishEye is installed. See Selecting Files for Review using by Exploring Repositories.
    - '**Search for Files**' — Allows you to search a Source Code Management (SCM) repository for files or changesets to add to a review. This option only appears when FishEye is installed. See Selectin

g Files for Review using the Crucible Search.

- '**Suggest Files**' — Analyses the list of files in the current review and makes suggestions based on certain logic (for example, suggesting a newer version of a file if one exists). This option only appears when FishEye is installed. See Using the Suggestions Feature When Adding Files to a Review.
- '**Pre-commit**' — Allows you to upload patch files to a review. See Adding Pre-commit Patch Files to a Review.
- '**Attachments**' — Allows you to upload any file to a review, including binary files and files outside of a Source Code Management (SCM) repository. See Adding Attachments to a Review.

3. Follow the appropriate instructions in the sections below to add content to your review.
4. Click the '**Done**' button to finish adding content to the review.

Screenshot: The Add Content menu for Crucible reviews
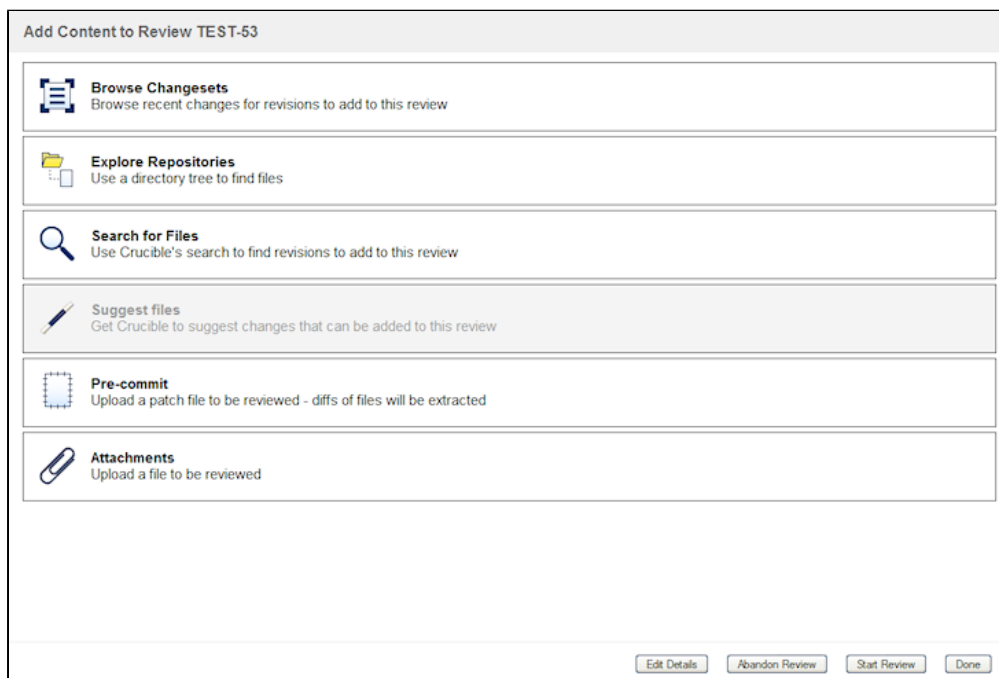
Selecting Changesets for Review

Click the '**Browse Changesets**' option on the '**Add Content**' dialogue to add changesets to your review.

Screenshot: The Browse Changesets View in the Add Content dialogue

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

Click the checkbox next to a changeset ID to add the entire changeset. Note,

- 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 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?.

Options for adding changesets:

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

Choosing the way files are added to the review:

When adding files to a review, you can set the form of review taking place in the **'Add to Review as'** drop-down menu:

- '**Whole Files**' — Adds the entire file with all content, rather than just a diff with context.
- '**Diffs**' — This is the default behaviour. This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
- '**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.
- '**Diffs to Last Branch Point**' — This adds files with a diff to the revision each file was last branched.

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

Selecting Files for Review by Exploring Repositories

Click the '**Explore Repositories**' option on the '**Add Content**' dialogue to browse repositories for files to add to your review.

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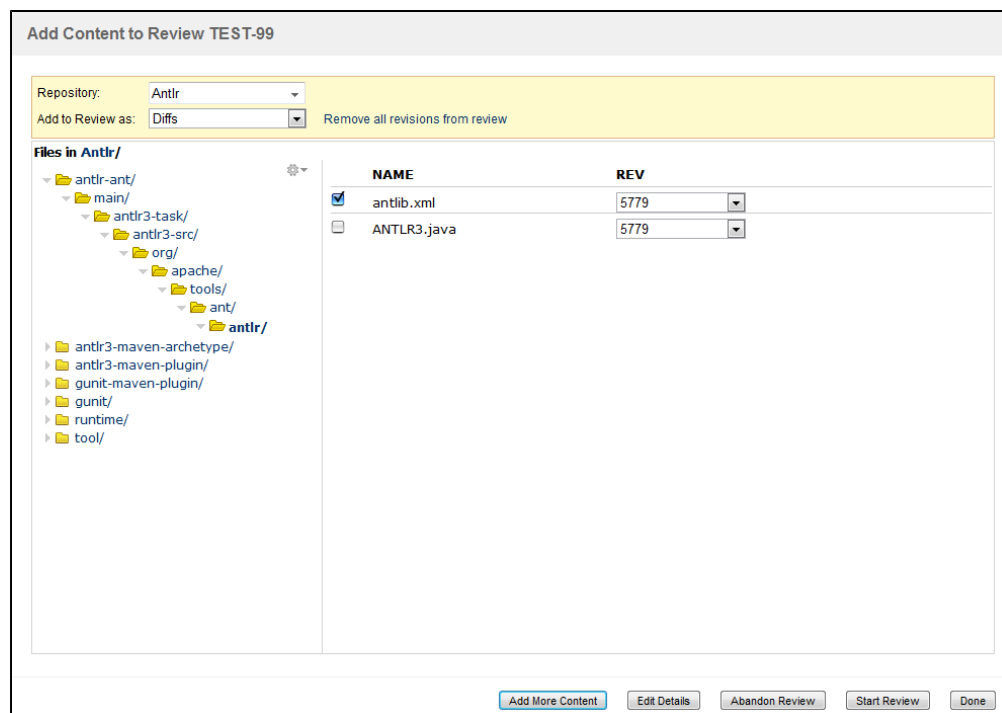
- To find a file, browse the folders by clicking the relevant folder. The folders by default are sorted by path name but can sorted by last-commit or first-commit.
- To choose a file for reviewing, click the checkbox to the left of the filename and if required the revision number.
- To select a particular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

> **ⓘ** Please note the following information when browsing files to add to a review:
>
> - Empty folders will be greyed out.
> - If the folders contain empty folders then a toggle option called '**Hide Empty**' will appear under the 'Sort' options.
> - To see or ignore **deleted** files, you can click the '**Hide**' and '**Show**' options located above the file names on the left.

Choosing the way files are added to the review:

When adding files to a review, you can set the form of review taking place in the **'Add to Review as'** drop-down menu:
- '**Whole Files**' — Adds the entire file with all content, rather than just a diff with context.
- '**Diffs**' — This is the default behaviour. This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
- '**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.

- '**Diffs to Last Branch Point**' — This adds files with a diff to the revision each file was last branched.

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

Selecting Files for Review using the Crucible Search

Click the '**Search for Files**' option on the '**Add Content**' dialogue to use the Crucible search to find files to add to your review.
⚠ The 'Search' view is only available when using FishEye with Crucible.

Screenshot: Searching for files to add to a review

If you are not certain about which changesets/revisions/files to include in a review, use the Search view to find them. Adjust the search filters to find the files you need. If the simple filters are not enough, read about EyeQL queries in the FishEye documentation.

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

Choosing the way files are added to the review:

When adding files to a review, you can set the form of review taking place in the **'Add to Review as'** drop-down menu:
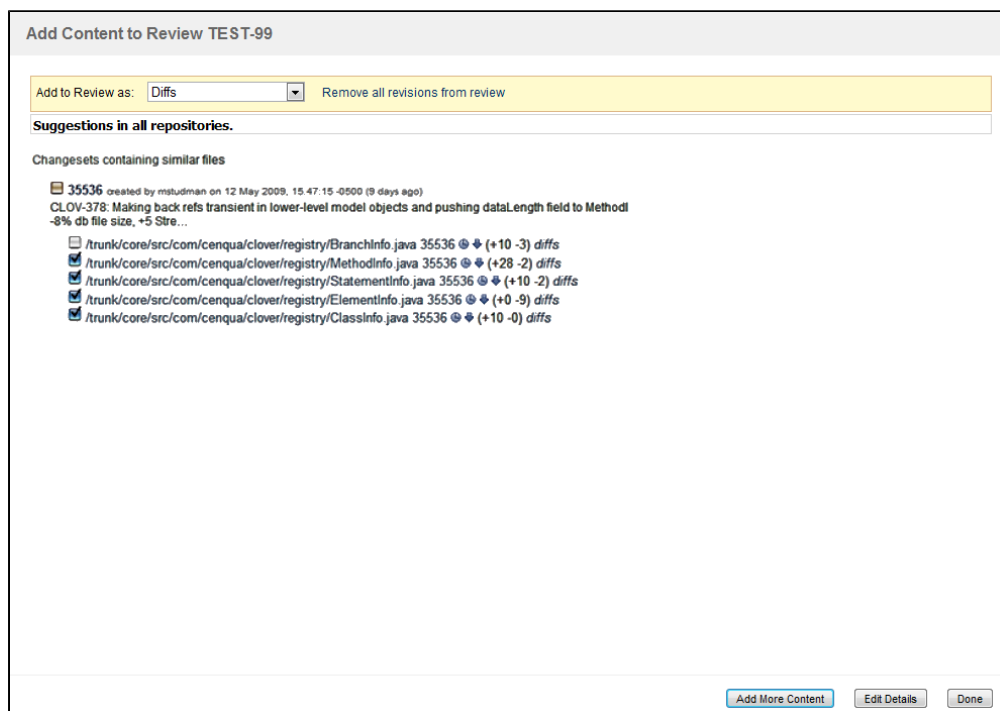- '**Whole Files**' — Adds the entire file with all content, rather than just a diff with context.
- '**Diffs**' — This is the default behaviour. This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
- '**Diffs to Last Reviewed 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.
- '**Diffs to Last Branch Point**' — This adds files with a diff to the revision each file was last branched.

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

Using the Suggestions Feature When Adding Files to a Review

Click the '**Suggest Files**' option on the '**Add Content**' dialogue to view and add files suggested by Crucible to your review. You need to have already added some file(s) to review for Crucible to suggest additional files.

Images: Viewing and adding file suggestions to a review

Crucible can make intelligent suggestions when you are creating a review. The Suggestions feature logic is based around the following:

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

Choosing the way files are added to the review:

When adding files to a review, you can set the form of review taking place in the **'Add to Review as'** drop-down menu:
- '**Whole Files**' — Adds the entire file with all content, rather than just a diff with context.
- '**Diffs**' — This is the default behaviour. This allows you to add multiple revisions of a file to one review and compare them in-review, in context with the change history.
- '**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.
- '**Diffs to Last Branch Point**' — This adds files with a diff to the revision each file was last branched.

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

Adding Pre-commit Patch Files to a Review

Click the '**Pre-commit**' option on the '**Add Content**' dialogue to add pre-commit patch files to your review.

Screenshot: Adding pre-commit patch files to a review

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

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

Adding Attachments to a Review

Click the '**Attachments**' option on the '**Add Content**' dialogue to add attachments to your review.

Screenshot: Adding attachments to a review

You can upload additional files to be used in the review, including binary files, images or code files that are not stored in a version control repository. The '**Upload**' view contains various controls to help you do this. These are listed below.

Choose the '**Upload Method**' as either '**Select file from the file system**' or '**Paste text from clipboard**':

- Displays if '**Upload method**' is 'Select file from the file system':
  - '**Character Set**' (if any) — Click the edit icon (✎) to choose the character set being used. ' **US-ASCII** ' is the default.
  - '**File**' — Click '**Browse**' to find the file that you want to add to the review.
- Displays if '**Upload method**' is 'Paste text from clipboard':
  - '**Patch text**' — Paste your copied text in this text area.

Click the '**Upload**' button, when you have made your selection. Once complete, a list of uploaded files is

displayed at the bottom of the screen. To add another iteration of a file, make changes to the file and upload it again with the same filename. It will be added as a new version.

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

Iterative Reviews

Crucible allows you to review several revisions of a file within one review, seamlessly switching between them. Comments are linked and relative to a specific revision. This allows you to review every change that has occurred on a code file within a given period of time. This lets you see the evolution of the file through various revisions (within one Crucible review).

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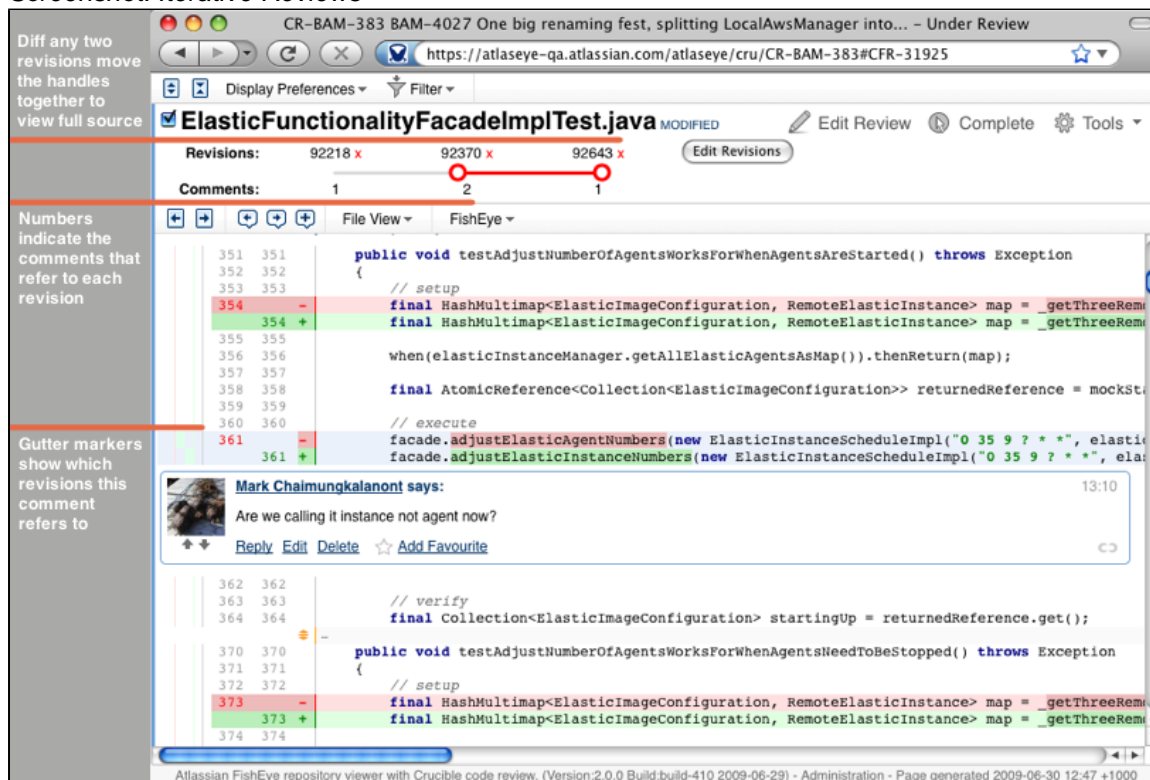
Adding Reviewers
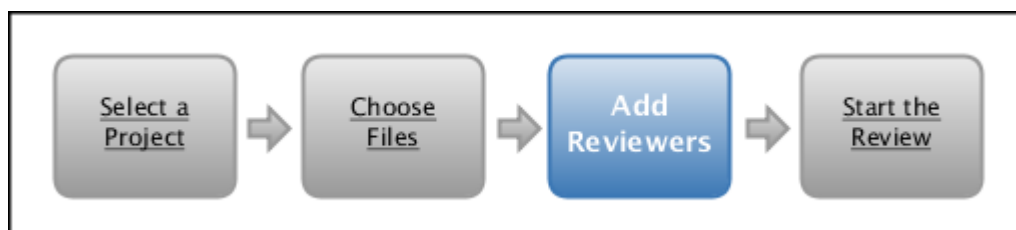
This page explains how to add reviewers to a new review, after it has been created. See Creating a Review for information about creating reviews.

On this page:

- Entering Basic Information
- Adding Reviewers
    - Adding Users to a Review
    - Crucible Suggests Reviewers
    - Inviting Non-Registered Users to the Review
    - Checking the Draft and Starting the Review

- [Next Steps](#)

Entering Basic Information

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

Screenshot: The Edit Review dialog

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

You must also select reviewers.

Adding Reviewers

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

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

Adding Users to a Review

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

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

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

Crucible Suggests Reviewers

Crucible will automatically suggest reviewers, by analysing 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.

Inviting Non-Registered Users to the Review

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

⚠️ There are two prerequisites:

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

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

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

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

Checking the Draft and Starting the Review

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

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

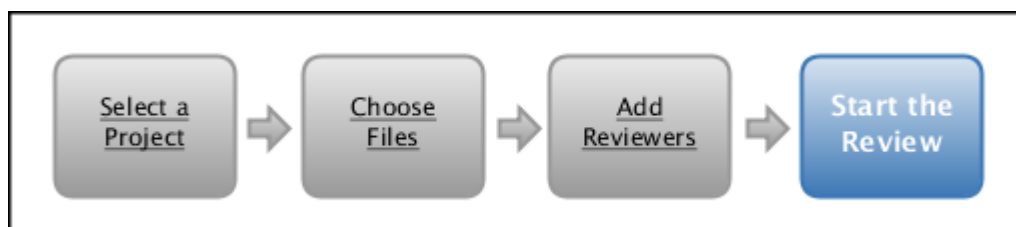*Screenshot: A newly created Crucible review*
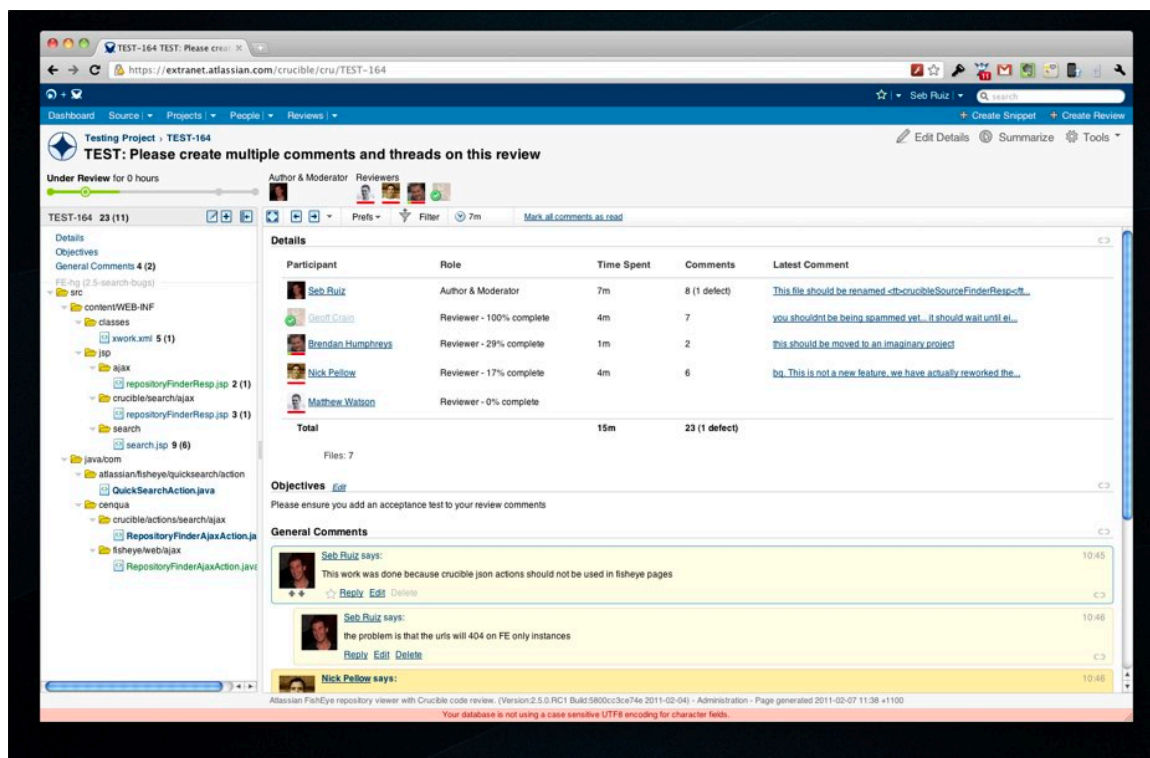
Next Steps

You can now begin [Performing the review](#).

If you have a moderator controlling your review process, you can move onto [Issuing a review](#).

Issuing a review

On this page:

- [Starting a review](#)
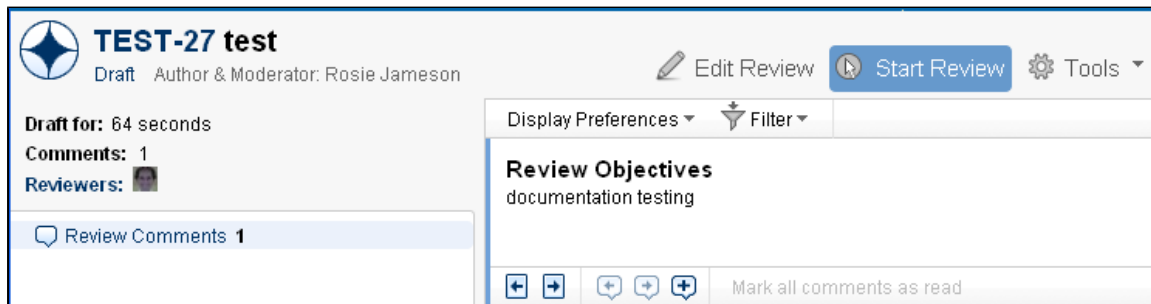- [Editing a review once started](#)

Starting a review

Issuing a review simply means formally starting it and inviting people to take part.

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

To start the review:

- If you are the moderator of the review, click **Start 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'.

Screenshot: Starting a Review

ⓘ Note that only people with the 'Approve' permission can start a review.

Editing a review once started

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

- Add files to a review by clicking the 'Add content' button.
- Remove files by clicking the 'Edit' button in the left navigation panel, then clicking the red cross icons beside files to be removed.

You can edit all the details of a review, including the content, by clicking **Edit Details** near the top right of the review.

Screenshot: Launching Edit Mode

Screenshot: Crucible Edit Mode for Review Content
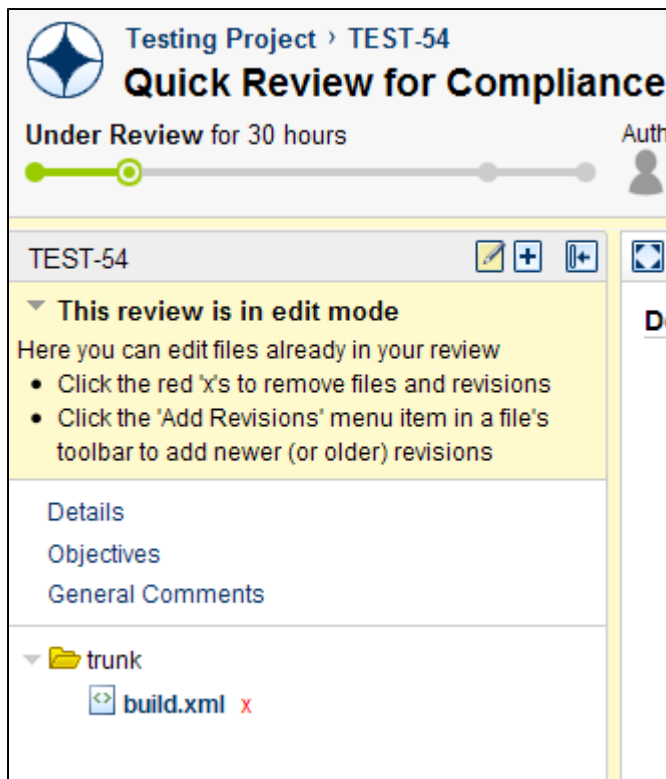
Performing the review

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

On this page:

- [Browse your reviews on the Dashboard](#)
- [Browse all reviews on the 'Reviews' tab](#)
- [When files change during a review](#)
- [Next steps](#)

> ℹ️ **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 is displayed, which shows your current reviews and other activity related to you.

Use the Dashboard to manage your reviews. Read the overview on [filtering your view](#).

Active reviews are listed on each [reviewer](#)'s dashboard under the default **To Review** filter. Reviews are listed under **Out for Review** until all reviewers indicate they are complete. The reviews then move to the **To Summarize** list.

Read more about using the [Dashboard](#).

Browse all reviews on the 'Reviews' tab

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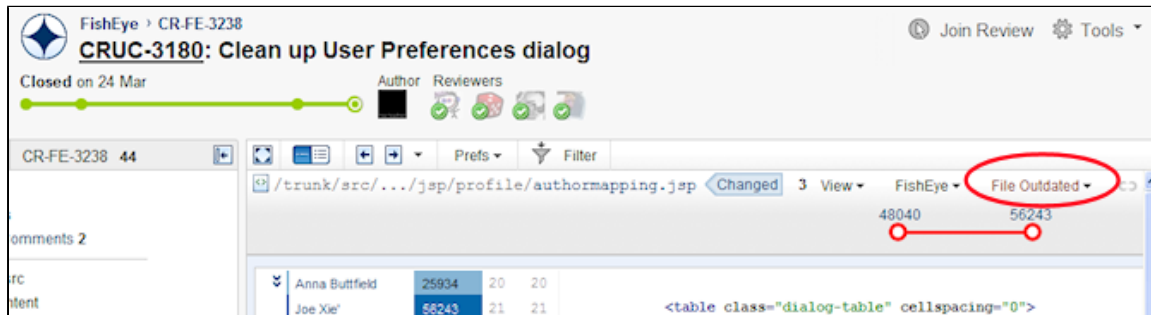
From the **File Outdated** menu, you can choose to view the latest revision of the updated file, or add the latest revision to the review:

Screenshot: 'File Outdated' menu
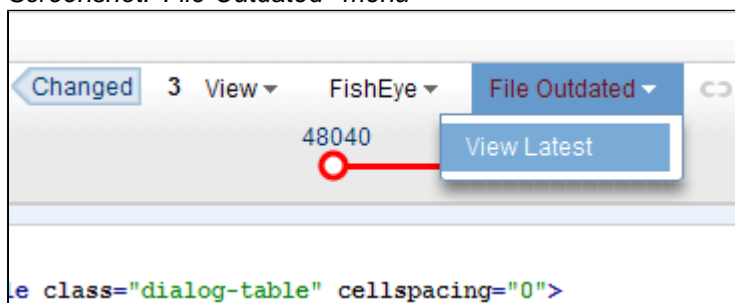
Next steps

- Adding comments
- Flagging Defects
- Creating JIRA issues from the review
- Completing your review
- Sending a review's comments via Email
- Using JIRA Integration in Crucible Reviews
- Using the Review History Dialog
- Tracking Crucible Review Metrics

Adding comments

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

On this page:

- Locating existing comments
- Adding comments
- Draft comments

Locating existing comments

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.)

Screenshot: Comments
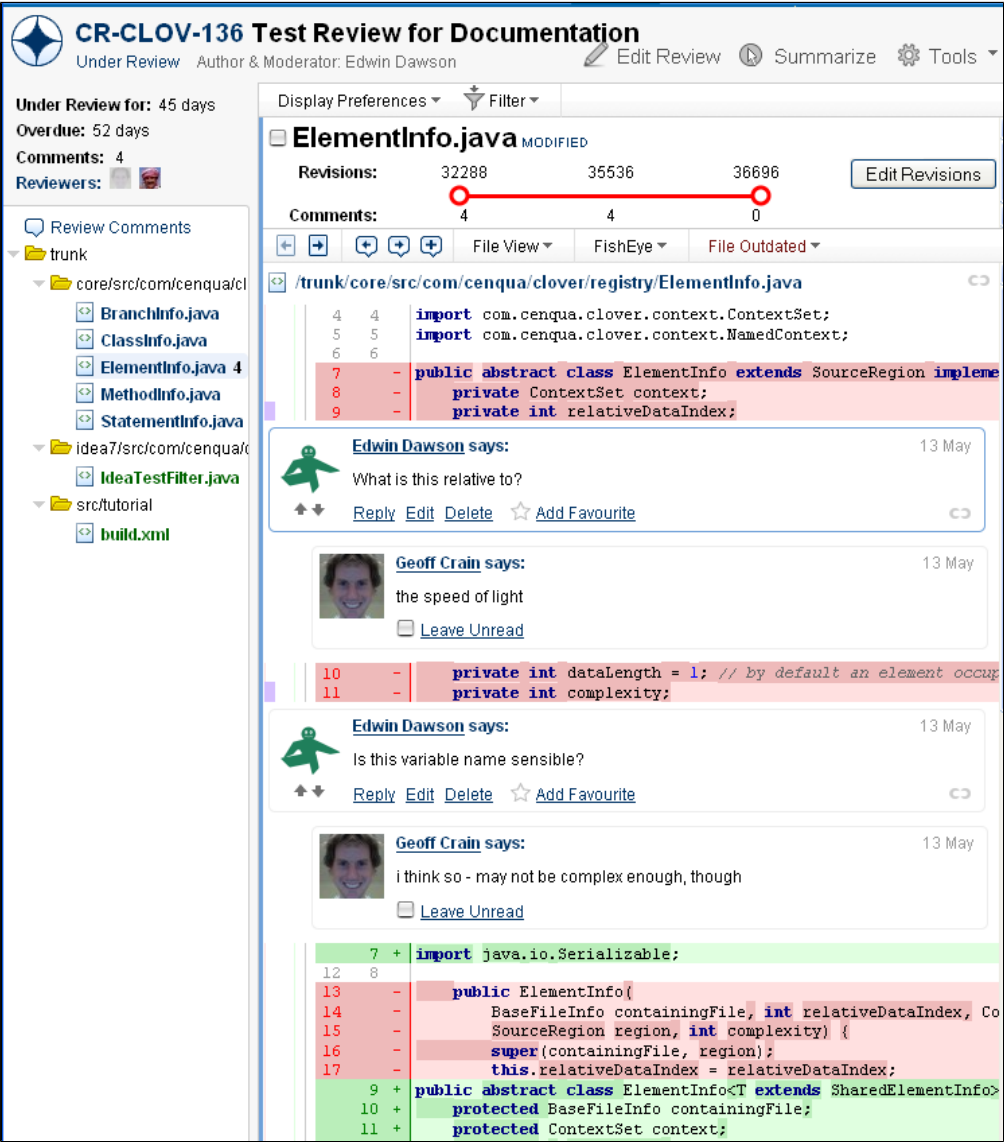
Adding comments

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

To comment on...	
The whole review	Click **General Comments** (in the left-hand panel), then **Add a general comment** (in the 'General Comments' section).
A source file under review	Click on the file in the left-hand panel, then click **Add a file comment** at the top of the file.

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.
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.

ℹ Read about flagging defects too.

Screenshot: Adding a Comment

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.

Screenshot: Draft 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.

Screenshot: Defects

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

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

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

Creating JIRA issues from the review

From any review comment (general, file, inline) in Crucible, you can create a JIRA 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.

Inline issue creation allows:

- Tracking of the status of the comment
- A faster way to pull out incidental suggestions raised in reviews as JIRA issues
- A quick link back to the comment from the JIRA 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 issues, unrelated to the current review, to track matters to be dealt with later.

Creating a JIRA issue

To create a JIRA 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.

Crucible suggests a JIRA 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 by your administrator.

Once the issue is created. the comment displays a link to the issue in JIRA, and in JIRA, 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 issues.

See Using JIRA Integration in Crucible Reviews for more information.

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 **Tools** menu:

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

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

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

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

Screenshot: Draft comments
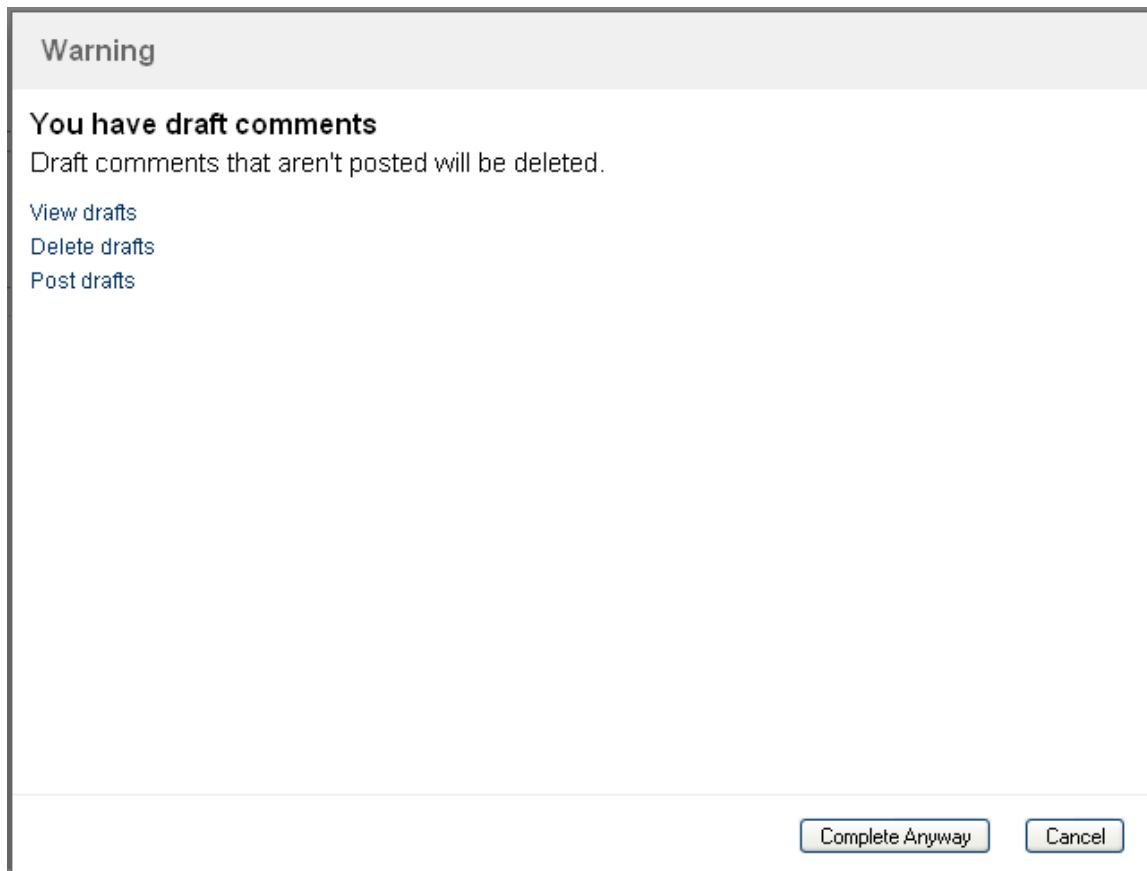
Screenshot: Review complete

Sending a review's comments via Email

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

Before you begin:

- Users who are not logged in cannot send email, but they can view the text content of the review's comments by clicking **View Text** that replaces **Email Review** in the **Tools** menu.
- Emails of reviews are only sent in plain text, not HTML. HTML emails are only available via watches.

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

1. In Crucible, navigate to the review in question.
2. Choose **Tools** > **Email Review** (see *Screenshot 1* below).
3. Specify the email recipients:

To	You can enter multiple email addresses, separated by commas.
Recipients	You can type usernames from your Crucible instance, to add them to the distribution list.
Send to Review Participants	Send the email to all reviewers.

4. Click **Next**.
5. You can modify both the Subject and Message parts of the email if required.
6. Click **Send** when you are ready. The 'Status' page confirms that your email has been sent.

Screenshot 1: The Email Review option in Crucible
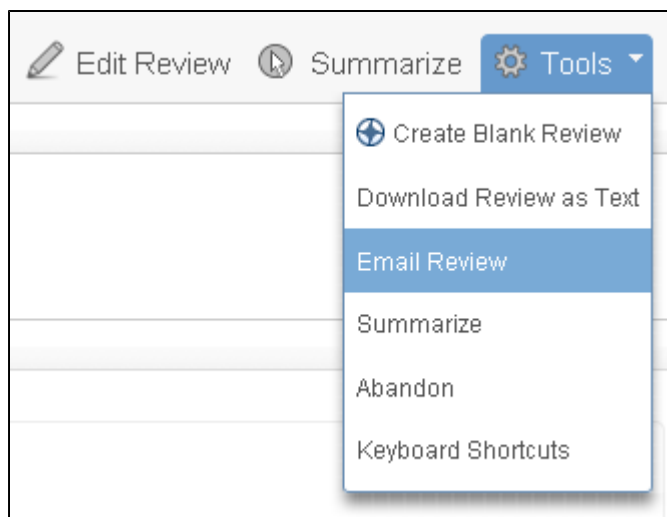
Screenshot 2: The 'Recipients' Screen in Crucible

Screenshot 3: The 'Message' Screen in Crucible

Screenshot 4: The Email Confirmation Screen in Crucible

Using JIRA Integration in Crucible Reviews

This page describes ways that you can use JIRA integration with Crucible reviews.

On this page:

1. Create a JIRA issue for your review
2. Create your review and link it to a JIRA issue
3. Comment, or flag a defect, on the review
4. Create a JIRA issue from a comment

ⓘ Before you begin, both your Crucible and JIRA instances must be configured to use make use of these JIRA integration features. The Crucible project requires a linked JIRA project before issues can be linked to reviews.

[Read more](#).

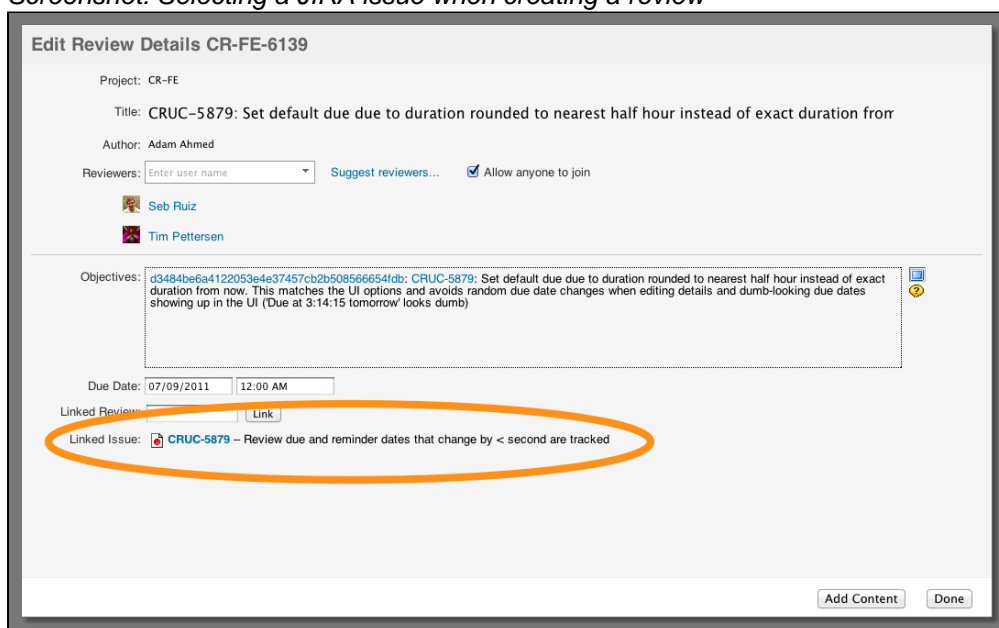### 1. Create a JIRA issue for your review

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

2. Create your review and link it to a JIRA issue

When creating, or editing, your review, Crucible will suggest a JIRA issue that can be linked to the review, if a JIRA issue key is found in the review title. You can:

- click the suggested JIRA issue key, to link it to the review
- delete the suggested JIRA issue, and then
- specify a different issue key and click **Link** to save it.

Screenshot: Selecting a JIRA issue when creating a review

3. Comment, or flag a defect, on the review

You can [add a comment](#) to the review. Check **Defect** when creating the comment to [flag this as a defect](#).

4. Create a JIRA issue from a comment

Click **Create Issue** in a comment to create a JIRA issue. Crucible suggests the JIRA 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.

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.
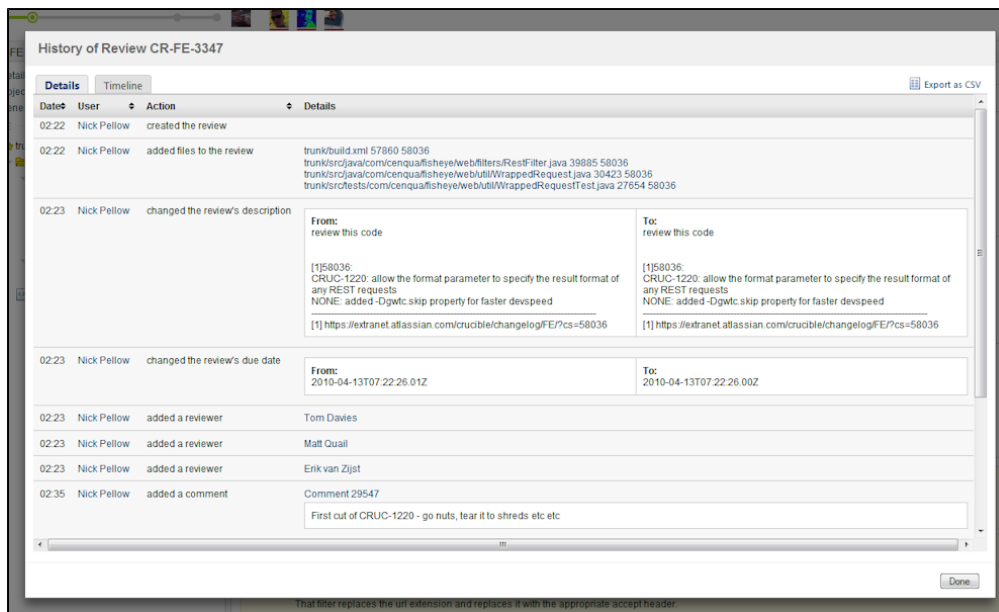
To open the Review History dialog:

1. Open a review in Crucible.
2. Click '**Tools**' > '**Review History**' at the toolbar at the top right corner of the screen.
3. The Review History dialog opens.

This information can also be displayed in the new timeline mode, a graphical visualisation that shows events on a horizontal graph over time (click the '**Timeline**' tab at the top of the dialog to switch from '**Details**' to timeline view). Click and drag inside the timeline view to scroll the graph left and right. You can also click on the section showing months to scroll over a greater time scale.

Additionally, you can get access to the entire review history through the '**CSV export**' link in the upper right hand corner, allowing for easy data import into a spreadsheet or other application.

Screenshot: The Crucible Review History Dialog

Screenshot: The Crucible Review History Dialog, Timeline View
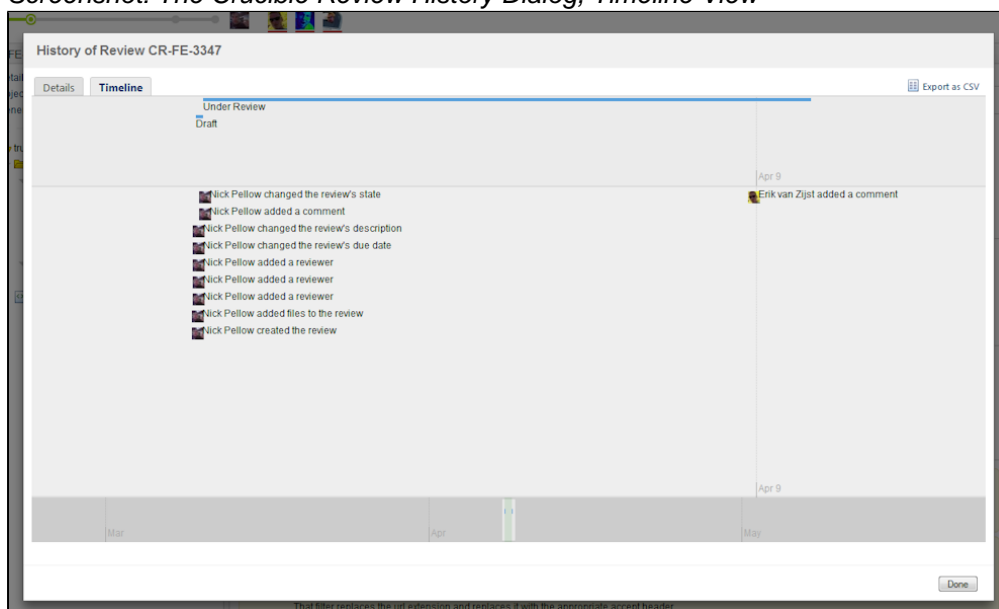
Tracking Crucible Review Metrics

Crucible tracks each participant's percentage completion through each review and the total time they have spent.

To learn about these features, see the following pages:

- Using Progress Tracking
- Using Time Tracking

Using Progress Tracking

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

On this page:

- How progress tracking works in Crucible
- Viewing the progress tracking totals
- How to adjust progress tracking on a review
- Adjusting settings for progress tracking
- Further reading

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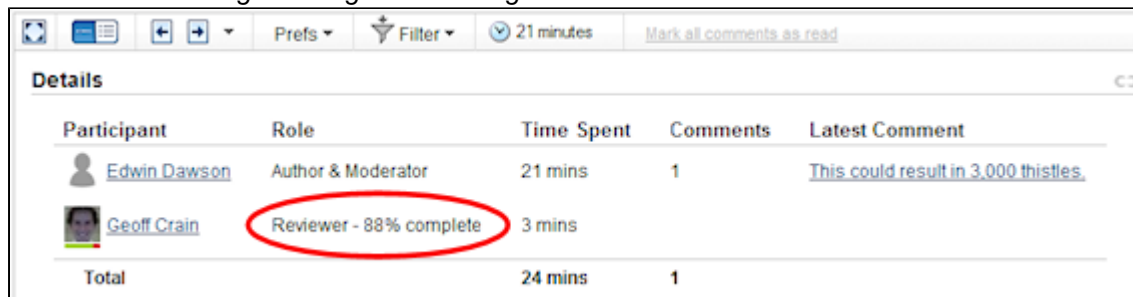
When participating in iterative reviews, progress tracking also takes lines of code and revisions into account.

Viewing the progress tracking totals

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

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

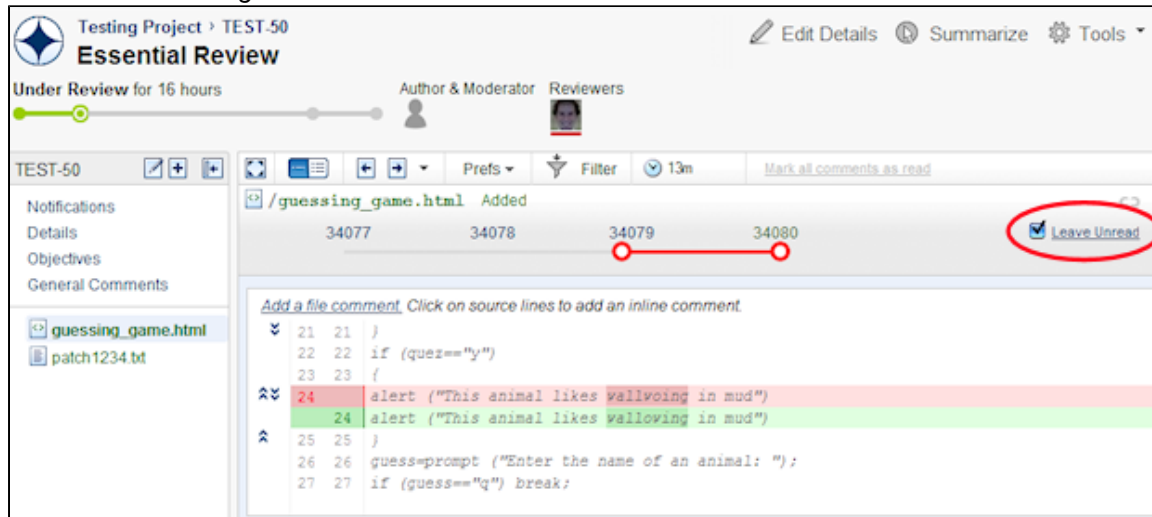*Screenshot: Viewing the Progress Tracking Totals*

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. In the source view, you have an option at the top left of the screen, '**Leave Unread**'. If you select that, then the file you are looking at will not be added to your progress percentage.
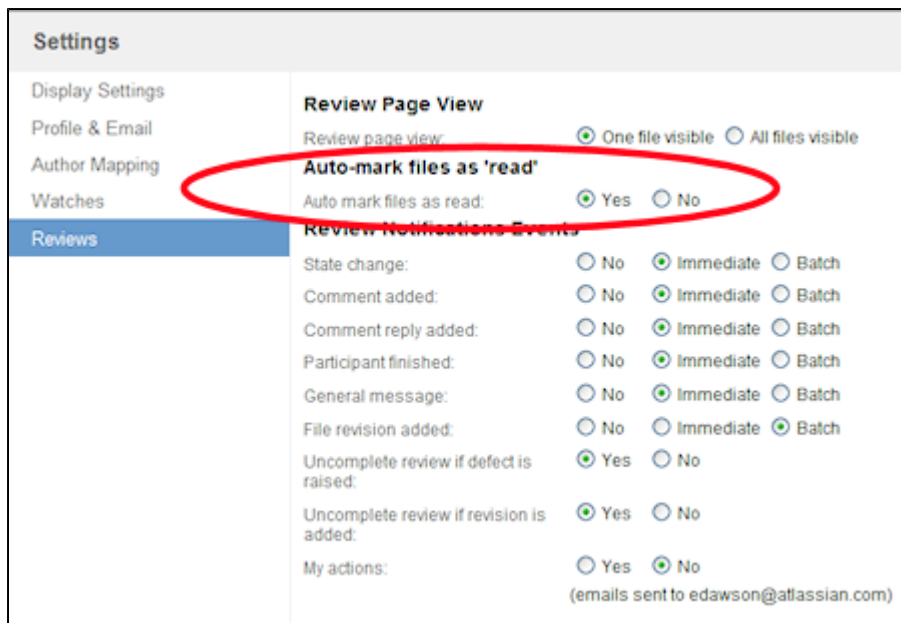
Screenshot: Marking a File as Unread

Adjusting settings for progress tracking

Progress tracking is a configurable user preference that can be changed in the '**User Settings**' menu, in the '**Re views**' sub-section. '**Auto-mark files as read**' is on by default. When '**Auto-mark files as read**' is set to '**No**', marking files as read or unread is left to the user to manually manage.

Screenshot: Adjusting the Progress Tracking User Settings

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.
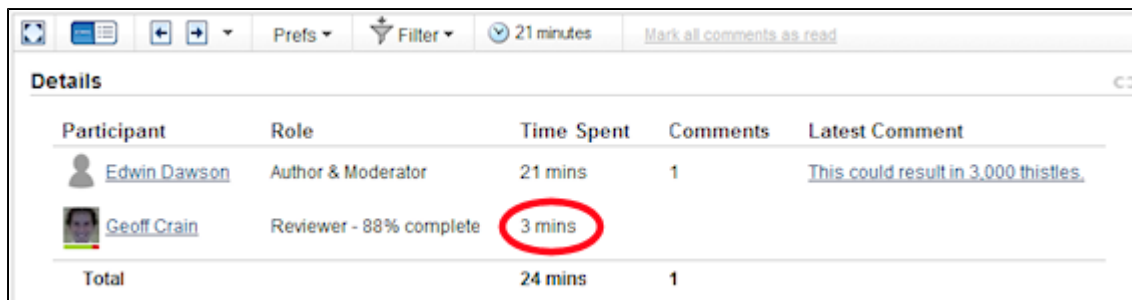
Screenshot: Crucible Time Tracking
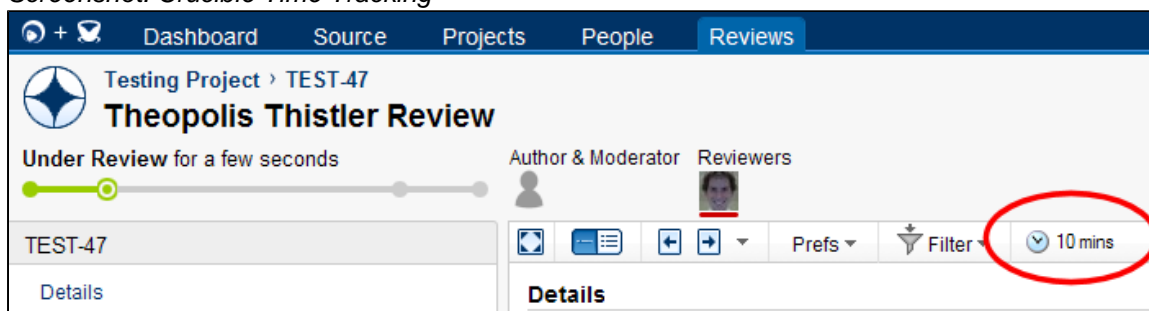
How to adjust the time tracked on a review

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**' view shows a summary of the progress and time tracked on each file.

Screenshot: Crucible Tracking Totals

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

Further reading

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

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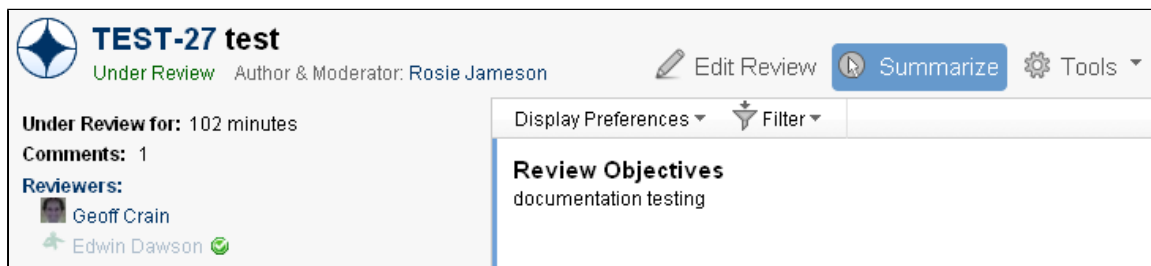
ℹ Normally, we recommend that you wait for all reviewers to complete their reviews.
The reviews that the reviewers have completed will be in your **Ready to Close** menu on the Dashboard.

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**.

Screenshot: 'Summarize' button

The above review is not yet complete

We can see that Geoff Crain has still not finished reviewing, because there is no green tick next to his name.
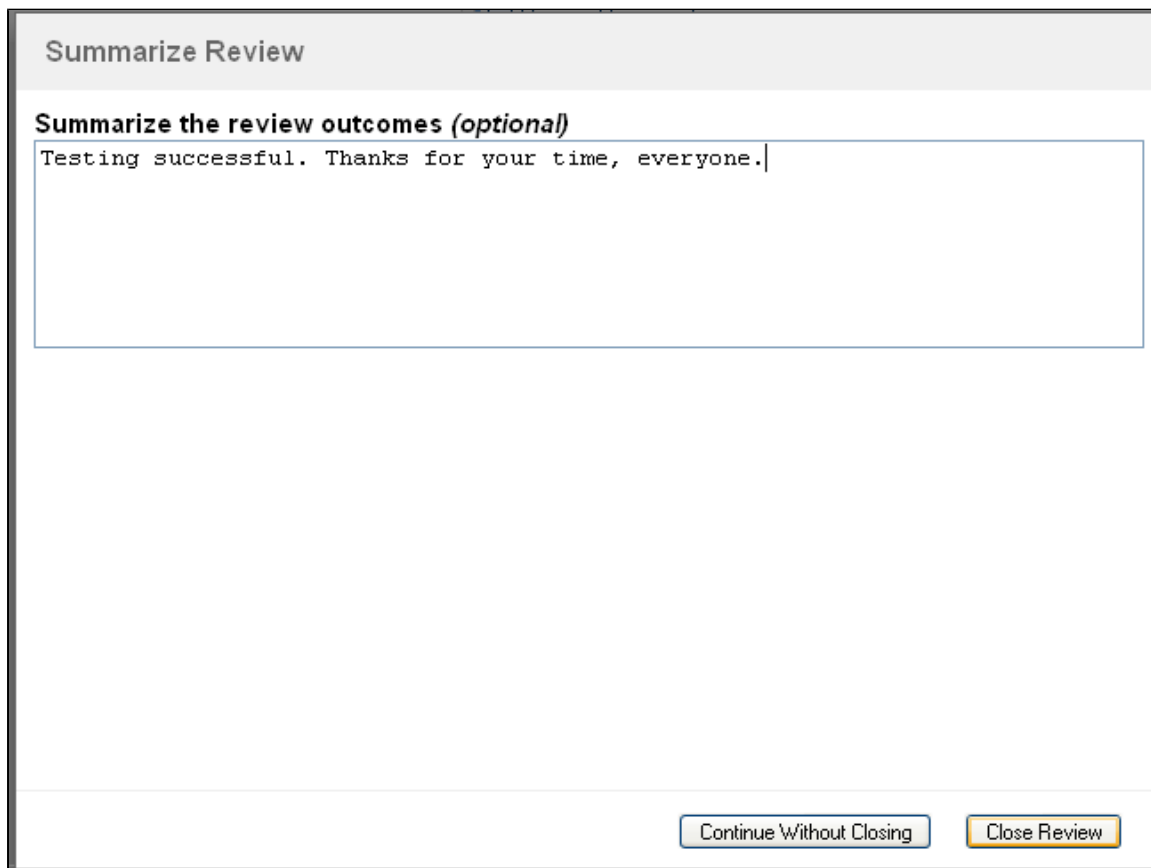
On clicking **Summarize**, the user may be prompted to confirm the action if there are incomplete reviews or draft comments in the review.

The requests for confirmation are warnings only

The review can still be summarized and closed.

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

Screenshot: Summarize Review

Screenshot: Review Closed

The summary is sent to all participants and 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.

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

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

Transitioning JIRA issues

For Crucible reviews that are linked to JIRA issues, you can now advance their JIRA workflow from within Crucible.

In the **Close Review** dialog, Crucible will now display any available workflow transitions for the linked issue.

Transitioning a linked issue

To transition a linked issue:

- In the Review Summary screen, choose **Access**: **Review** > (**Summarize** or **Close**) > **Close Review**. The possible transitions appear as buttons below the transition.
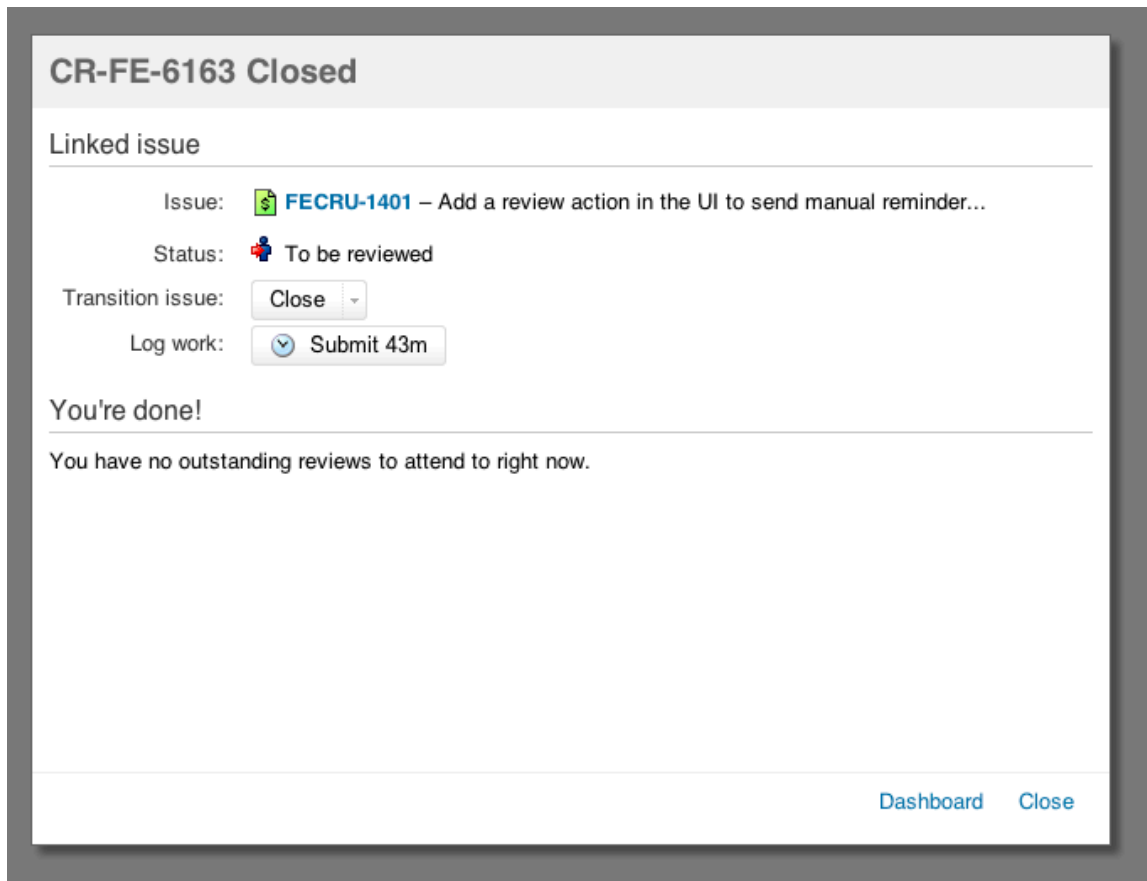
Screenshot above: Transitioning a JIRA issue in Crucible.

Notes
- JIRA administrators can [activate or deactivate JIRA issue transitions](#) (the workflow) from the JIRA administration interface.
- 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.

ℹ This feature does not support editing fields on the issue, only setting the *resolution* field if required by the transition.

Related Links
- [Using JIRA Integration in Crucible Reviews](#)
- [JIRA integration in Crucible](#)

Moving a Review to Another Project

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

To move a review between projects:

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

Screenshot: Changing a Review's Parent Project

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
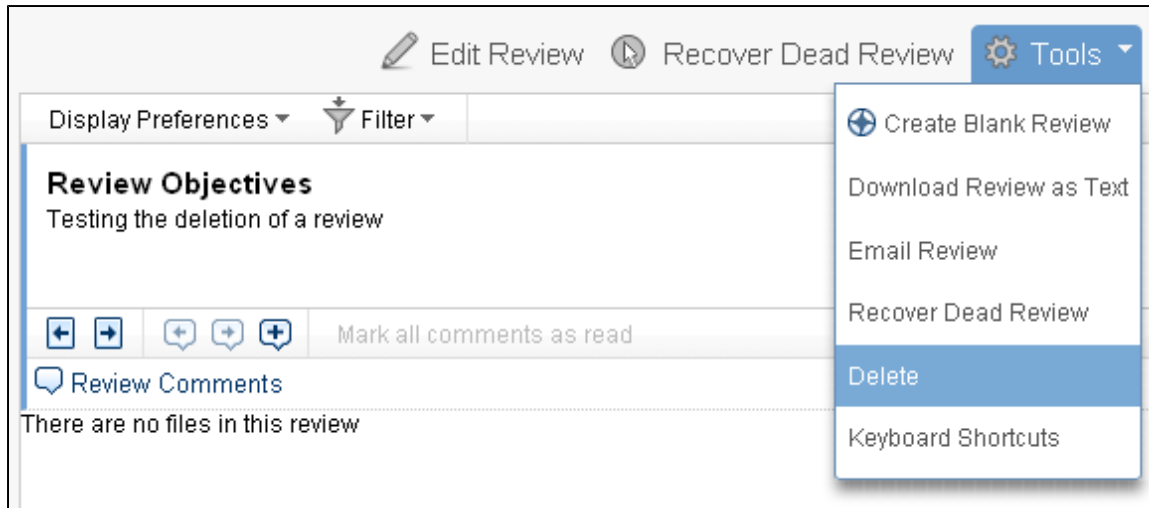
Defining your Workflow

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

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

Lightweight Code Commenting with Crucible (individual)

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

Diagram: Workflow for Lightweight Code Commenting
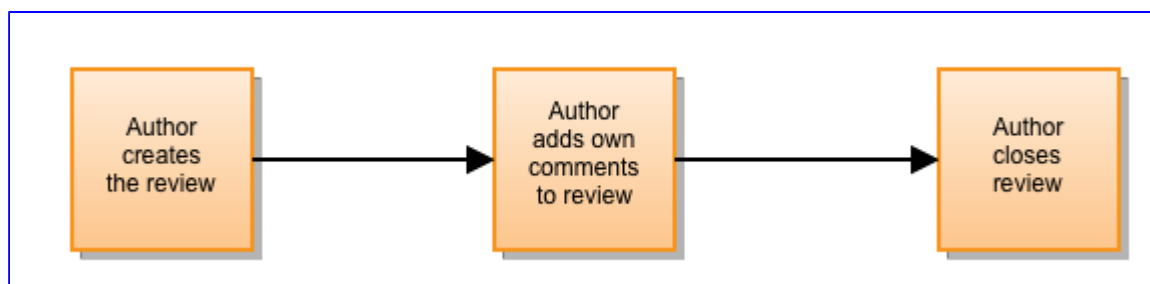
One-to-One Reviews (Agile Pair)

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

*Diagram: Workflow for One-to-One Reviews*
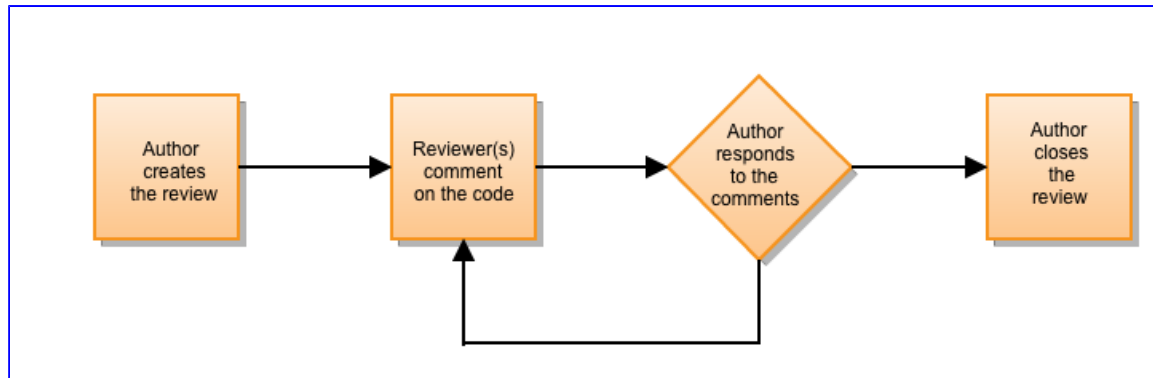
For more information on one-to-one reviews, see The Crucible workflow. The workflow process in Crucible is covered in detail within this document.

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

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

Diagram: Workflow for One-to-Many Reviews
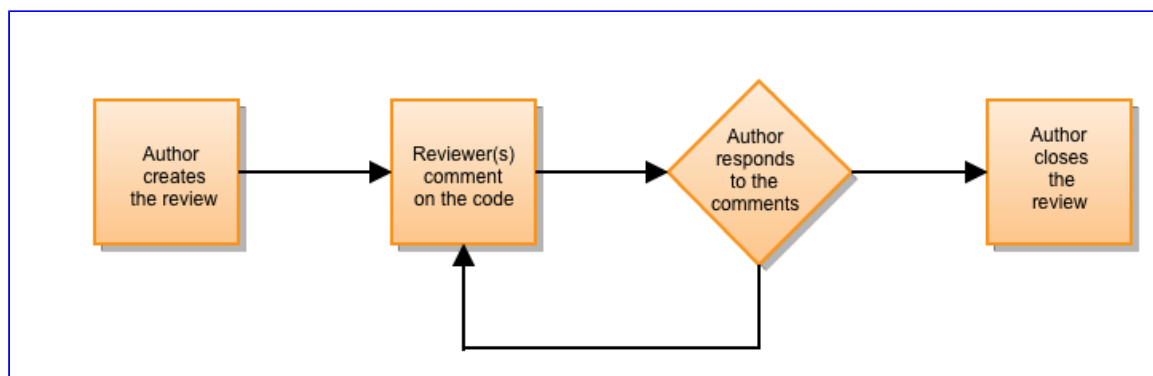
Formal Group Reviews (CMM Team)

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

Diagram: Workflow for Formal Group Reviews
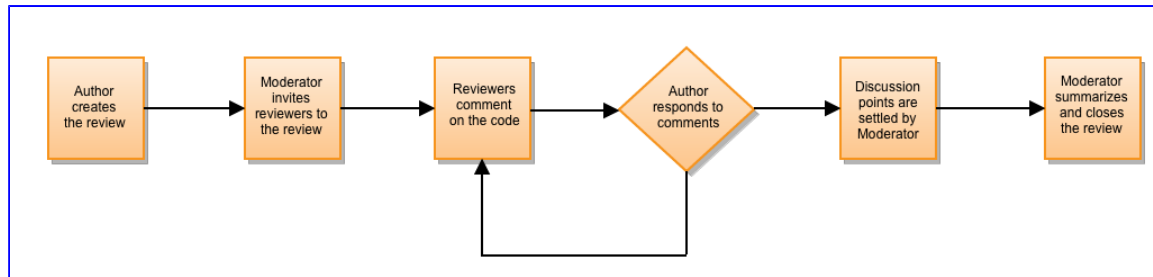
To see a simple example of how to use Crucible with two people, see The Crucible workflow.

Viewing Your Favourites

This page contains instructions on how to view, rename and remove your **existing favourites** in Crucible. You can select code reviews, changesets, files, people and repositories as favourites in Crucible. This allows you to personalise the information that you see in your activity stream.

See Using Favourites for instructions on how to add new favourites.

On this page:

- Viewing your Existing Favourites
- Renaming a Favourite
- Removing a Favourite

Viewing your Existing Favourites

To view your favourites,

1. Click the favourites icon (⭐) in the header.
   The code reviews, changesets, files, people and repositories that you have selected will be displayed in a list on the dashboard (see screenshot).
2. Click the favourites icon (⭐) next to a favourite to show the dialogue for renaming (i.e. changing the 'Label') the favourite or removing the favourite. See Using Favourites for more information.

☑ *Tip: You can add favourites wherever you see star icons next to code reviews, changesets, files, people and repositories, not just on this screen.*
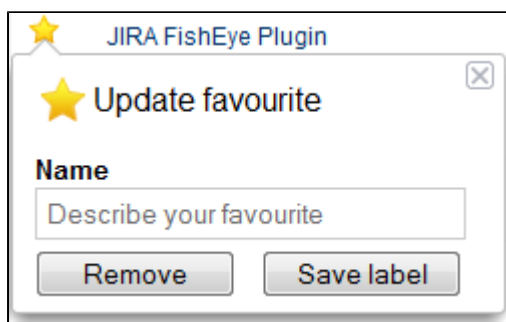
Screenshot: Viewing favourites

Renaming a Favourite

To rename a favourite,

1. Click the favourites icon (⭐) in the header.
   The code reviews, changesets, files, people and repositories that you have selected will be displayed in a list on the dashboard (see screenshot).
2. Click the favourites icon (⭐) next to a favourite to show the 'Update favourite' dialogue (see screenshot below).
3. Enter the new name for the favourite in the '**Name**' field and click the '**Save label**' button. The favourite will be renamed on the dashboard.

*Screenshot: The 'Update favourite' dialogue*

Removing a Favourite

To view your favourites,

1. Click the favourites icon (⭐) in the header.
   The code reviews, changesets, files, people and repositories that you have selected will be displayed in a list on the dashboard (see screenshot).
2. Click the favourites icon (⭐) next to a favourite to show the 'Update favourite' dialogue (see screenshot below).
3. Click the '**Remove**' button. The code review, changeset, file, person or repository will be removed as your favourite (the icon will be greyed out) and will not display the next time you view your list of favourites.

✅ *Tip: You can remove favourites wherever you see star icons next to code reviews, changesets, files, people and repositories, not just on this screen, not just on this screen.*

Screenshot: The 'Update favourite' dialogue
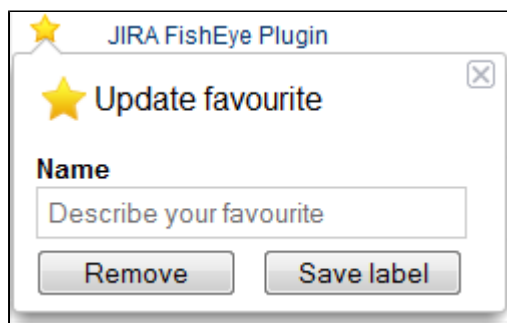
Using Favourites

This page contains instructions on **adding favourites** in Crucible. You can select code reviews, changesets, files, people and repositories to be added to your favourites. This allows you to personalise the information that you see in your [activity stream](#). We suggest you select items that you are currently working on as favourites, to create a more relevant personalised view.

See [Viewing Your Favourites](#) for instructions on how to view your existing list of favourites and how to rename and remove favourites.
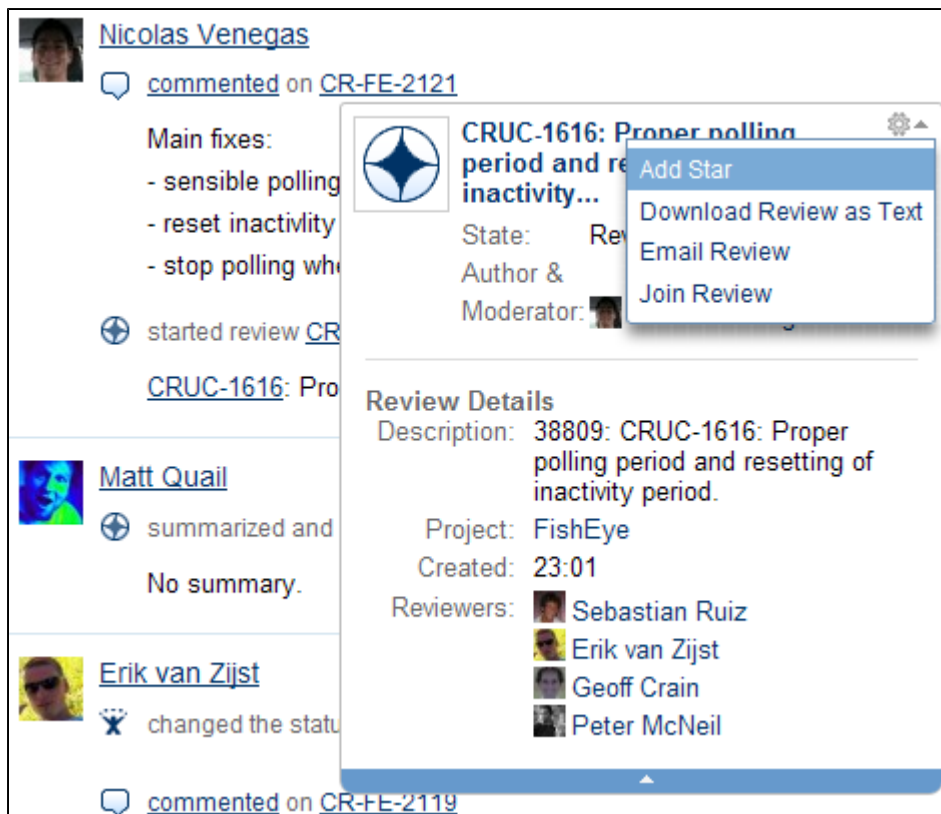
On this page:

- [Adding a Review to Your Favourites](#)
- [Adding a Review Comment Thread to Your Favourites](#)
- [Adding a Project to Your Favourites](#)
- [Adding a Person to Your Favourites](#)
- [Adding a Changeset to Your Favourites](#)
- [Adding a File or Folder to Your Favourites](#)
- [Adding a Repository to Your Favourites](#)

Adding a Review to Your Favourites

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

Screenshot: Adding a Review To Your Favourites

Adding a Review Comment Thread to Your Favourites

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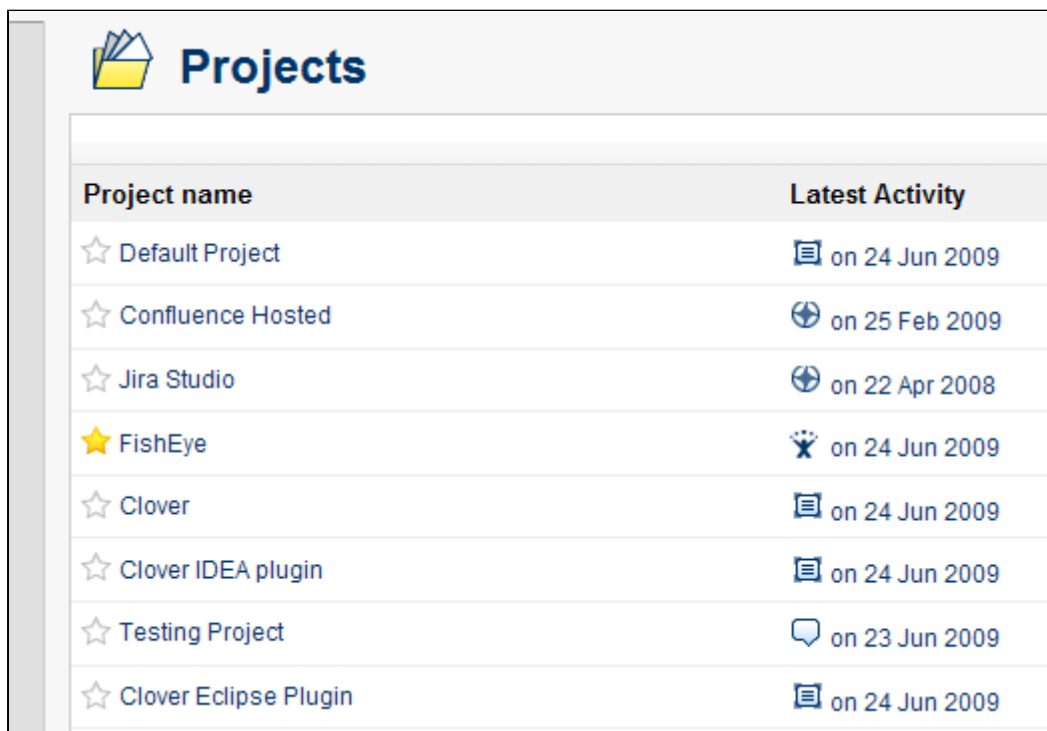
Screenshot: Adding a Review Comment Thread to Your Favourites
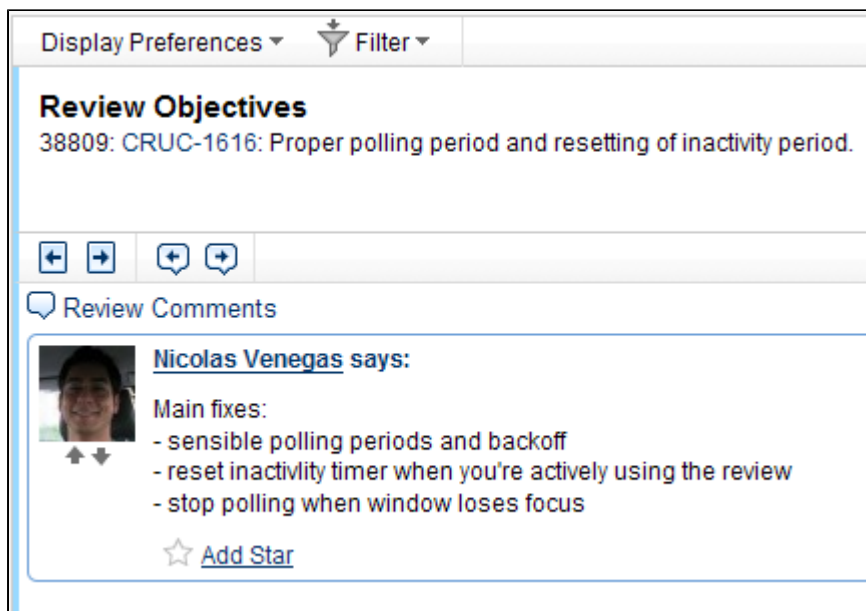
Adding a Project to Your Favourites

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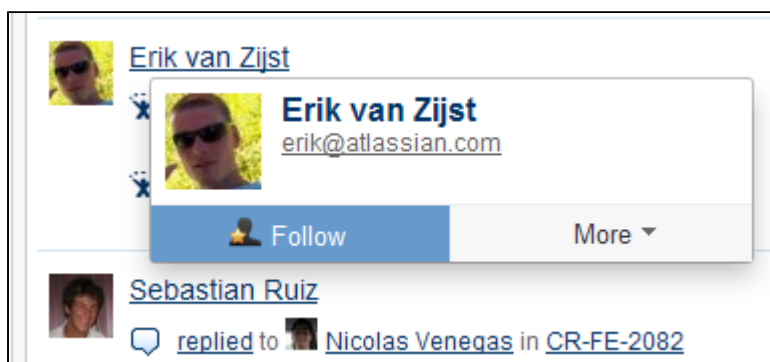
Screenshot: Adding a Project to your Favourites

Adding a Person to Your Favourites

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

Screenshot: Adding a Person to Your Favourites

Adding a Changeset to Your Favourites

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

Screenshot: Adding a Changeset to Your Favourites

Adding a File or Folder to Your Favourites

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

Screenshot: Adding a File or Folder to Your Favourites

Adding a Repository to Your Favourites

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

Screenshot: Adding a Repository to Your Favourites

Using Keyboard Shortcuts in Crucible

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

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

See the tables below for full details:

General Shortcuts

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

Custom Navigation

Key	Function
,	(Comma) Go to the previous element, which can be a file, comment or defect, depending on your current context.
.	(Period) Go to the next element, which can be a file, comment or defect, depending on your current context.

Comment Navigation Shortcuts

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

File Navigation Shortcuts

Key	Function
j	Go to next file.
k	Go to previous file.
shift + k	Go to first file.
shift + j	Go to last file.
u	Go to next unreviewed file.
i	Go to previous unreviewed file.
y	Set file reviewed and go to next unreviewed file.

shift + y	Toggle file reviewed/unreviewed status.
e	Expand current file.
c	Collapse current file.
shift + e	Expand all files.
shift + c	Collapse all files.

Using RSS Feeds in Crucible

Subscribing to an RSS Feed

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

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

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

Using Wiki Markup in Crucible

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

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

ℹ️ 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**

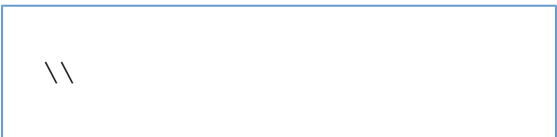| h5.Small heading	Turns text into a heading at size 5.
	Small heading
h6.Smallest heading	Turns text into a heading at size 6.
	Smallest heading
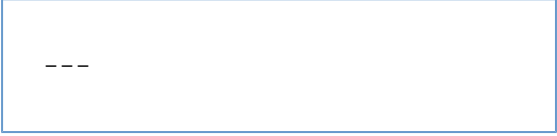
Text Effects

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 <u>underlined</u>.
??citation??	Makes text appear in *citation* form.
-strikethrough-	Makes text appear ~~struck through~~.
^superscript^	Makes text appear in ^{superscript}.
~subscript~	Makes text appear in $_{subscript}$.
`{{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 organise content in a rows and columns, with a header row if required.

Notation	Description

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

The code above produces a table that looks like this:

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

Advanced Formatting

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

Notation	Description
{noformat}	Makes a preformatted block of text with no syntax highlighting. All the optional parameters of the {noformat} macro are valid for the {panel} macro as well. Example: ``` This is a no-formatted piece of text, so *no* _formatting_ is done here. ```
{panel}	Embraces a block of text within a fully customizable panel. The optional parameters you can define are as follows. • title: Title of the panel • borderStyle: The style of the border this panel uses (solid, dashed and other valid CSS border styles) • borderColor: The color of the border this panel uses • borderWidth: The width of the border this panel uses • bgColor: The background color of this panel • titleBGColor: The background color of the title section of this panel Examples: Some text in a basic panel **My Title** Some text with a title

```
{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	🙂	🙁	😛	😄	😉	👍	👎	ℹ️	✅	❌	⚠️

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

Using gadgets in Crucible

This page explains how to add a Crucible gadget to a JIRA dashboard. The process is similar for adding a gadget to a Confluence page.

On this page:

Overview of Crucible gadgets

As of the release of Crucible 2.3, you can display Crucible data in other Atlassian applications, such as JIRA and Confluence, by using Crucible gadgets.

Crucible bundles the following gadgets by default:

Gadget Name	Description and Gadget URL	Screenshot
Hassle Review Blockers	This gadget shows you who you are still waiting on; in other words, which reviewers haven't completed your reviews. The URL for this gadget is: ```http://HOSTNAME:8060/rest/gadgets/1.0/g/com.atlassian.fecru.fecru-gadgets-plugin:overdueReviews/gadgets/hassle.xml``` Where `HOSTNAME:8060` is the hostname of your Crucible instance.	

Overdue Reviews	This gadget shows you reviews that are yet to be completed in the project, across all authors. This is useful for managers or team leads.	
	The URL for this gadget is:	

	`http://HOSTNAME:8`
	`060/rest/gadgets/`
	`1.0/g/com.atlassi`
	`an.fecru.fecru-ga`
	`dgets-plugin:over`
	`dueReviews/gadget`
	`s/overdueReviews.`
	`xml`	

	Where `HOSTNAME:8060` is the hostname of your Crucible instance.	
Review Inbox	This gadget is a list of Crucible to-do items including reviews to do, comments to read or reviews to summarise.	
	The URL for this gadget is:	

	`http://HOSTNAME:8`
	`060/rest/gadgets/`
	`1.0/g/com.atlassi`
	`an.fecru.fecru-ga`
	`dgets-plugin:over`
	`dueReviews/gadget`
	`s/todo.xml`	

	Where `HOSTNAME:8060` is the hostname of your Crucible instance.	

Crucible: Overdue reviews in All Pro

ID	Name
CR-3	FE-3456: Make stuff nicer
CR-4	FE-4567: Disentangle states

Crucible: Review Inbox

state	ID	Name
Summarize	CR-3	FE-3456:
Respond	CR-1	FE-1234:
Review	CR-2	FE-2345:

| Review Coverage | This gadget shows content from the innovative Review Coverage report, letting you investigate how much of your codebase has been under code review.

The URL for this gadget is as follows:

`http://HOSTNAME:8060/rest/gadgets/1.0/g/com.atlassian.fisheye.review-coverage-report/gadgets/recent-changesets.xml`

Where `HOSTNAME:8060` is the hostname of your Crucible instance. | |

1. Add JIRA to FishEye as an OAuth consumer

Firstly, you need to set up an application link from FishEye to JIRA:

1. In the FishEye admin area, click **Application Links** (under 'Global Settings') in the left navigation bar.
2. Click **Add Application Link** and complete the wizard. Leave the **Also create a link** checkbox selected. Choose the **These servers fully trust each other** option.

2. Add the Crucible gadget to JIRA's Gadget Directory

As a JIRA administrator you allow the use of these gadgets by adding them to the Gadget Directory. You'll need the URL listed in the table above for each gadget that you want to add.

See the JIRA documentation for details on this process.

3. Add the Crucible gadget to a JIRA dashboard

As a JIRA user, you can add gadgets to a JIRA dashboard that you have created.

Once added, the gadget will appear on your JIRA dashboard and display information drawn from Crucible and FishEye.

You can also add Crucible gadgets to the Confluence dashboard. See the General Gadgets Documentation for more information.

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), Crucible (by default) will send a preset reminder to all of the pending reviewers, one working day before the deadline.

To edit the timeframe for the reminder, firstly edit the review, then click **Remove** next to **Send Reminder**. Now, set the day for the reminder email to be sent (a number of working days before the deadline).

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

Manual Reminders

Crucible can send manually-initiated reminders to all the reviewers that have not yet completed their reviews.

To do this, click **Tools** > **Notify Pending Reviewers**. The reminder message will be sent immediately.

🛈 This capability is only open to those participants who are authorised to summarize the review.